

Universidade de Aveiro

Departamento de Electrónica e Telecomunicações

Compressão Digital de Sinais Áudio
Aplicando
Critérios Perceptuais e
Adaptação para Trás

João Manuel de Oliveira e Silva Rodrigues

Licenciado em Engenharia Electrónica e Telecomunicações
pela
Universidade de Aveiro

Novembro de 1995

Resumo

Neste trabalho, 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, realçando a estrutura comum em que assentam, e caracterizam-se as variantes possíveis nos vários módulos dessa estrutura. Com base neste estudo, foi desenvolvido um novo sistema de codificação perceptual de áudio.

O novo codificador inclui um banco de filtros não uniforme, quantizadores logarítmicos de ganho adaptativo, codificação aritmética, e um modelo psico-acústico explícito para adaptar os quantizadores de acordo com critérios perceptuais. Ao contrário de codificadores perceptuais existentes, o novo sistema é adaptado para trás, isto é: a adaptação depende exclusivamente de amostras já quantizadas, e não do sinal original. Discutimos as vantagens do uso de adaptação para trás e mostramos que esta técnica pode ser aplicada com sucesso à codificação perceptual. Apresentam-se alguns resultados preliminares.

Abstract

In this work, the problem of digital coding of high quality audio signals is analysed, and the principles of perceptual coding are identified as the most satisfactory solution. We present a synthesis of the perceptual coding systems found in the literature, stressing their common structure, and we characterize the possible variants that each module in that structure allows. Based on this study, a new perceptual audio coding system was developed.

The new coder includes a nonuniform filter bank, gain-adaptive logarithmic quantizers, arithmetic entropy coding and an explicit psychoacoustic model to adapt the quantization according to perceptual considerations. Unlike existing perceptual coders, the new system is backward-adaptive, that is: adaptation depends exclusively on already quantized samples, not on the original signal. We discuss the advantages of backward-adaptiveness and show that it can be successfully applied to perceptual coding. Some preliminary results are presented.

Índice

1	Introdução	1
1.1	Motivação	1
1.2	Organização da Tese	2
2	Codificação de Áudio Digital	5
2.1	Sistemas de Codificação	5
2.1.1	Modelo Geral	5
2.1.2	Níveis de Codificação	6
2.1.3	Compressão de Sinais	7
2.1.4	Técnica de Projecto	8
2.2	Compressão de Sinais Áudio	9
2.2.1	Especificação do Problema	9
2.2.2	Motivação para as Técnicas de Codificação Perceptual	10
2.3	Codificação Perceptual no Domínio da Frequência	11
2.3.1	Bancos de Filtros usados em Codificadores Perceptuais	12
2.3.2	Atribuição de Bits ou Ruído	12
2.3.3	Quantização e Codificação	14
2.3.4	Alguns Exemplos	15
2.4	Técnicas Alternativas de Codificação	20
2.4.1	Codificação com Realimentação de Ruído	20
2.4.2	Quantização Vectorial	21
3	Bancos de Filtros para Codificação	25
3.1	Definições	25
3.1.1	Bancos de Filtros Multifrequência	25
3.1.2	Relação Entrada-Saída	26
3.1.3	Sinais Multiresolução	27
3.1.4	Bancos de Filtros e Transformadas	28
3.2	Características Desejáveis	29
3.3	Bancos de Filtros Uniformes	30
3.3.1	<i>Quadrature Mirror Filters</i>	30
3.3.2	<i>Conjugate Quadrature Filters (CQF)</i>	31
3.3.3	Pseudo-QMF	32
3.3.4	Bancos de Filtros com Reconstrução Perfeita	32
3.3.5	Bancos de Filtros Modulados com Reconstrução Perfeita	33
3.4	Bancos de Filtros Não Uniformes	34

3.4.1	Estrutura em Árvore	34
3.4.2	Estruturas <i>Split-and-Merge</i>	37
4	Codificador Perceptual com Adaptação para Trás	41
4.1	Estrutura do Codificador	41
4.2	Transformação Tempo-Frequência	42
4.3	Quantização e Codificação	44
4.3.1	Caracterização dos Sinais	44
4.3.2	Quantização	45
4.3.3	Codificação de Comprimento Variável	47
4.4	Algoritmo de Adaptação	48
4.4.1	Cálculo do Limiar de Mascaramento (Modelo Psico-acústico)	48
4.4.2	Cálculo dos Passos de Quantização	49
4.5	Adaptação para Trás	50
4.5.1	Características, Vantagens e Inconvenientes	50
4.5.2	Validação da Técnica	52
5	Resultados	55
5.1	Descrição dos Sinais de Teste	55
5.2	Avaliação da Qualidade	56
5.2.1	Método	56
5.2.2	Resultados	57
5.3	Débito	59
5.4	Resumo dos Resultados	60
6	Conclusões e Trabalho Futuro	63
6.1	Conclusões	63
6.2	Trabalho Futuro	64
A	Psico-acústica: Factos e Modelos	65
A.1	Limiar de Audição e Limiar de Dor	65
A.2	Bandas Críticas	65
A.3	Mascaramento	67
A.3.1	Efeitos de Mascaramento no Tempo	68
A.3.2	Mascaramento no Domínio da Frequência	69
A.3.3	Índice de Mascaramento	71
B	Ferramentas Desenvolvidas	75
B.1	Bancos de Filtros e Transformadas	75
B.1.1	Transformadas em Blocos	75
B.1.2	Transformadas Sobrepostas	76
B.1.3	Sinais Multiresolução	78
B.1.4	Estruturas Não Uniformes	81
B.1.5	Análise de Bancos de Filtros	84
B.2	Quantização	85
B.2.1	Quantização Genérica	85
B.2.2	Curvas de Compressão/Expansão	86

B.2.3	Análise de Quantizadores	86
B.3	Codificação Aritmética	88
B.4	Modelo Psico-acústico	90
B.5	Codificação Perceptual de Áudio	91

Lista de Abreviaturas

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

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

Capítulo 1

Introdução

A *revolução digital*, caracterizada pelo rápido desenvolvimento dos sistemas de processamento e transmissão de informação discreta, inicialmente confinada aos mais avançados centros de investigação, chega agora ao homem comum nas mais diversas formas. Os sinais áudio e de voz, anteriormente gravados analogicamente em discos de vinil ou em fitas magnéticas, estão hoje disponíveis em Discos Compactos (CD) ou em fitas digitais (DAT). O aparecimento das Redes Digitais com Integração de Serviços (RDIS), da difusão digital de áudio (DAB), e dos sistemas multimédia, vulgarizarão ainda mais o áudio digital.

O sucesso da representação digital de sinais pode atribuir-se a uma série de vantagens técnicas em relação à representação analógica [Jayant 84]: transmissão ou armazenamento virtualmente sem degradação; facilidade de regenerar, de proteger contra erros e de cifrar; integração fácil com qualquer outro tipo de informação digital; e a possibilidade de aplicação de uma vasta gama de técnicas de processamento de sinal que seriam impossíveis ou impraticáveis no domínio analógico. Apesar destas vantagens, a representação digital de sinais, e do áudio em particular, tem um grave problema: os códigos simples vulgarmente usados são pouco eficientes, requerendo um grande número de bits para armazenar informação que, em grande medida, é redundante ou irrelevante. Por exemplo, para atingir a qualidade desejada, o sinal gravado num CD requer 16 bits por amostra (706 kbit/s por canal), enquanto medidas da *entropia perceptual* de diversos trechos musicais indicam que cerca de 2 bits por amostra seriam suficientes [Johnston 88, Noll 93]. Em certas aplicações, a capacidade limitada do canal de transmissão ou dos sistemas de processamento não permite tal desperdício de recursos. Felizmente, a própria natureza digital do sinal possibilita uma solução parcial deste problema, mediante o uso de técnicas de compressão de dados digitais.

1.1 Motivação

Nos últimos anos tem-se registado uma forte actividade na área de codificação de sinais áudio de alta qualidade e de voz de banda larga. 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 Psico-acústica 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 para a frente por um algoritmo

que inclui um modelo perceptual. Os melhores sistemas conseguem codificação perceptualmente “transparente” com débitos tão baixos como 2 bits por amostra. No entanto, há a necessidade de reduzir ainda mais o débito, assim como a complexidade e o atraso de codificação. Em resultado disto, procura-se otimizar os algoritmos existentes bem como desenvolver algoritmos novos que permitam atingir esses objectivos.

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. Em particular, investigamos a utilização de retro-adaptação¹ no sistema, o que parece nunca ter sido feito anteriormente em codificadores perceptuais. Desenvolvemos um novo algoritmo de compressão e, simultaneamente, criamos um conjunto de ferramentas, num ambiente de programação adequado, para facilitar a simulação de codificadores de áudio. Os resultados obtidos com o novo codificador são animadores, apesar de revelarem ainda alguns problemas.

1.2 Organização da Tese

No Capítulo 2 faz-se uma abordagem ao tema desta tese: a codificação de sinais áudio de alta qualidade. Começa-se primeiro por “localizar” o problema no campo mais vasto da codificação digital em geral, e no campo da compressão de sinais e dados em particular. O problema específico de compressão de sinais áudio tratado neste trabalho é então definido, e justifica-se a necessidade das técnicas de codificação perceptual para o resolver. Algumas dessas técnicas são descritas, juntamente com diversos exemplos de codificadores 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. A importância que o módulo de decomposição tem no desempenho global de um codificador e a grande diversidade de alternativas disponíveis justificam um capítulo próprio para o que de outra forma seria tratado numa única secção do Capítulo 2. 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 relatamos o desenvolvimento de um novo algoritmo de codificação perceptual de áudio. O sistema proposto é “desmontado” e descrito bloco por bloco, pondo em evidência as decisões de projecto mais importantes. Este codificador aplica uma estratégia de adaptação para trás, o que o diferencia de outros codificadores perceptuais de áudio existentes. Por esta razão, o capítulo termina com uma discussão das vantagens desta abordagem e uma verificação da sua aplicabilidade em codificadores perceptuais.

No Capítulo 5 apresentamos alguns resultados preliminares respeitantes ao débito e à qualidade subjectiva do codificador proposto no Capítulo 4.

As conclusões desta tese, bem como possíveis direcções de trabalho futuro, são apresentadas no Capítulo 6.

O Apêndice A resume os principais resultados da Psico-acústica com relevância para a definição de modelos perceptuais para codificação de áudio.

¹Nesta tese usamos indiferentemente as expressões *retro-adaptação* e *adaptação para trás*.

No Apêndice B apresenta-se um lista comentada das funções desenvolvidas para projecto, simulação e análise dos vários componentes do sistema de codificação.

Capítulo 2

Codificação de Áudio Digital

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 4. Embora a compressão seja a motivação primordial deste trabalho, um sistema de codificação completo tem outras vertentes igualmente importantes que justificam uma breve abordagem ao problema geral da codificação de sinais. É o que se faz na Secção 2.1 que foca ainda aspectos relativos ao projecto de codificadores. A Secção 2.2 aborda o problema específico da compressão de sinais áudio e aí surge a motivação para as técnicas de codificação perceptual que são o tema das restantes secções do capítulo. Na Secção 2.3 descrevem-se as técnicas de codificação perceptual no domínio da frequência, e apresentam-se diversos codificadores de áudio actuais que usam esta estrutura de compressão. A Secção 2.4 apresenta outras técnicas de codificação de áudio, menos comuns, que também se podem aplicar segundo critérios perceptuais.

2.1 Sistemas de Codificação

2.1.1 Modelo Geral

A Figura 2.1 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.

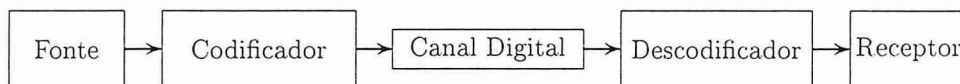


Figura 2.1: Um sistema de codificação genérico.

2.1.2 Níveis de Codificação

Na definição genérica de sistema de codificação dada atrás, podem-se enquadrar 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 [Zeng 91].
- 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 [Hamming 80]. 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, minimizando qualquer distorção 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 para garantir privacidade; e códigos para detecção e correcção de erros no canal. Os diversos

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

componentes são ligados em cascata, como mostra a Figura 2.2, criando uma hierarquia de codificação. Esta estrutura tem a vantagem de o projecto de cada nível ser praticamente independente dos restantes. Assim, o sistema de criptografia, por exemplo, pode ser projectado assumindo que o nível inferior é um canal fiável com probabilidade de erro muito baixa. O nível de compressão, por sua vez, pode assumir que dispõe de um canal simultaneamente fiável e seguro.

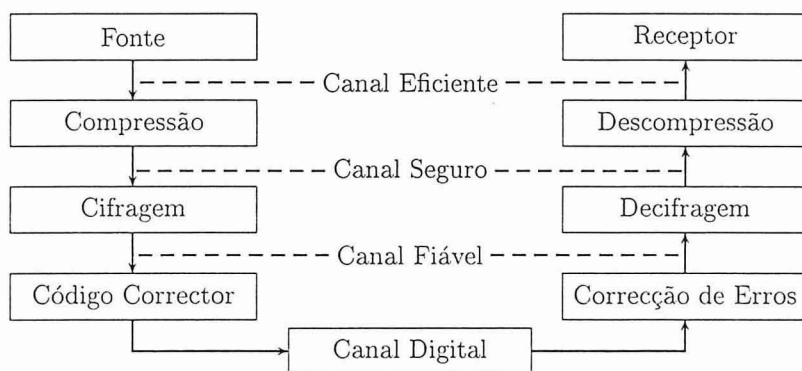


Figura 2.2: Um sistema de codificação com compressão, criptografia e correcção de erros. Cada nível de codificação corrige um “defeito” do nível abaixo e fornece um “canal” melhorado ao nível seguinte: a correcção de erros fornece um canal mais fiável, a cifragem torna-o mais seguro, e a compressão melhora a sua eficiência.

2.1.3 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 tese. 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 “direções” 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 prevêê-las com alguma confiança a partir de outras amostras. A distribuição não uniforme de amplitudes e a previsibilidade das amostras são exemplos de conteúdo estrutural 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 realmente 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 abaixo dos 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 2.3, 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 seja significativa em relação à sensibilidade do receptor. A curva decrescente, $D(R)$, representa a função distorção-débito ideal do sistema [Berger 68, Jayant 84]. O ponto C_0 representa o sinal original (já discreto), 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 discretos 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 a codificação NFC, usada no sistema *Super Bit Mapping*, como referido na Subsecção 2.4.1. 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.

2.1.4 Técnica de Projecto

O projecto de um sistema de codificação consiste em definir os blocos de codificação e decodificação de forma a cumprir determinadas especificações e satisfazer condições impostas

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

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

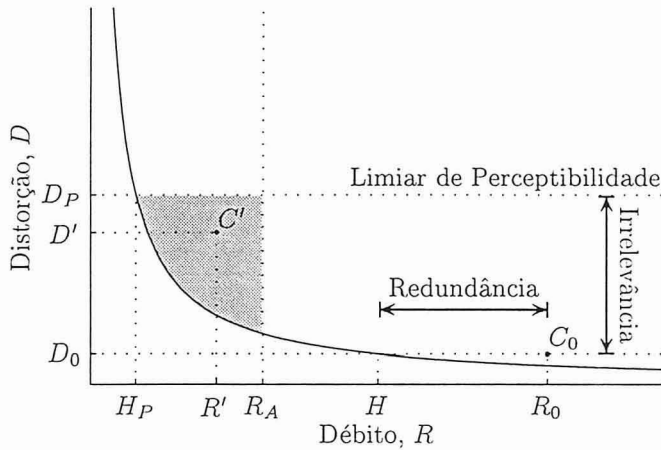


Figura 2.3: Função distorção-débito de um sistema de codificação.

pela fonte, pelo canal e pelo receptor. Por isso, é fundamental começar por caracterizar o 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 extraível, e determinar os limites de percepção do receptor, que estabelecem 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 decodificador; 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.

2.2 Compressão de Sinais Áudio

2.2.1 Especificação do Problema

O problema tratado nesta tese é o desenvolvimento de um sistema 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 qualidade CD, isto é: 44100 amostras por segundo quantizadas uniformemente com resolução de 16 bits.⁴ 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.

⁴Estes valores são assumidos no resto da tese mas poderia generalizar-se facilmente para outras resoluções e frequências de amostragem, dentro de certos limites.

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 electroacústica (por auscultadores ou altifalantes) e a propagação da onda acústica até ao ouvido. Assumiremos que todos estes processos são lineares e têm uma resposta plana na gama de frequências audíveis⁵ e que podem portanto ser caracterizados por um único parâmetro: o *ganho acústico* do sistema de reprodução.⁶ Uma vez que o ganho do amplificador pode ser regulado pelo ouvinte, terá que se considerar uma situação *worst-case* 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,⁷ 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 decodificador 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.

2.2.2 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,⁸ 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 [Johnston 91].⁹

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,

⁵Estas 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 [Sporer 92]. Apesar disso, em trabalhos de codificação de áudio, é comum partir-se destes pressupostos.

⁶O ganho acústico pode especificar-se através da razão entre o nível de pressão sonora (*Sound Pressure Level*, SPL) de um som e a amplitude do sinal digital que lhe deu origem.

⁷Esta 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 correção de erros.

⁸Por esta razão, também são chamadas de técnicas de *codificação de fonte*.

⁹A 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 [Jayant 84]. (O ganho de codificação é definido como o aumento de SNR que um dado código proporciona em relação ao PCM com o mesmo débito.)

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 otimizar os codificadores de sinal tradicionais, como a relação sinal-ruído ou o erro médio quadrático, não reflectem adequadamente a qualidade percebida pelo ouvido humano. Um exemplo disto é o caso de um sinal referido em [Johnston 91] 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 psico-acústica, 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.

2.3 Codificação Perceptual no Domínio da Frequência

O mascaramento de ruído por um som mais forte (ver Apêndice A) parece ser o fenómeno psico-acústico 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 2.4, 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 psico-acústico. 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 lateral necessária para a decodificação e produzem uma sequência binária para transmissão ou armazenamento. Neste passo pode também adicionar-se protecção contra erros, sequências para sincronização e informações diversas.

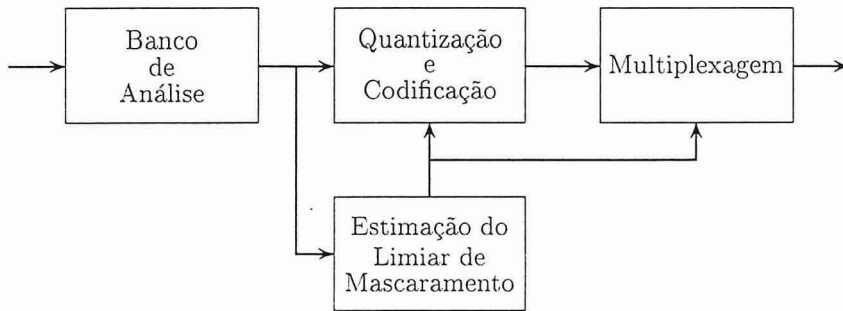


Figura 2.4: Diagrama de blocos genérico de um codificador perceptual a trabalhar no domínio da frequência.

No decodificador, 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.

2.3.1 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 da Secção 2.3.4, 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.

2.3.2 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 adaptativa (ATC) de Zelinski e Noll [Zelinski 77]: 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 Psico-acústico

Os métodos anteriores usam resultados simples da psico-acústica 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 melhor aproximação ao limiar correcto aplicando explicitamente um modelo das características psico-acústicas 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.

2.3.3 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 otimizado para a distribuição de amplitudes do sinal (quantizador de Max [Max 60]); o factor de escala ou passo de quantização pode ser adaptado dinamicamente a partir das amostras originais (adaptação para a frente) ou das amostras quantizadas anteriormente (adaptação para trás). 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 decodificador 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*.¹⁰ 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

¹⁰Outros códigos, como os de Lempel-Ziv, poderiam ser usados mas na literatura consultada não foi encontrada qualquer referência nesse sentido.

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 psico-acústico. 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 maior esforço computacional. O sinal começa por ser quantizado usando um ciclo de controlo de débito. O ruído de quantização real é determinado pela diferença entre os valores 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 a quantização e o 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 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.

2.3.4 Alguns 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 psico-acústica (ver [Schroeder 87]). 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 lateral. 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 psico-acústica 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 lateral.

Testes realizados com sinais de CD codificados a 128 kbit/s por canal 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 [Brandenburg 88], 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 lateral.

Perceptual Transform Coding (PXFm)

A codificação PXFM [Johnston 88] 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 psico-acústico 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 espraio 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 lateral 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.

Optimum Coding in the Frequency Domain (OCF)

Este codificador decompõe o sinal em 512 bandas usando uma transformada semelhante à do LC-ATC: 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 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. O ciclo interior, de controlo do débito, tem semelhanças com o do codificador PTFM. 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. O ciclo é iterado até que o dispêndio de bits seja suficientemente baixo.

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.

Hybrid Coder

Em [Brandenburg 90], 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 dos pré-ecos 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.

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 psico-acústico 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 [Spille 89], 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 [Vargas 93], 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 [Tsutsui 92]. 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.

MPEG-Audio

Os organismos internacionais de normalização ISO¹¹ e IEC¹² criaram em 1988 um grupo de especialistas denominado MPEG¹³ 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 [MPEG 92]. 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 a débitos mais elevados. Foi concluída em Abril de 1994 com a aprovação do documento [MPEG 94] conhecido vulgarmente pelo nome

¹¹International Standards Organization.

¹²International Electro-technical Commission.

¹³Moving Pictures Expert Group.

de MPEG-2. Actualmente decorre a fase 4 do MPEG, que visa aplicações operando a débitos muito baixos (dezenas de kbit/s).

No que diz respeito ao áudio, o MPEG-1 define três níveis (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.

É importante notar que a norma só define o decodificador 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.

Todos os níveis decompõem o sinal em 32 bandas usando um banco de filtros pseudo-QMF (ou polifásico) 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 usando uma MDCT com janela dinâmica de 6 ou 18 amostras (e janelas de transição). É necessário um andar adicional para redução do *aliasing* que resulta da combinação 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 psico-acústicos que podem ser usados em qualquer dos três níveis 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 de Fourier de 512 ou 1024 pontos e produzem o mesmo tipo de informação—uma estimativa da relação sinal-ruído mascarável¹⁴ 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

¹⁴SMR, *Signal-to-Mask Ratio*.

diminui para as bandas de índice crescente. Em média, estas estratégias permitem reduzir a fracção do *bit stream* dedicada à informação lateral 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 por *run-length*.

2.4 Técnicas Alternativas de Codificação

Apesar de os codificadores por transformada e por sub-bandas se terem vulgarizado no campo da codificação perceptual de áudio e possuírem diversas vantagens, um codificador perceptual não tem que trabalhar forçosamente no domínio da frequência. Existem outras técnicas, nomeadamente no domínio do tempo, que se podem aplicar à codificação perceptual e que convém estudar, não só para se ficar com uma perspectiva mais global do problema, mas também porque têm características interessantes que poderão contribuir para futuros desenvolvimentos nesta área.

2.4.1 Codificação com Realimentação de Ruído

Pode verificar-se que num codificador diferencial clássico (o chamado DPCM em malha fechada [Jayant 84]) o erro de reconstrução é igual ao erro de quantização e logo tem um espectro aproximadamente plano, o que não é conveniente em termos perceptuais.¹⁵

Uma generalização desta técnica temporal, 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*¹⁶ [Atal 79, Jayant 84] esquematizada na Figura 2.5. 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* (psd) do erro de reconstrução deste codificador é dada por [Atal 79]

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

onde $S_{qq}(z)$ é a psd 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.

É interessante notar que a introdução de *noise shaping* não aumenta a complexidade do decodificador e portanto, em aplicações que o permitam, pode-se implementar um algoritmo bastante complexo no codificador sem modificar os receptores.

¹⁵Mesmo que o quantizador seja relativamente grosseiro, o ruído de quantização é aproximadamente branco porque num codificador diferencial a entrada do quantizador é o erro de predição e este deve ser bastante decorrelacionado se o preditor for eficiente.

¹⁶*Noise Feedback Coding* (NFC).

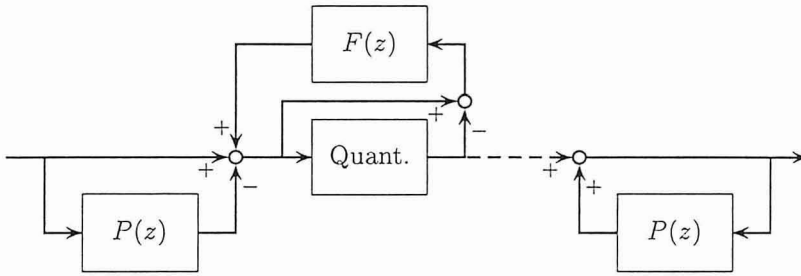


Figura 2.5: Codificador com realimentação de ruído (NFC).

Em [Schroeder 79], Schroeder, Atal e Hall desenvolveram uma medida objectiva de degradação de sinais de voz e sugeriram um codificador com realimentação de ruído projectado de forma a minimizar essa medida. A medida baseia-se num modelo psico-acústico 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.

Uma aplicação mais recente deste método é o sistema *Super Bit Mapping* [Akune 92] que usa princípios psico-acústicos para gravar discos compactos comuns com qualidade melhorada. O diagrama de blocos é o da Figura 2.5 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 existente, sem necessidade de substituir os receptores já instalados.

2.4.2 Quantização Vectorial

Todos os codificadores descritos acima se baseiam em quantizadores escalares que traduzem cada amostra ou coeficiente num símbolo a transmitir. Ora, um resultado conhecido da teoria da informação indica que se pode sempre conseguir melhor desempenho se se codificar vectores em vez de escalares, mesmo que a fonte produza sinais descorrelacionados. Esta constatação é um forte argumento para o estudo de métodos de conversão directa de vectores contínuos em símbolos discretos, isto é: métodos de 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”.¹⁷

¹⁷Trata-se portanto de um algoritmo de procura do *vizinho mais próximo*, como os usados em reconhecimento de padrões.

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 minimizem o valor esperado da distorção, tendo em conta a distribuição de probabilidade da fonte. Na prática, como não se conhece a verdadeira distribuição dos vectores, usa-se um processo iterativo que tenta minimizar a distorção média avaliada para uma sequência de vectores de treino. Se a sequência de treino for suficientemente grande, esta aproximação dá bons resultados mesmo para processos não estacionários e não ergódicos [Gray 84].

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 aproximadamente exponencial da complexidade computacional com a dimensão dos vectores. Se considerarmos um *codebook* com N vectores de dimensão d , então a taxa de transmissão é $R = \frac{1}{d} \log_2 N$ bits por amostra e, equivalentemente, $N = 2^{dR}$. Para armazenar esse *codebook* é necessária uma capacidade de memória de $dN = d2^{dR}$ palavras. Simultaneamente, se usar um algoritmo de pesquisa exaustiva, o codificador precisa de fazer 2^{dR} operações por vector ou seja $2^{dR}/d$ operações por amostra. Pode verificar-se que para R fixo o consumo de recursos depende exponencialmente da dimensão d . 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 [Gersho 83, Gray 84]. 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 [Gersho 91]. 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 vec-

torial. 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. Não encontramos, na literatura consultada, qualquer referência ao uso de quantização vectorial para a codificação de áudio de qualidade. As exigências e as características particulares dos sinais áudio são possivelmente os principais obstáculos à introdução desta técnica recente. De qualquer forma, parece-nos que a sua versatilidade e as provas que já deu noutros domínios justificam um estudo mais aprofundado no futuro da sua aplicabilidade à codificação perceptual de áudio.

Capítulo 3

Bancos de Filtros para Codificação

Neste capítulo apresentamos um estudo resumido dos 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 multiresolução 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

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* [Crochiere 83, Vaidyanathan 93].

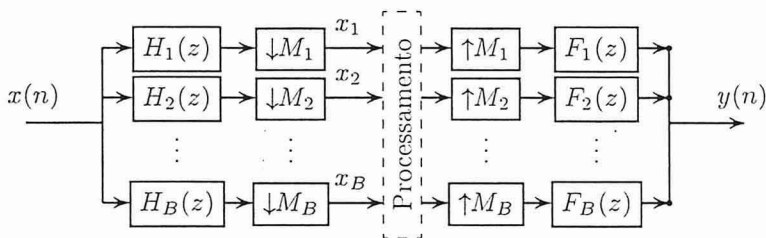


Figura 3.1: Sistema de processamento em sub-bandas com factores de decimação inteiros. Os blocos marcados com setas para baixo (\downarrow) e para cima (\uparrow) representam as operações de subamostragem (decimação) e sobreamostragem (interpolação), respectivamente.

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* extraem 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

$$\sum_{b=1}^B \frac{1}{M_b} = 1, \quad (3.1)$$

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 o banco de filtros deve ter decimação máxima (ver Secção 3.2), por isso assumiremos 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).¹ 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

¹Considerando que não há processamento e que os filtros de análise e síntese são lineares e invariantes no tempo (LTI).

(desde que não haja processamento das sub-bandas). Nesse caso, diz-se que o sistema tem *cancelamento de aliasing*, o que implica que se torne 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 Multiresolução

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:

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

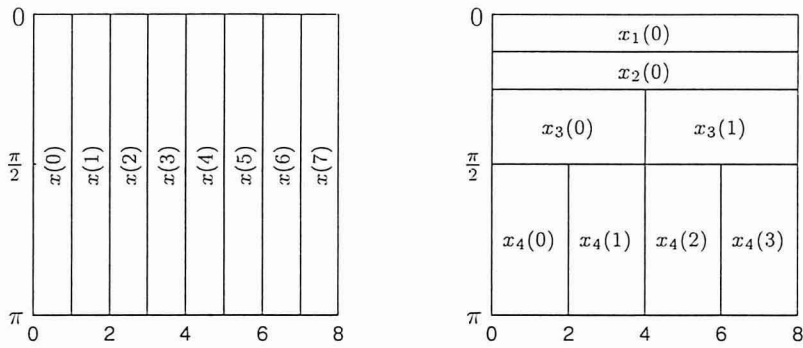
Um período

(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 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 ,

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

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 multiresolução* 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 multiresolução. Um sinal multiresolução pode ser representado graficamente por um *mosaico* como os seguintes.²

²Nestes 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.



O mosaico da direita representa um bloco do sinal 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.³

3.1.4 Bancos de Filtros e Transformadas

Historicamente, os conceitos de banco de filtros decimados e de transformada em blocos surgiram e foram desenvolvidos separadamente. Nos últimos anos tem havido uma progressiva integração destes conceitos resultante do desenvolvimento da teoria dos sistemas multifrequência [Crochiere 83, Vaidyanathan 93] por um lado, e das transformadas com sobreposição⁴ [Cassereau 85, Malvar 89, Malvar 92] por outro. Actualmente sabe-se 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.

Uma consequência das origens distintas destas duas formas de decomposição de sinais é a diferente notação e nomenclatura usada nos dois casos para noções equivalentes ou equiparáveis, como se resume abaixo.

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

Nesta tese usamos termos de ambas as colunas, de acordo com o uso mais comum na literatura ou consoante a conveniência de clareza de exposição.

³Para uma definição formal da localização e dimensões de uma célula no tempo e na frequência, veja-se [Bodin 95, Chapter 2].

⁴*Lapped Transforms*.

3.2 Características Desejáveis

Johnston e Brandenburg [Johnston 91] 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 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 espalhamento [Sporer 93], 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 [Jayant 84].
- 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 Apêndice A), o que daria origem aos chamados *pré-ecos*. Devido à grande variação de largura das bandas críticas (entre 100 Hz e 4 kHz, aproximadamente), é impossível 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 pouca complexidade computacional.

Alguns codificadores usam um segundo banco de filtros para fazer a análise espectral usada no modelo psico-acústico 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.

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 impulsivas 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 todas 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

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 [Crochiere 83, Vaidyanathan 93]

$$Y(z) = T(z)X(z) + A(z)X(-z) \quad (3.2)$$

onde

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

⁵Na 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).

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:

$$H_2(z) = H_1(-z), F_1(z) = H_1(z), F_2(z) = -H_1(-z). \quad (3.3)$$

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

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 [Smith 86]. Pode-se, no entanto, otimizar os filtros para aproximar essa condição. Neste 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 [Johnston 80], 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 [Jain 84].

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 [Smith 86]:

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

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 [Vaidyanathan 88]. 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 [Vaidyanathan 93]. 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*.

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

⁷Para 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.

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 [Malvar 92, Section 3.3]). Para minorar esse fenómeno é necessário usar filtros de ordem elevada, o que acarreta maior complexidade e atraso global.

A alternativa é usar bancos de filtros projectados de raiz para $M > 2$ canais. Um dos primeiros sistemas desenvolvidos para decomposição em M bandas criticamente decimadas foi o chamado *pseudo-QMF*. Cox [Cox 86] refere o desenvolvimento independente destes filtros em [Nussbaumer 81] e [Rothweiler 83], 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 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 [Vaidyanathan 93] 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 paraunitariedade 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) [Malvar 89] 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 $f(n)$:

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

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

para $b = 1, 2, \dots, M$. As frequências de modulação correspondem às frequências centrais de M bandas uniformes com empilhamento ímpar [Crochiere 83] 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 vimos.

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 [Gopinath 93] 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) [Princen 86]; *modulated lapped transforms* (MLT) [Malvar 90a]; e também, DCT modificada (MDCT). Em [Malvar 90b] 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 [Koilpillai 92], pondo em evidência a paraunitariedade 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 [Vaidyanathan 93, Section 6.4.3].

3.4 Bancos de Filtros Não Uniformes

Um banco de filtros multiresolução 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 combinação de bancos uniformes eficientes usando estruturas em árvore ou estruturas *split-and-merge*.

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.

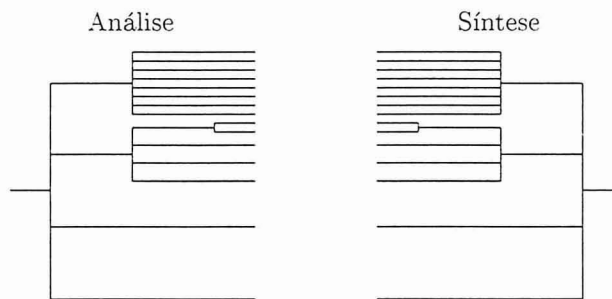


Figura 3.2: Decomposição não uniforme usando uma estrutura em árvore.

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 equalizar 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

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

decomposição em oitavas. As *discrete-time wavelet transforms* (DTWT) também podem ser implementadas por estruturas diádicas [Soman 93]. Outro exemplo de estrutura em árvore são as *hierarchical lapped transforms* (HLT), nas quais os andares de decomposição são formados por ELTs [Malvar 92].

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 pelo mais lento. 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 decodificadores; ou reparti-los igualmente para manter uma dualidade 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* 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 frequência 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* [Vaidyanathan 93], pode reduzir-se essa cascata a um único filtro decimado aplicando a seguinte regra:

$$\rightarrow \boxed{A(z)} \rightarrow \boxed{\downarrow P} \rightarrow \boxed{B(z)} \rightarrow \boxed{\downarrow Q} \rightarrow \equiv \rightarrow \boxed{A(z)B(z^P)} \rightarrow \boxed{\downarrow PQ} \rightarrow .$$

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

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.

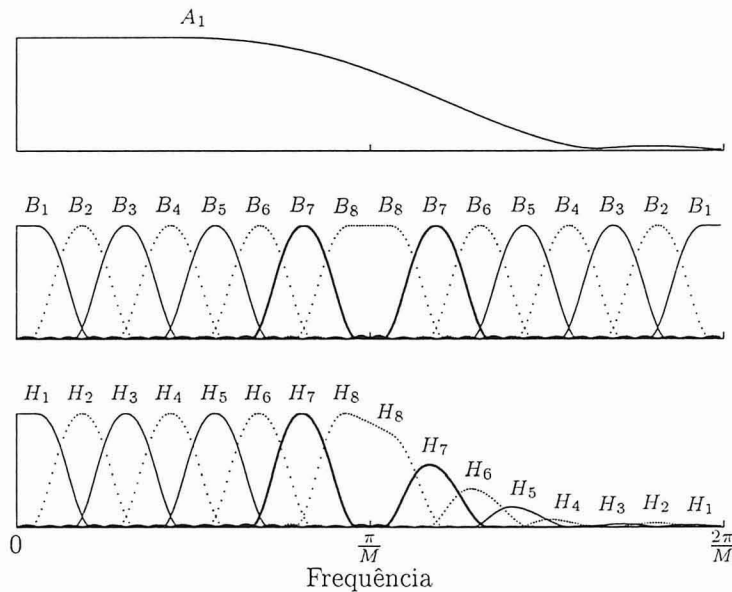


Figura 3.3: Funções de transferência de uma estrutura de análise em árvore. As funções de transferência do segundo andar, $B_b(z)$, aparecem comprimidas e replicadas M vezes. Representa-se apenas uma fracção do espectro total. O eixo vertical mede a amplitude das respostas numa escala linear.

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:

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. A segunda solução implica a utilização de filtros de ordem elevada, com grande complexidade computacional e atraso. O pós-processamento com estruturas em “borboleta” como no Layer III parece interessante porque não aumenta muito a complexidade. No entanto, esta técnica 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 mais vantajosas de implementar bancos uniformes.¹⁰ Não encontramos na bibliografia consultada qualquer referência à forma de aplicar esta técnica a estruturas não uniformes em geral.¹¹

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.

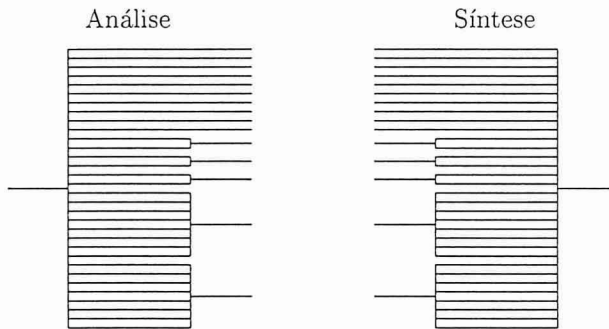


Figura 3.4: Decomposição não uniforme usando uma estrutura *split-and-merge*.

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.

¹⁰O Layer III usa esta técnica para aproveitar o banco de filtros que já existe para os Layers I e II.

¹¹É certo que o Layer III faz uma decomposição não uniforme quando usa “janelas curtas” em algumas bandas, porém [MPEG 92] não esclarece como é que a redução de *aliasing* se processa neste caso.

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 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 [Nayebi 93].¹² 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}, \dots, A_{J+L} e B_1, \dots, B_L

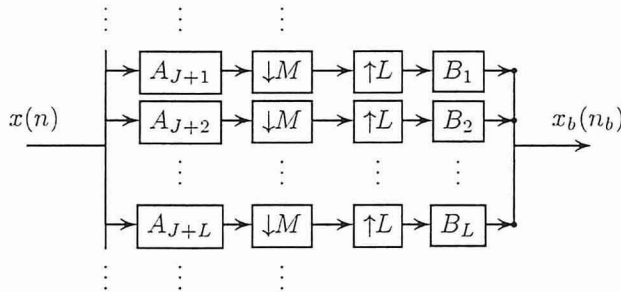


Figura 3.5: Uma banda de um banco de análise SAM.

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 do sinal de entrada extraídas pelos filtros A_{J+1} a A_{J+L} . Nesse caso, a função de transferência é dada por:

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

¹²Isso só é possível se os factores de interpolação e de decimação forem primos entre si, permitindo a permutação dos decimadores e dos interpoladores.

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 impulsivas 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 impulsivas 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” idêntica. Esta estratégia foi proposta em [Cox 86]. No mesmo artigo, Cox usou uma modificação do programa de Johnston [Johnston 80] 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 *extended lapped transform*—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

Codificador Perceptual com Adaptação para Trás

Neste capítulo descreve-se o projecto de um novo algoritmo de codificação perceptual de áudio que desenvolvemos no âmbito desta tese. O novo codificador baseia-se num banco de filtros não uniforme, quantizadores logarítmicos de ganho variável, e codificação aritmética. O ganho dos quantizadores é adaptado dinamicamente por um modelo psico-acústico explícito. Ao contrário de outros codificadores perceptuais, este sistema é retro-adaptativo: a adaptação depende exclusivamente de amostras já quantizadas e não das amostras originais.

Na Secção 4.1 apresentamos em traços gerais a estrutura e o funcionamento global do sistema de codificação. Nas secções seguintes descrevem-se as características e alguns aspectos de projecto de cada um dos blocos que o compõem: a estrutura de transformação tempo-frequência (Secção 4.2); o processo de quantização e a codificação de comprimento variável (Secção 4.3); e o algoritmo de adaptação baseado num modelo psico-acústico (Secção 4.4). Finalmente, na Secção 4.5 discutimos as vantagens do uso de adaptação para trás e mostramos que esta técnica é válida mesmo para débitos muito baixos, quando a quantização é muito grosseira.

4.1 Estrutura do Codificador

A Figura 4.1 representa a estrutura do sistema de codificação desenvolvido neste trabalho. Chamamos-lhe: *Codificador Perceptual de Áudio com Adaptação para Trás*, BAPAC¹ ou ainda PAC-AQB.

O bloco de análise (T) faz uma decomposição do sinal em bandas não uniformes usando uma estrutura *split-and-merge*. As amostras contínuas de cada banda são convertidas em símbolos discretos por quantizadores não uniformes de ganho adaptativo. Os símbolos são codificados numa sequência binária usando um código de comprimento variável (VLC) para reduzir a redundância. O algoritmo de adaptação “recupera” as amostras acabadas de quantizar \hat{x} e, usando um modelo psico-acústico que avalia continuamente a evolução do limiar de mascaramento em função desse sinal, determina os novos ganhos ou passos Δ para quantização das amostras futuras com erro inferior a esse limiar. No destino, a sequência binária é decodificada, os símbolos são desquantizados e as amostras recuperadas são passadas ao

¹*Backward-Adaptive Perceptual Audio Coder.*

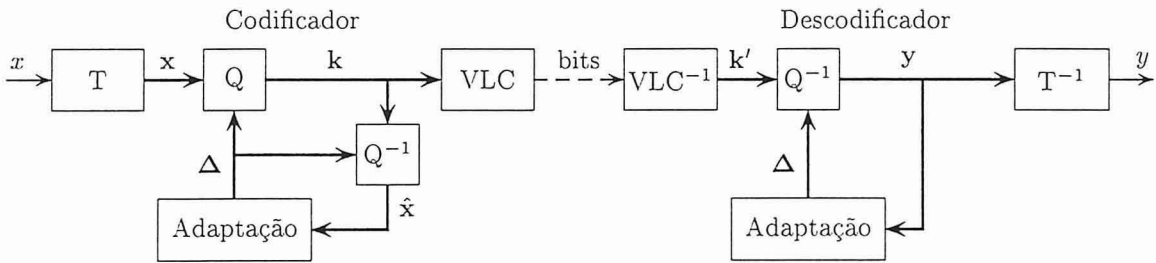


Figura 4.1: Sistema de codificação perceptual com adaptação para trás (PAC-AQB). As linhas finas representam seqüências temporais enquanto as linhas grossas denotam sinais multiresolução.

banco de síntese que produz uma réplica do sinal original. Os passos de quantização são adaptados pelo mesmo algoritmo que o codificador usa e a partir dos mesmos valores quantizados.² Deste modo, codificador e decodificador são mantidos em sincronismo e não há necessidade de transmitir informação lateral.

4.2 Transformação Tempo-Frequência

Uma das primeiras decisões tomadas no projecto deste sistema foi a de usar uma única transformada para fazer simultaneamente a decomposição do sinal em bandas e a análise espectral usada no cálculo do limiar de mascaramento.³ As razões que motivam a acumulação das duas funções na mesma estrutura já foram referidas na Secção 3.2. Uma razão adicional que influenciou esta decisão foi a constatação de que em vários codificadores actuais, para estimar o limiar de mascaramento, o sinal passa por diversas representações no plano tempo-frequência e as manipulações que sofre nas várias conversões introduzem erros na estimação. Um dos modelos psico-acústicos propostos pelo MPEG, por exemplo, parte de uma decomposição de Fourier em 512 *linhas* espectrais, agrupa linhas adjacentes em *partições* de larguras aproximadamente proporcionais às bandas críticas para calcular o perfil de excitação, reconverte o resultado para o domínio das linhas espectrais para combinação com o limiar absoluto, e finalmente, reagrupa as linhas em *bandas* para determinação dos factores de escala da quantização. Cada uma das conversões entre domínios—de linhas para partições, de partições para linhas e de linhas para bandas—tornam o codificador mais complexo e implicam cálculos aproximados com a conseqüente acumulação de erros na avaliação do limiar de mascaramento. Parece-nos vantajoso evitar este tipo de conversões usando uma única representação do sinal ao longo de todo o processo de codificação.

Para satisfazer as difíceis especificações de resolução temporal e espectral requeridas, decidimos usar uma decomposição não uniforme do sinal que garantisse bandas com larguras de aproximadamente meio Bark. A alternativa seria o uso de transformadas com comutação

²Se não houver erros de transmissão, como supomos neste trabalho, então $k' = k$ e portanto, $y = \hat{x}$.

³Lembremos que na norma MPEG (e no MUSICAM que lhe serviu de base) não é esta a estratégia proposta—o modelo psico-acústico faz a sua própria análise espectral independente do banco de filtros do codificador—o que, aliás, é compreensível uma vez que o banco de filtros usado tem pouca resolução (32 bandas nos *Layers I e II*).

dinâmica de janela como em [Ferreira 92] ou no MPEG Layer III, mas a primeira opção parece-nos uma forma mais “natural” de lidar com a resolução não uniforme do ouvido humano. Optámos por uma estrutura *split-and-merge* composta de ELTs que permite reconstrução perfeita, é decimada maximamente e tem um baixo custo computacional. A estrutura da transformada é resumida na Tabela 4.1. O primeiro andar é formado por uma ELT que

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

Tabela 4.1: Especificação da estrutura de decomposição do codificador.

divide o sinal em 256 bandas uniformes, subamostradas criticamente. No segundo andar, as primeiras 32 bandas passam inalteradas para a saída através de duas unidades de atraso, enquanto as bandas restantes são agrupadas em grupos de duas, quatro, oito ou dezasseis, e recombinadas com IELTs para produzir bandas mais largas com melhor resolução temporal. Todas as saídas têm linhas de atraso projectadas para normalizar o atraso global do sistema pelo atraso das transformadas mais “lentas”. Com esta estrutura, cada bloco de 256 amostras consecutivas é transformado num bloco de 256 coeficientes distribuídos espectral e temporalmente como se representa na Figura 4.2. Assim, as 62 bandas resultantes têm resolução espectral entre 86 e 1378 Hz e resolução temporal entre 5.8 e 0.36 ms, como mostra a Tabela 4.2. Esta tabela também mostra as larguras das bandas em Bark.

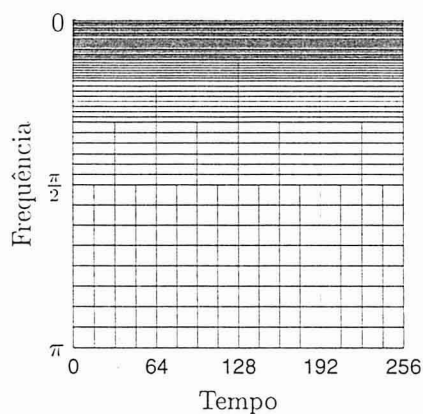


Figura 4.2: Mosaico do sinal multiresolução produzido pela transformada projectada.

Os protótipos das ELTs usadas na transformada foram otimizados individualmente 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

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

Tabela 4.2: Resolução das 62 bandas resultantes, no tempo (Δt) e na frequência (Δf e Δz).

forma a garantir a semelhança dos filtros protótipo, e assim permitir o cancelamento parcial de *aliasing* na estrutura SAM, como discutido na Subsecção 3.4.2.⁴ A Figura 4.3 representa a amplitude das funções de transferência obtidas nas várias bandas da transformada SAM resultante.

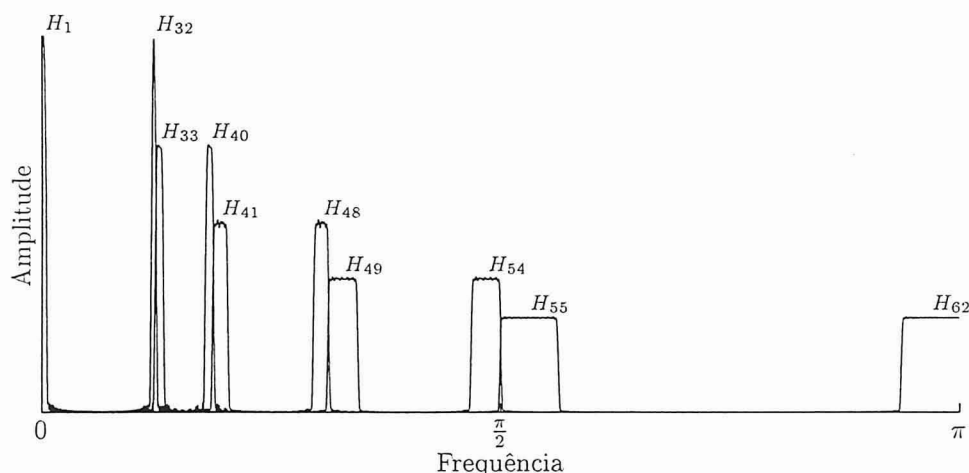


Figura 4.3: Respostas em frequência dos filtros da transformada de 62 bandas descrita nesta secção. Só se apresenta a amplitude das respostas; a fase é aproximadamente linear na banda passante.

A estrutura SAM completa (directa ou inversa) pode ser implementada com 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.

4.3 Quantização e Codificação

4.3.1 Caracterização dos Sinais

Para projectar os quantizadores e o código de comprimento variável é necessário caracterizar estatisticamente os sinais em cada banda. Interessa particularmente conhecer a sua

⁴Mais precisamente, fixou-se o factor de sobreposição, $K = 2$, e a frequência de corte do protótipo, $\omega_S = \pi/M$, o que equivale a $LB = 2K = 4$ e $TP = \frac{\omega_S}{\pi/M} - \frac{1}{2} = \frac{1}{2}$.

distribuição e gama dinâmica. Assim, fizemos algumas medições a partir de um sinal considerado representativo do universo de sinais a codificar. Esse sinal, que baptizámos de *mix*, foi formado por concatenação de curtos trechos de músicas diversas—castanholas, cravo, voz feminina, orquestra sinfónica, *rock*—e contém zonas impulsivas, sons harmónicos, sinais de banda larga e uma gama dinâmica considerável. Entre trechos foi usado *fade-out/fade-in* para evitar transições bruscas.

A Figura 4.4 apresenta quatro histogramas típicos, obtidos das bandas #10, #38, #50 e #56, extraídas do sinal *mix*. A variável estudada em cada histograma é x_b/Δ_b , onde x_b é a amplitude das amostras da banda e Δ_b é uma quantidade proporcional ao limiar de mascaramento estimado dinamicamente em função do próprio sinal pelo algoritmo de adaptação descrito na Secção 4.4.⁵ Observando os gráficos podemos tirar algumas conclusões quanto à distribuição estatística em cada banda:

- As distribuições são simétricas e têm média nula.
- O desvio padrão varia de banda para banda.
- Algumas bandas, como a #38, apresentam uma distribuição aproximadamente gaussiana mas a maioria é modelada melhor por uma distribuição laplaciana ou, nalguns casos, por distribuições ainda mais localizadas em torno da média.

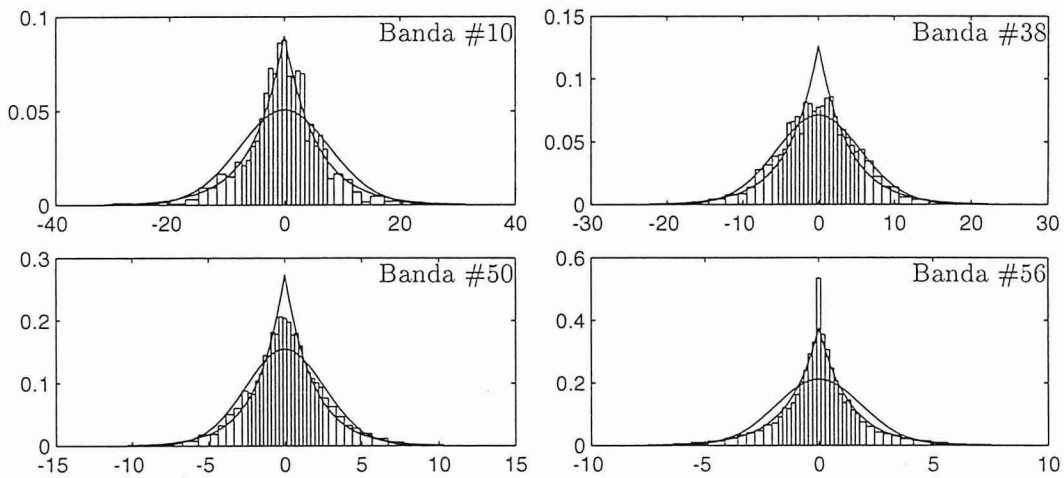


Figura 4.4: Histogramas das amostras de quatro bandas do sinal *mix*. Em cada gráfico, o eixo horizontal representa a fracção entre a amplitude da amostra e o passo de quantização: x_b/Δ_b . O passo de quantização foi avaliado a partir de amostras não quantizadas. As curvas sobrepostas representam distribuições laplacianas e gaussianas de média nula e variância igual à medida nos próprios sinais.

Os valores de maior amplitude ocorrem durante variações bruscas da amplitude do sinal, quando o modelo psico-acústico ainda não teve tempo de se adaptar. A maior amplitude verificada no sinal *mix* foi de 585Δ , ocorrida na banda #21 durante um “toque” de castanholas.

⁵Faz mais sentido estudar esta variável do que simplesmente x_b porque a resolução dos quantizadores deve acompanhar a variação do limiar de mascaramento, ou seja: proporcionalmente a Δ_b .

4.3.2 Quantização

Para manter a complexidade baixa, decidimos usar quantizadores com a mesma forma em todas as bandas. Deste modo, apenas o ganho ou passo de quantização Δ é variado de banda para banda e ao longo do tempo. Optámos por quantizadores com 127 níveis de reconstrução y_k distribuídos simetricamente em torno de zero segundo a curva de compressão logarítmica representada na Figura 4.5. Na verdade trata-se de uma aproximação por troços lineares a uma característica logarítmica, semelhante à que se faz em telefonia digital com a lei- μ ou a lei-A. Os níveis de decisão x_k foram colocados nos pontos médios dos níveis de reconstrução adjacentes, como convém para minimizar a distorção [Max 60].⁶

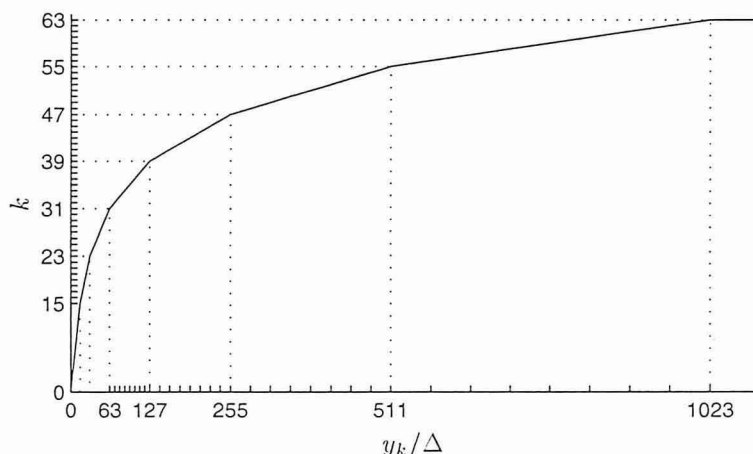


Figura 4.5: Curva de compressão-expansão do quantizador.

Para amostras de amplitude compreendida entre -16Δ e 16Δ , que representam uma fracção considerável do total, o quantizador comporta-se como um quantizador uniforme de passo Δ . Para amostras mais elevadas, que ocorrem especialmente durante transições do sinal, o passo de quantização aumenta “proporcionalmente” à amplitude, o que permite cobrir uma gama dinâmica muito grande sem saturação do quantizador. Uma vez que a adaptação do codificador se baseia em amostras quantizadas previamente, esta característica é particularmente importante para permitir a passagem dos “picos” mais fortes do sinal e assim assegurar uma adaptação rápida do sistema.

A Figura 4.6 mostra o desvio padrão σ_e do erro introduzido pelo quantizador logarítmico em função do passo de quantização Δ . Os gráficos foram obtidos por integração numérica da equação

$$\sigma_e^2 = \sum_{k=-63}^{63} \int_{x_k}^{x_{k+1}} (x - y_k)^2 p(x) dx$$

onde $x_{-63} = -\infty$, $x_{64} = +\infty$ e $p(x)$ é a função densidade de probabilidade da entrada x . Tendo em conta a caracterização dos sinais feita atrás, considerámos duas distribuições

⁶Nesta secção, x_k e y_k representam respectivamente, os níveis de decisão e de saída de um sistema de quantização genérico com entrada x e saída y . x é a saída de uma das bandas, mas omite-se o índice b para simplificar a notação.

$p(x)$ —laplaciana e gaussiana—de média nula e variância σ_x^2 . Sobre os gráficos traçamos ainda a linha de equação

$$\sigma_e^2 = \frac{\Delta^2}{12} \quad (4.1)$$

que traduz o desempenho assintótico de um quantizador uniforme de passo Δ .

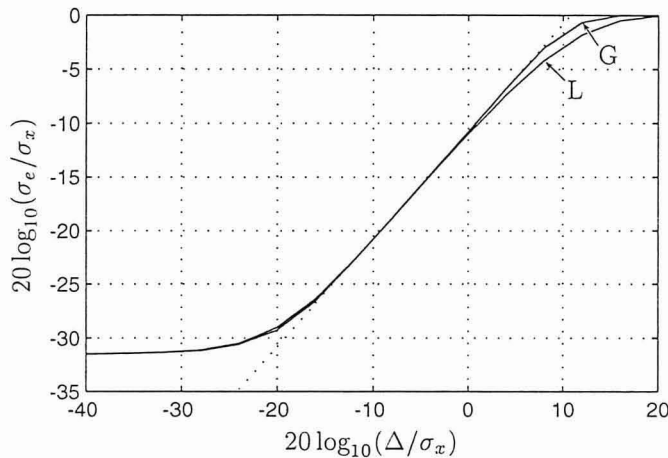


Figura 4.6: Potência de ruído versus passo do quantizador. As duas curvas traçadas correspondem às situações em que o sinal x tem distribuição (G)aussiana ou (L)aplaciana.

Verifica-se que o comportamento do quantizador não depende muito da distribuição do sinal de entrada e que, para valores moderados de Δ , é semelhante ao de um quantizador uniforme. Quando Δ diminui muito abaixo de σ_x , passa a predominar a característica logarítmica do quantizador, fazendo com que a relação sinal-ruído não cresça muito acima de 31 dB.⁷ Esta propriedade é desejável porque sabemos que a relação sinal-ruído em cada banda nunca precisa de superar o valor máximo do índice de mascaramento: cerca de 28 dB. Deste modo, o quantizador fornece um mecanismo intrínseco de modelação de ruído que evita a sobre-codificação dos “ataques” do sinal.

4.3.3 Codificação de Comprimento Variável

Naturalmente, tendo em conta a distribuição não uniforme do sinal de entrada e dos níveis de quantização, os quantizadores debitam cada um dos 127 símbolos possíveis com diferente frequência. Para eliminar a redundância associada a esta distribuição não uniforme, os símbolos são codificados com um codificador aritmético adaptativo [Witten 87].

Um codificador aritmético é controlado por um modelo da sequência a codificar que não é mais que uma tabela da probabilidade de ocorrência de cada símbolo. Uma vez que cada banda apresenta uma distribuição de probabilidades potencialmente diferente, o nosso sistema mantém 62 tabelas distintas e comuta entre elas à medida que processa os símbolos de cada banda. Assim, o codificador aritmético integra simultaneamente a função de multiplexagem dos dados das várias bandas. A adaptação dos modelos é feita após a transmissão de cada

⁷Note-se que $SNR = 10 \log_{10}(\sigma_x^2/\sigma_e^2) = -20 \log_{10}(\sigma_e/\sigma_x)$.

símbolo, aumentando na tabela adequada a sua probabilidade de ocorrência. Aumenta-se também a probabilidade do símbolo simétrico de modo a manter sempre a simetria da distribuição modelada. No “arranque”, as tabelas de probabilidades dos modelos são inicializadas com distribuições LQ de média nula e com níveis de decisão iguais aos do quantizador descrito acima.⁸ A variância de cada distribuição foi otimizada de forma a minimizar a quantidade de informação que resultaria da codificação não adaptativa do sinal mix. Deste modo garante-se um modelo razoável desde o início da codificação.

Considerámos a utilização de outros códigos de comprimento variável, nomeadamente códigos de Huffman, mas a codificação aritmética tem uma série de vantagens que conduziram à sua escolha: é mais eficiente sem necessitar de fazer agrupamento de símbolos como nos códigos de Huffman; separa claramente o processo de codificação do modelo da fonte; admite modelos adaptativos com grande facilidade; é comparativamente rápido, especialmente para modelos adaptativos.

4.4 Algoritmo de Adaptação

O algoritmo de adaptação baseia-se num modelo psico-acústico explícito para estimar dinamicamente o limiar de mascaramento. Os passos dos quantizadores são então determinados de forma que o ruído de quantização não ultrapasse esse limiar. Devido ao processo de adaptação para trás, o algoritmo é executado incrementalmente em alternância com a quantização das amostras do sinal. Isto é necessário porque uma amostra só pode ser quantizada quando o respectivo passo de quantização está disponível e o passo de quantização só pode ser calculado depois de todas as amostras anteriores terem sido quantizadas.

Trata-se portanto de um algoritmo que procura manter qualidade perceptual constante. Naturalmente, um sistema destes tem débito variável. Consideramos que isto não é uma limitação porque certas aplicações (principalmente de armazenamento) não requerem débito constante. Actualmente interessa mesmo considerar sistemas de débito variável para transmissão nas futuras redes exploradas em modo assíncrono (ATM). Além disso, é sempre possível incluir um ciclo de controlo de débito em aplicações que exijam débito fixo, mas para isso convém primeiro estudar o sistema em malha aberta.

4.4.1 Cálculo do Limiar de Mascaramento (Modelo Psico-acústico)

O modelo psico-acústico calcula, a partir do sinal multiresolução quantizado \hat{x} , uma estimativa do limiar de mascaramento do sinal, isto é: uma estimativa da máxima energia de ruído que se poderá introduzir em cada banda sem que se torne audível. Esta estimativa é actualizada amostra-a-amostra em cada banda e portanto a saída deste bloco de processamento é um sinal multiresolução com a mesma distribuição espectral e temporal que a sua entrada. Internamente, todas as operações são feitas também neste domínio, evitando quaisquer conversões. O cálculo é feito em quatro passos que se descrevem a seguir.⁹

Espalhamento temporal As amostras do sinal \hat{x} são elevadas ao quadrado para produzir valores da energia contida em cada célula do plano tempo-frequência. Os valores de cada

⁸Definimos distribuição LQ (laplaciana quantizada) como a função de probabilidade discreta que resulta de uma determinada quantização de uma variável aleatória contínua com distribuição laplaciana. Esta distribuição tem, assim, três parâmetros: a variância e o valor médio da variável contínua, e o vector de níveis de decisão que caracterizam a quantização.

⁹Mais pormenores sobre os métodos usados aqui são dados no Apêndice A.

banda são passados por um filtro passa-baixo para extrair uma média de curto prazo da energia nessa banda. O filtro é um IIR de primeira ordem que produz uma estimativa centrada da média e tem uma resposta impulsional exponencial decrescente. Este tipo de filtro é usado simultaneamente como modelo do fenómeno de pós-mascaramento, descrito na Subsecção A.3.1. Assim, as constantes de tempo τ_b dos filtros nas várias bandas foram projectadas de forma a reflectir a dependência deste fenómeno com a frequência segundo a fórmula empírica A.2. 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 A.3. A operação de renormalização é também incluída neste passo. Como a resolução varia de banda para banda, cada célula recebe contribuições parciais das células mais longas nas bandas anteriores e contribuições múltiplas das células mais curtas nas bandas seguintes. Esta operação é linear e pode ser implementada por uma multiplicação com uma matriz de dimensão 256×256 . Felizmente, esta matriz é bastante esparsa uma vez que muitas células não interagem entre si por não serem contemporâneas ou por estarem muito afastadas na frequência. No total, a matriz só tem 7082 elementos não nulos (11%).

Dedução do índice de mascaramento O limiar de mascaramento é derivado da excitação por subtracção (numa escala logarítmica) de um índice de mascaramento. Neste modelo usámos o índice de mascaramento de ruído devido a tons (TMN) avaliado pela fórmula empírica A.4. Não foi adoptada, nesta fase, qualquer interpolação com o índice de mascaramento de tons por ruído (NMT). Esse poderá ser um melhoramento a introduzir no futuro, mediante a inclusão de uma medida de tonalidade como em [Johnston 88] ou [Brandenburg 90].

Correcção para o limiar absoluto Os valores do limiar de mascaramento são comparados com o limiar absoluto e corrigidos (substituídos por este) caso lhe sejam inferiores. Assume-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.

No que respeita à complexidade, este modelo requer cerca de 30 multiplicações e 30 adições por amostra processada. Necessita também de cerca de 8000 palavras de memória fixa para armazenamento de coeficientes, e quantidades desprezáveis de memória de acesso aleatório.

4.4.2 Cálculo dos Passos de Quantização

A saída do modelo psico-acústico é uma estimativa Ψ^2 da potência do ruído que se pode injectar no sinal sem que se torne perceptível. Porém, os quantizadores são controlados pela variação dos seus passos de quantização Δ . Torna-se necessário relacionar estas duas variáveis.

Vimos atrás que na região de interesse (SNR entre 0 e 28 dB), a potência do erro de quantização σ_e^2 é modelada razoavelmente pela Equação 4.1. Assim, para se obter ruído de quantização proporcional ao limiar de mascaramento ($\sigma_e^2 = \phi^2 \Psi^2$), calcula-se o passo de quantização pela fórmula:

$$\Delta = \sqrt{12} \phi \Psi.$$

O factor de proporcionalidade ϕ é um parâmetro global do sistema que permite controlar a qualidade e, indirectamente, o débito da codificação: com $\phi = 1$, o ruído introduzido

acompanha o limiar estimado; para $\phi > 1$, o ruído supera o limiar degradando a qualidade e reduzindo o débito; para $\phi < 1$, estabelece-se uma margem de segurança entre ruído e limiar à custa de um débito mais elevado. Das expressões anteriores, poderíamos esperar que uma duplicação de ϕ produzisse um aumento de cerca de 6 dB no perfil de ruído. Na prática porém, a variação vai ser inferior porque há sempre uma fracção dos quantizadores que estão a trabalhar numa “região” com relações sinal-ruído muito perto de 0 dB onde Δ é sistematicamente subestimado.

4.5 Adaptação para Trás

Uma característica fundamental do sistema de codificação aqui proposto é a de ser um sistema com adaptação para trás. Apesar de muito usada em codificadores tradicionais de voz e até de áudio, de que é exemplo a norma G.722 do CCITT [Maitre 88], a técnica de adaptação para trás não tem sido aplicada em codificadores perceptuais. Por esta razão, justifica-se um estudo das suas vantagens e inconvenientes e uma verificação da sua aplicabilidade à codificação perceptual de áudio.

4.5.1 Características, Vantagens e Inconvenientes

Para melhor apreciação das implicações do uso de adaptação para trás (AQB), considere-se uma variante do sistema com adaptação para a frente (AQF) como mostra a Figura 4.7. Esta versão do sistema segue a estrutura usual dos codificadores perceptuais no domínio da frequência representada na Figura 2.4.

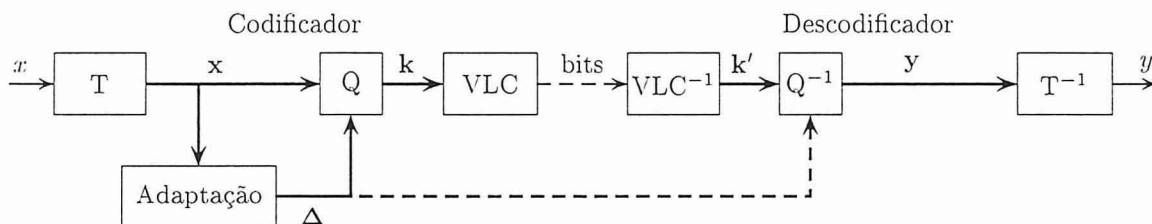


Figura 4.7: Versão com adaptação para a frente do sistema de codificação proposto (PAC-AQF).

Comparando com a Figura 4.1, 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 x ; no sistema AQB, a adaptação depende do sinal já quantizado \hat{x} . Em consequência desta diferença estrutural, os dois sistemas apresentam um conjunto características distintas que analisamos abaixo.

Informação lateral Um sistema AQB praticamente não tem informação lateral.¹⁰ Podemos dizer que no sistema da Figura 4.1 a informação lateral é formada apenas pelo factor de qualidade ϕ , que não precisa de ser transmitido frequentemente. Mesmo que fosse

¹⁰ Consideramos como *informação lateral* apenas a informação necessária à adaptação do decodificador, não incluindo informação de sincronismo ou correcção de erros.



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. Em AQF, pelo contrário, a informação lateral 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 lateral (passos de quantização 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 lateral 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 lateral seria inoportável. Num sistema AQB, por não haver informação lateral, a adaptação pode ser feita amostra-a-amostra sem custo adicional, 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 lateral. Em compensação, o descodificador é mais complexo porque inclui um modelo psico-acústico 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 com adaptação para a frente, 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.¹¹

Qualidade dos parâmetros estimados Num sistema AQB, a estimação do limiar de mascaramento e dos parâmetros de quantização associados é afectada pelo erro de quantização introduzido no sinal. Para baixos ritmos de transmissão, a quantização pode ser muito grosseira e afectar significativamente o cálculo dos parâmetros, piorando ainda mais a qualidade de codificação. Num sistema AQF, apesar de o limiar ser avaliado a

¹¹No 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 ADPCM robusto [Jayant 84].

partir do sinal original, os parâmetros também não são exactos já que eles próprios sofrem uma quantização para transmissão ao receptor, e além disso, são em muito menor número que num sistema AQB.

As principais vantagens dos sistemas AQB são a eliminação da informação lateral, 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 decodificador, 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 retro-adaptação em codificadores perceptuais. Esta questão será tratada a seguir.

4.5.2 Validação da Técnica

Um problema que surge da análise acima é a possibilidade de o erro de quantização comprometer irremediavelmente a estimação do limiar de mascaramento num sistema AQB.

Numa situação normal, como o limiar em cada célula depende da energia contida em muitas outras células (anteriores e de outras bandas), 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 espraio espectral *filtram* os erros de quantização. As piores situações que podemos antever são casos extremos quando: 1) o sinal numa banda diminui muito abaixo do respectivo passo de quantização ou 2) o sinal ultrapassa o ponto de saturação do quantizador. No primeiro caso, o sinal é quantizado como nulo, o que provoca uma tendência de decaimento rápido no limiar e no passo de quantização.¹² 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 “encosta” ao limiar imposto pelas outras bandas ou ao limiar absoluto. No segundo caso, a energia do sinal é subavaliada, o que obriga o limiar a aumentar mais lentamente que o desejável. Isto conduz a uma sobre-codificação desnecessária do sinal. Uma forma de minorar este problema é garantir que o codificador tem uma gama dinâmica capaz de abranger grandes flutuações.

Desta discussão heurística podemos concluir 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. No entanto, para verificar e quantificar o efeito real da quantização no cálculo do limiar, decidimos realizar uma experiência simples:

- Usando o sinal *mix* como fonte, registámos o limiar “exacto” Ψ_0 obtido pelo esquema PAC-AQF a partir do sinal não distorcido x .
- Registámos os limiares Ψ_ϕ calculados a partir do sinal quantizado \hat{x} , usando a versão PAC-AQB com três níveis de qualidade progressivamente pior: $\phi = 1, 2, 4$.

¹²Isto se o quantizador for *mid-tread* ou seja, se tiver um nível de saída nulo. Se for *mid-rise*—com nível de decisão nulo—o sinal será quantizado como $\pm\Delta/2$, resultando num decaimento mais lento ou até num crescimento instável se não forem tomados os cuidados devidos. De qualquer forma, não é aconselhável usar quantizadores *mid-rise* quando os sinais têm distribuição muito localizada em torno de zero, como sucede neste caso [Jayant 84].

- Finalmente, calculámos as relações Ψ_ϕ/Ψ_0 em dB, e traçámos os histogramas correspondentes.

A Figura 4.8 mostra um desses histogramas: o obtido com $\phi = 4$, que representa uma quantização muito grosseira e corresponde a um débito médio inferior a 1.4 bits por amostra.

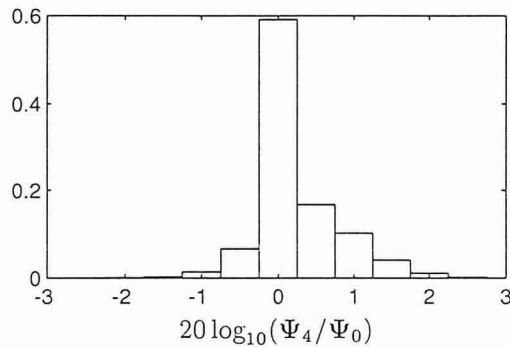


Figura 4.8: Histograma das diferenças (em dB) entre o limiar estimado a partir de amostras quantizadas grosseiramente ($\phi = 4$) e o limiar estimado a partir de amostras não quantizadas.

A tabela seguinte resume os resultados obtidos:

	Percentagem de amostras com Ψ_ϕ/Ψ_0 menor que:					
	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.

Capítulo 5

Resultados

Neste capítulo apresentamos alguns resultados relativos ao desempenho, em termos de débito e qualidade de reprodução, do codificador perceptual descrito no Capítulo 4. Na Secção 5.1 descreve-se o conjunto de trechos musicais originais e os vários codificadores que foram usados nos testes. A Secção 5.2 apresenta a metodologia usada e os resultados obtidos nos testes informais de audição realizados para avaliar a qualidade de codificação. Os resultados relativos ao débito de cada codificador são apresentados na Secção 5.3. Finalmente, na Secção 5.4 resumem-se os resultados obtidos para facilitar a comparação dos codificadores testados.

5.1 Descrição dos Sinais de Teste

A tabela abaixo mostra os trechos musicais usados nos testes, a sua duração e uma breve caracterização. A maioria dos sinais foram extraídos de um disco compacto emitido pela EBU¹ com material destinado especificamente a testes subjectivos [EBU 88]. Os sinais, originalmente estereofónicos, foram convertidos para monofónicos fazendo a média dos dois canais.

Trecho	Duração (seg)	Comentário
Castanholas	6.76	Muitas transições, “ataques” fortes.
Cravo	5.27	Tonal, banda larga.
Sarasate	12.07	Acordes de viola.
Sting	17.93	Música <i>Rock</i> .
Stravinsky	14.07	Sinfonia de instrumentos de sopro.
Suzanne	9.71	Voz feminina sem acompanhamento.
Violino	9.66	Tonal.

Criámos quatro versões codificadas de cada trecho passando o sinal original por sistemas de codificação diferentes. Três dessas versões (V1, V2 e V3) foram obtidas com o nosso codificador para níveis de qualidade de $\phi = 1$, $\phi = 2$ e $\phi = 3$. A quarta versão (V4) foi produzida com uma implementação *shareware* do MPEG *Layer III*,² para débito de 64 kbit/s.

¹*European Broadcasting Union.*

²Os programas *l3enc* e *l3dec* desenvolvidos no Fraunhofer-Gesellschaft.

Versão	Obtida com o codificador
V1	BAPAC, $\phi = 1$
V2	BAPAC, $\phi = 2$
V3	BAPAC, $\phi = 3$
V4	Layer III, 64 kbit/s

A intenção era possibilitar a classificação do nosso algoritmo a diferentes níveis de qualidade e débito, dispondo simultaneamente de um padrão de referência contra o qual se pudesse “calibrar” a escala de avaliação.

5.2 Avaliação da Qualidade

5.2.1 Método

Há dois tipos de medidas de qualidade para avaliar o desempenho de codificadores: medidas objectivas e medidas subjectivas. As medidas objectivas mais comuns são a relação sinal-ruído (SNR) e diversas variantes como a SNR segmental (SNRSEG) ou a SNR com ponderação espectral [Jayant 84, Appendix E]. Já referimos no Capítulo 2 a inadequação destas medidas para a avaliação da qualidade percebida por um ouvinte humano. Recentemente foram desenvolvidas medidas objectivas de qualidade baseadas em critérios psico-acústicos [Schroeder 79, Brandenburg 92, Paillard 92, Beerends 92, Stuart 94, De 94]. Algumas destas medidas apresentam uma forte correlação com resultados de testes subjectivos [Beerends 92]. As medidas subjectivas de qualidade [Jayant 84, Appendix F] baseiam-se na apreciação de sinais codificados por um painel de ouvintes escolhidos criteriosamente. Os ouvintes classificam ou pontuam os vários sinais e a medida de qualidade é extraída dos resultados por processos estatísticos. Uma das medidas subjectivas mais usadas é a *Mean Opinion Score* (MOS), dada pela média das pontuações atribuídas pelos ouvintes numa escala de N pontos (geralmente 5 pontos). Estas medidas são mais significativas que as medidas objectivas porque resultam da avaliação feita pelo destinatário final da codificação—o ouvinte humano—mas infelizmente obrigam à realização de testes demorados.

Devido à inadequação das medidas objectivas tradicionais e à indisponibilidade de medidas mais significativas, decidimos avaliar o desempenho do algoritmo de codificação que desenvolvemos com base em testes subjectivos de *estímulo triplo com referência escondida* [Noll 93]. (Esta metodologia foi usada em testes realizados no âmbito do MPEG e do CCIR, tendo mostrado resultados bastante fiáveis.) Num teste deste tipo, apresentam-se três sinais ao ouvinte: R, X e Y. O sinal R é sempre o sinal original ou sinal de referência. Dos sinais X e Y, um é de novo o original enquanto o outro é uma versão codificada que se pretende avaliar. O ouvinte é informado disto e deverá tentar identificar qual dos dois sinais é o original, atribuindo-lhe a pontuação máxima. O ouvinte pode escutar os três sinais livremente, pela ordem que entender e quantas vezes quiser. A sua tarefa consiste em avaliar a *degradação* perceptível de cada um dos sinais X e Y em relação à referência R dando pontuações na escala adjectival de cinco valores dada na recomendação 562 do CCIR:

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

Desenvolvemos um programa em Tcl/Tk que possibilita a realização dos testes de forma autónoma pelo ouvinte. O programa apresenta uma interface gráfica fácil de utilizar (Figura 5.1) com “botões” para audição dos três sinais R, X e Y; e escalas para dar as pontuações. Quando o ouvinte termina um teste, o programa regista a sua composição, as pontuações atribuídas, o número de audições de cada sinal, a data e a hora.

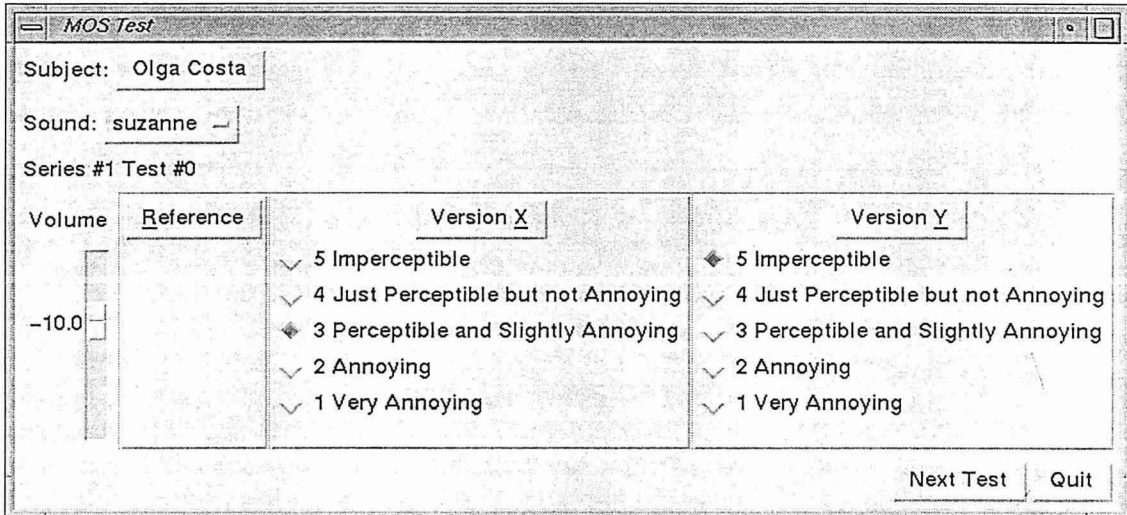


Figura 5.1: Programa de condução dos testes de audição.

Os testes decorreram num laboratório sem isolamento acústico especial. Foram realizados numa estação de trabalho *SiliconGraphics IRIS INDIGO* com *hardware* especializado para sinais áudio e usaram-se auscultadores SONY MDR CD550. Os ouvintes podiam regular livremente o volume de audição no programa de teste. Participaram 10 pessoas nos testes de audição, duas mulheres e oito homens, com idades compreendidas entre os 25 e os 40 anos. Destes, metade considera-se “ouvinte experiente” ou “conhecedor”. Seis dos ouvintes estudaram música e/ou tocam algum instrumento musical.

Cada ouvinte completou uma série de 8 testes para cada um dos 7 trechos musicais disponíveis. Cada série incluía dois testes de cada uma das quatro versões codificadas. Mais especificamente, cada série era composta pelos testes

R/R/V1 R/R/V2 R/R/V3 R/R/V4
R/V1/R R/V2/R R/V3/R R/V4/R

apresentados segundo uma ordem determinada pseudo-aleatoriamente pelo programa de teste no início da série. Apesar do programa permitir a interrupção dos testes em qualquer altura, a maioria dos ouvintes fez os 56 testes numa única sessão, com uma duração média de cerca de 1 hora.

5.2.2 Resultados

Para cada trecho musical, calculámos a média

$$\bar{S} = \frac{1}{N} \sum_{i=1}^N S_i$$

e o desvio padrão

$$\sigma_S = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (S_i - \bar{S})^2}$$

das pontuações S_i dadas pelos ouvintes a cada versão codificada. Cada versão foi testada (e pontuada) duas vezes por cada ouvinte, resultando em $N = 20$ amostras para as medidas estatísticas. A Tabela 5.1 mostra estes resultados.

Trecho	V1		V2		V3		V4	
	\bar{S}	σ_S	\bar{S}	σ_S	\bar{S}	σ_S	\bar{S}	σ_S
Castanholas	4.20	0.83	3.85	0.88	3.70	0.92	4.20	0.62
Cravo	4.30	0.86	3.45	1.00	2.55	1.15	4.30	0.73
Sarasate	4.60	0.50	3.75	1.07	2.40	1.31	4.75	0.44
Sting	4.75	0.44	4.65	0.59	4.30	0.80	4.70	0.47
Stravinsky	4.85	0.37	4.50	0.69	3.90	0.79	4.40	0.99
Suzanne	3.00	1.03	1.85	0.81	1.40	0.60	3.25	1.02
Violino	3.00	0.92	1.65	0.88	1.20	0.62	3.40	1.23

Tabela 5.1: Pontuação média (\bar{S}) e desvio padrão (σ_S) obtidos por cada codificador nos vários trechos.

Outro estudo feito foi a contagem do número de identificações erradas do original no par de sinais X e Y. Considerou-se uma identificação como errada quando a pontuação dada ao sinal original diferia de 5. Também se contabilizou o número total de audições (de R, X e Y) que os ouvintes necessitaram para fazer a classificação de cada versão. Este número pode servir como medida da “dificuldade” sentida pelos ouvintes em distinguir as degradações nos sinais de maior qualidade. Estas contagens são apresentadas na Tabela 5.2.

Trecho	Identificações erradas				Número de audições			
	V1	V2	V3	V4	V1	V2	V3	V4
Castanholas	5	3	4	4	172	132	139	187
Cravo	3	0	0	2	302	193	102	190
Sarasate	1	0	0	0	133	159	120	176
Sting	1	2	1	1	129	136	145	175
Stravinsky	3	0	2	1	158	142	104	138
Suzanne	0	0	0	2	102	62	50	83
Violino	1	0	0	1	95	75	53	104

Tabela 5.2: Número de identificações erradas e número de audições em 20 testes de cada versão e trecho.

Pode observar-se que a qualidade subjectiva atingida pelos vários codificadores depende muito do trecho codificado. Os trechos “Suzanne” e “Violino” mostraram-se os mais críticos em todos os codificadores. Não surpreendentemente, esses trechos foram considerados pela generalidade dos ouvintes como os mais “fáceis” de distinguir e classificar. Em contrapartida,

os ouvintes queixaram-se da “dificuldade” em distinguir o sinal original particularmente nos trechos “Sting” e “Stravinsky”. O número de audições corrobora em parte os comentários dos ouvintes. No entanto, o trecho “Cravo”, apesar de ter sido o mais ouvido, não foi geralmente indicado como dos mais “difíceis”, o que pode ter a ver com a sua curta duração. Mesmo nos sinais codificados com maior qualidade, verifica-se que os ouvintes erraram poucas vezes a identificação do sinal original. Outro aspecto importante é que a degradação introduzida quando se aumenta ϕ no codificador BAPAC (de V1 para V2 para V3) é muito mais notória em alguns trechos do que noutros. Um exemplo disto são os trechos “Castanholas” e “Sarasate”: o primeiro desce de uma pontuação de 4.20 em V1 para 3.70 em V3, enquanto o segundo cai de 4.60 para 2.40, nos mesmos codificadores.

5.3 Débito

O codificador que desenvolvemos tem débito variável. Para caracterizar as flutuações temporais deste parâmetro, registámos o número de bits usados para codificar blocos de 256 amostras de cada um dos trechos de teste nos três níveis de qualidade V1, V2 e V3. Tendo em consideração que um eventual ciclo de controlo de débito usará possivelmente um *buffer* com mais que 256 amostras,³ contámos também o número de bits em blocos de 1024 amostras, que revela obviamente variações menores em relação à média. A Figura 5.2 mostra a evolução do débito ao longo do trecho “castanholas” codificado com $\phi = 1$ (V1). Este trecho é dos que apresenta variações mais bruscas de débito.

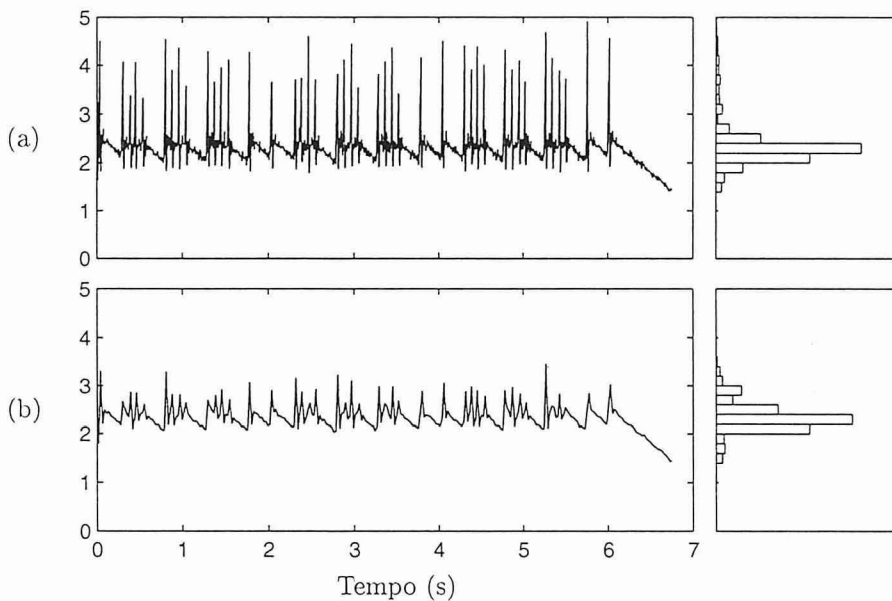


Figura 5.2: Variação do débito (em bits por amostra) medido no trecho “castanholas”, codificado com $\phi = 1$: (a) usando “janelas” de 256 amostras; (b) usando “janelas” de 1024 amostras. Também se apresentam os histogramas correspondentes.

³O controlo de débito no *Layer III*, por exemplo, é feito em tramas de 1152 amostras.

Na Tabela 5.3 apresenta-se o débito médio (independente da dimensão da “janela”) do codificador nos três níveis de qualidade com cada um dos trechos. Os valores máximos do débito medido em cada caso com janelas de 256 e de 1024 amostras são apresentados na Tabela 5.4.

Trecho	V1	V2	V3
Castanholas	2.34	1.77	1.46
Cravo	2.52	1.91	1.59
Sarasate	2.20	1.67	1.37
Sting	2.61	1.99	1.63
Stravinsky	2.20	1.66	1.36
Suzanne	2.40	1.82	1.49
Violino	2.18	1.63	1.33

Tabela 5.3: Débito médio (em bits por amostra) de cada versão e trecho.

Trecho	Janela de 256 amostras			Janela de 1024 amostras		
	V1	V2	V3	V1	V2	V3
Castanholas	4.92	4.19	3.73	3.46	2.75	2.34
Cravo	4.21	3.16	2.67	3.40	2.60	2.28
Sarasate	4.48	3.89	3.50	3.83	3.15	2.64
Sting	3.77	3.14	2.73	3.23	2.49	2.11
Stravinsky	3.59	2.93	2.54	3.06	2.42	2.07
Suzanne	4.22	3.49	3.10	3.43	2.77	2.37
Violino	3.21	2.48	2.20	2.81	2.25	2.02

Tabela 5.4: Valor máximo de débito atingido em cada versão e trecho.

Observando a Tabela 5.3, verifica-se que o débito médio do codificador BAPAC diminui coerentemente com o aumento do factor ϕ . Além disso, o débito flutua relativamente pouco em torno do valor médio, mesmo para sinais bastante variáveis: em blocos de 1024 amostras, o valor máximo do débito nunca atingiu o dobro do valor médio. Tendo em conta estes resultados, não deverá ser difícil conceber um algoritmo de controlo de débito para este codificador.

5.4 Resumo dos Resultados

A Tabela 5.5 mostra o débito médio e a pontuação média (MOS) obtida por cada codificador no conjunto dos sete trechos de teste, tomados todos com igual “peso”. Estes valores resumem o desempenho médio dos codificadores testados.

Em termos médios, pode dizer-se que o codificador V1 tem praticamente a mesma qualidade que o codificador V4 mas requer um excesso de quase 1 bit por amostra. O codificador V3 tem débito comparável ao V4 mas qualidade nitidamente inferior.

	V1	V2	V3	V4
MOS	4.10	3.39	2.78	4.14
Débito	2.35	1.78	1.46	1.42

Tabela 5.5: Qualidade e débito de cada codificador, no conjunto dos sete trechos. No valor de débito apresentado para o codificador V4 (*Layer III*) foram descontados os 32 bits de cabeçalho em cada bloco de 1152 amostras.

Capítulo 6

Conclusões e Trabalho Futuro

6.1 Conclusões

Nesta tese apresentámos um estudo das técnicas de codificação de sinais áudio baseadas em critérios perceptuais. Foram criadas bibliotecas de rotinas em MATLAB para o projecto e implementação de diversos componentes de um sistema de codificação perceptual: bancos de filtros e transformadas; quantizadores; códigos de comprimento variável; e modelos psico-acústicos. Foi desenvolvido um novo algoritmo de compressão que explora essencialmente duas possibilidades diferentes para tentar conseguir mais qualidade com menos débito: 1) o uso de uma estrutura de decomposição não uniforme, procurando um aproveitamento mais eficiente das características do ouvido humano; e 2) o uso de adaptação para trás, evitando transmissão de informação lateral.

Os resultados obtidos com o novo codificador são animadores. Tratando-se de uma primeira versão, o codificador conseguiu, com um débito médio de 2.35 bits por amostra, uma boa qualidade de reprodução, comparável à obtida com uma implementação muito otimizada do MPEG *Layer III* a 1.42 bits por amostra. Detectámos, no entanto, alguns problemas:

- Apesar de ter débito variável, o sistema subcodifica alguns trechos críticos, resultando em distorções perfeitamente audíveis e até desagradáveis.
- Quando se reduz o débito, alguns trechos revelam uma degradação abrupta de qualidade, enquanto outros quase não variam.

Temos algumas suspeitas quanto à origem destes problemas. Por um lado, é possível que as funções de transferência do banco de filtros não sejam suficientemente selectivas para os requisitos do ouvido humano. Isto justificaria a pior qualidade observada essencialmente nos trechos com grande conteúdo tonal. Por outro, estamos conscientes que o modelo psico-acústico usado é bastante limitado, não considerando o índice de mascaramento de tons por ruído nem a dependência do mascaramento com a intensidade.

Em todo o caso, pensamos que estes problemas não invalidam a principal conclusão da tese: o algoritmo de compressão desenvolvido apresenta desde já um desempenho razoável e tem potencialidades que devem ser melhor exploradas no futuro. Referimo-nos, em particular, ao uso da técnica de adaptação para trás, que parece ser uma inovação em codificadores perceptuais de áudio, e por isso mesmo deverá merecer um estudo mais aprofundado.

6.2 Trabalho Futuro

Analisando retrospectivamente esta tese, toma-se consciência dos erros cometidos e do muito que ficou por fazer. Queremos por isso, antes de terminar, deixar algumas pistas para trabalho futuro na área de compressão de sinais áudio. Para começar, podemos identificar uma série de melhoramentos a introduzir nos diversos blocos do codificador apresentado no Capítulo 4:

No Banco de Filtros Procurar estruturas de decomposição alternativas, procurando garantir respostas na frequência mais satisfatórias. Uma possibilidade interessante do ponto de vista da complexidade é a utilização de uma transformada uniforme, eventualmente uma MLT ou ELT, com 256 ou 512 bandas. Apesar de a transformada não ter uma resolução temporal tão boa, o uso de adaptação amostra-a-amostra deve permitir evitar os problemas de pré-ecos que comprometem o desempenho de codificadores adaptados para a frente com muito menos bandas. Além disso, o uso de uma transformação largamente aplicada noutros codificadores facilitará a comparação de resultados.

No Quantizador Investigar o uso de uma curva de compressão ligeiramente diferente que nas amplitudes mais altas resulte em níveis de saída ainda mais separados. Com isto consegue-se maior gama dinâmica ou menor número de níveis, enquanto a maior quantidade de erro introduzido não deverá tornar-se audível devido ao *overshoot* do limiar de mascaramento no início de sinais fortes.

No Codificador Aritmético Pensamos que a eficiência do codificador aritmético poderá melhorar bastante com um novo algoritmo de adaptação dos modelos de probabilidade. A ideia é aproveitar melhor o conhecimento que se tem sobre a distribuição de probabilidades dos símbolos nas sub-bandas. Assim, poderíamos ter um conjunto de algumas tabelas fixas com diversas distribuições típicas,¹ progressivamente “mais largas”, e a adaptação consistiria em comutar a tabela quando isso trouxesse benefício. Esta adaptação poderia ser feita para a frente, analisando previamente cada bloco e transmitindo o índice da tabela mais adequada; ou para trás, baseando-se em medidas do bloco anterior. Em vez de tabelas fixas, poderia talvez usar-se modelos paramétricos. Em qualquer dos casos, a adaptação dos modelos deveria tornar-se muito mais rápida, e acompanhar melhor as transições do sinal.

No Algoritmo de Adaptação Integrar um ciclo de controlo de débito, baseado na adaptação do factor ϕ , possivelmente bloco-a-bloco. No modelo psico-acústico, incluir o índice NMT ponderado com uma medida de tonalidade, e usar uma função de espriamento com declives dependentes da intensidade do sinal. Para reduzir a complexidade do cálculo do limiar pode procurar-se um processo de cálculo iterativo por aproximações sucessivas. Assim, em vez de recalcular tudo, bastaria fazer pequenas correcções de cada vez, partindo do limiar anterior como aproximação inicial.

A mais longo prazo, pensamos que haverá interesse em estudar melhor as técnicas de codificação alternativas referidas na Secção 2.4, nomeadamente a codificação com realimentação de ruído e a quantização vectorial.

¹Possivelmente, laplacianas quantizadas (ver Secção 4.3).

Apêndice A

Psico-acústica: Factos e Modelos

A.1 Limiar de Audição e Limiar de Dor

Sons diferentes com o mesmo nível de pressão sonora¹ em geral não são percebidos como sendo igualmente “fortes”. A sensação subjectiva de *sonoridade* de um som depende não só da sua intensidade mas também da sua frequência e qualidade. A Figura A.1 apresenta contornos de igual sonoridade de tons puros (sinusóides) em condições de campo aberto. Estas curvas demonstram a relativa insensibilidade do ouvido a tons de baixa frequência e a tons de frequência muito alta. A sensibilidade máxima ocorre perto dos 4 kHz, o que é aproximadamente a primeira frequência de ressonância do canal externo do ouvido. Há um segundo pico de sensibilidade perto dos 13 kHz, correspondente a uma segunda ressonância.

A curva inferior corresponde ao *limiar absoluto de audição* que indica a intensidade mínima audível em cada frequência. 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).

A.2 Bandas Críticas

O conceito de bandas críticas é possivelmente o resultado mais estabelecido e aceite da Psico-acústica porque resulta da concordância de uma série de resultados experimentais relativos à selectividade espectral do ouvido. Alguns desses resultados:

¹O *nível de pressão sonora* (Sound Pressure Level ou SPL) é uma medida da pressão sonora dada numa escala logarítmica (em dB) segundo a fórmula:

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

onde p_{rms} é a amplitude efectiva das flutuações de pressão do som. A pressão de referência, $p_0 = 2 \times 10^{-5}$ Pa, corresponde aproximadamente ao limiar de audição de um tom de 1 kHz. O *nível de intensidade* de um som, definido por

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

é uma medida logarítmica da intensidade do som—potência transmitida por unidade de área da frente de onda. A intensidade de referência, $I_0 = 10^{-12}$ W/m², 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, dependente da geometria da fonte e da sala. Ver [Morse 91, Houtsma 87].

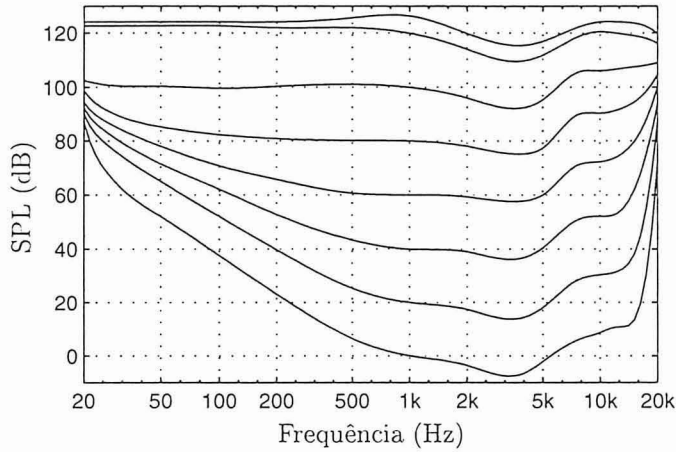


Figura A.1: Curvas de igual sonoridade de tons puros em condições de campo aberto.

- 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 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.

A largura da banda crítica em cada frequência, determinada por diversos processos, varia desde pouco menos de 100 Hz nas baixas frequências, até 12 a 26% da frequência central para frequências acima de 500 Hz. Com base nestas observações, foi adoptada uma nova escala de frequência mais adaptada às características perceptuais. Esta escala é graduada em Bark, onde um intervalo de 1 Bark corresponde à largura de uma qualquer banda crítica. Uma fórmula empírica que permite converter uma frequência de Hertz para Bark é:

$$\frac{z}{\text{Bark}} = 13 \arctan \frac{f}{1.3\text{kHz}} + 3.5 \arctan^2 \frac{f}{7.5\text{kHz}}. \quad (\text{A.1})$$

A região de frequências audíveis abrange cerca de 25 Bark. Isto não quer dizer que o ouvido tenha apenas 25 filtros. As bandas críticas são contínuas, isto é: podemos centrar uma banda crítica em qualquer frequência audível.

Esta escala 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. Este é mais um dado a sustentar o conceito de banda crítica.

A Cóclea: O Analisador Espectral do Ouvido

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. Depois de atravessarem o ouvido externo e o ouvido médio, as ondas de pressão acústica são transmitidas à base da cóclea e são transformadas numa vibração mecânica de uma membrana no seu interior—a membrana basilar. Células sensíveis (células ciliadas) dispostas ao longo desta membrana convertem as vibrações 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 da onda vai diminuindo ao longo do percurso e depende da frequência do estímulo. Destas características resulta que as altas frequências se propagam apenas distâncias muito curtas antes de serem fortemente atenuadas, enquanto as frequências baixas atingem distâncias maiores. Para cada frequência, a envolvente da onda progressiva atinge um valor máximo numa dada posição da membrana basilar. A cóclea funciona, portanto, como um analisador espectral mecânico.

Referências: [Ferreira 92, Allen 85, Lyon 88].

A.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 [Houtsma 87]:

- O mascaramento entre dois tons simultâneos é tanto mais efectivo quanto mais próximas forem as suas frequências.
- 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 por tons. É porém, mais efectivo.
- Um som forte consegue mascarar sons que comecem pouco depois do primeiro som terminar. Também consegue mascarar sons que ocorrem pouco antes do seu início.
- Sons procedentes de fontes espacialmente próximas mascaram-se mais do que sons afastados no espaço.

Em síntese, o mascaramento depende da relação de amplitudes dos dois sons e da sua proximidade no tempo, na frequência e no espaço. Nas subsecções seguintes, pormenoriza-se um pouco melhor a dependência do mascaramento com o tempo e a frequência. 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.

A.3.1 Efeitos de Mascaramento no Tempo

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 milisegundos. O efeito de mascaramento de sons posteriores é o chamado *pós-mascaramento*; o mascaramento de sons anteriores chama-se *pré-mascaramento*. A Figura A.2 ilustra a dependência temporal do fenómeno de mascaramento. A figura representa esquematicamente o limiar de mascaramento de um sinal de teste muito curto 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 de milisegundos. O pré-mascaramento só é significativo num período de poucos milisegundos, mas há um efeito de *overshoot* logo no início do sinal mascarante [Sporer 93]. 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 mais fracos iniciado anteriormente. 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.

Um modelo simples (e algo grosseiro) para o fenómeno de pós-mascaramento foi apresentado em [Beerends 92], onde se descreve uma medida para avaliar a qualidade perceptual de sistemas de processamento de áudio. O cálculo baseia-se na ideia de “espalhamento temporal” da energia em cada componente espectral do sinal. Este espalhamento é modelado por uma curva exponencial que é facilmente implementável por um filtro recursivo com um único atraso e ganho $\beta = e^{-T/\tau}$, onde T é o intervalo entre amostras consecutivas da mesma componente espectral, e τ é a constante de tempo do decaimento exponencial. Esta constante de tempo depende da frequência segundo uma lei que pode ser aproximada por:

$$\tau(f) = \frac{5}{f} + 2 \text{ ms.} \quad (\text{A.2})$$

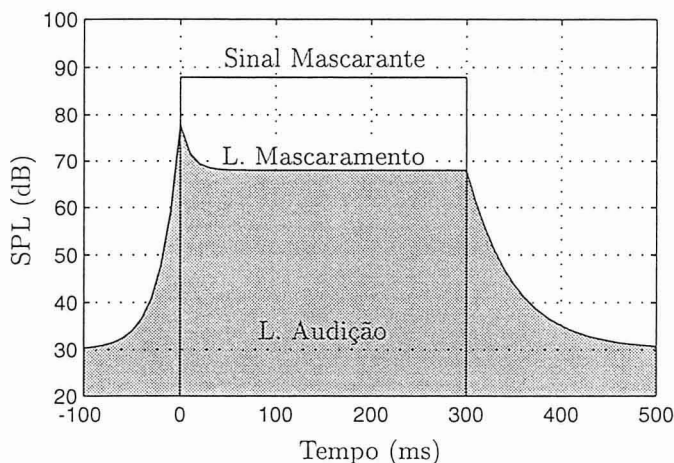


Figura A.2: Mascaramento temporal. O sinal mascarante é um *burst* de 300 ms de uma onda sinusoidal; o sinal de teste é um impulso de ruído de banda estreita (1 Bark) centrado na frequência do tom.

Nesse artigo, o pré-mascaramento não é modelado devido à sua curta duração relativamente à resolução temporal da análise espectral usada.

É pertinente notar que a maioria dos codificadores perceptuais existentes não explora 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 [Bosi 92], ou para implementar um outro mecanismo de protecção contra os *pré-ecos*² [Ferreira 92].

A.3.2 Mascaramento no Domínio da Frequência

O fenómeno de mascaramento depende fortemente da composição espectral dos sinais envolvidos. A Figura A.3 mostra o limiar de mascaramento de um tom na presença de uma banda subcrítica de ruído. O mascaramento é máximo quando o tom está no centro da banda de ruído mascarante. (A diferença entre os dois sinais nesta situação é o *índice de mascaramento*, tratado na próxima subsecção.) Para tons mais graves o limiar diminui com um declive de cerca de 27 dB/Bark, e para tons mais agudos o declive é de cerca de -10 dB/Bark. A forma da curva de mascaramento praticamente não depende da frequência do sinal mascarante desde que se use a escala perceptual graduada em Bark. Contudo, o declive decrescente da curva depende do nível do sinal mascarante—varia entre -20 e -5 dB/Bark para intensidades crescentes—e também é ligeiramente alterado em frequências muito baixas. Em experiências com tons a mascarar bandas de ruído ou a mascarar outros tons, observou-se essencialmente a mesma forma do limiar de mascaramento.

²Os pré-ecos são artefactos de codificação que podem surgir em codificadores a trabalhar no domínio da frequência, devido ao espalhamento do ruído de quantização em toda a duração dos blocos de processamento. Quando ocorre uma transição brusca perto do final de um bloco, o modelo psico-acústico pode sobreavaliar o limiar de mascaramento, e o codificador injecta ruído de quantização que se pode revelar audível no início do bloco.

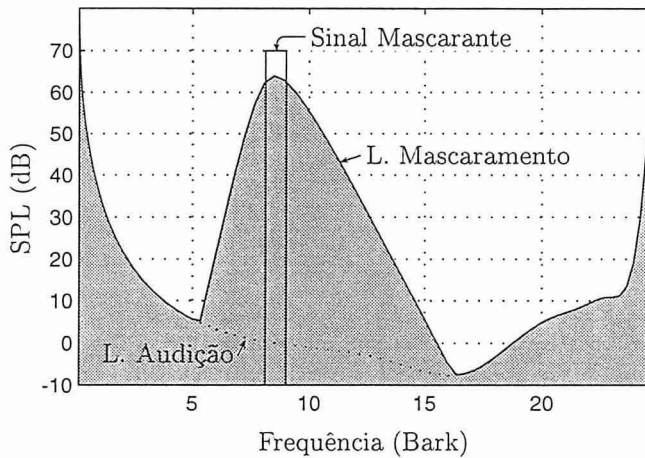


Figura A.3: Mascaramento de um tom por uma banda de ruído.

A forma do limiar de mascaramento resulta do perfil de excitação que o estímulo acústico induz na membrana basilar. Um estímulo sinusoidal de certa frequência, por exemplo, produz uma excitação que cresce desde a janela oval até certo ponto da membrana basilar—o ponto “sintonizado” para essa frequência—e depois decresce rapidamente.³ Há assim uma forma de “espraçamento” espectral do estímulo que resulta numa excitação que diminui rapidamente em direcção ao helicotrema (baixas frequências), e que diminui mais lentamente em direcção à janela oval (altas frequências). A curva de espraçamento na membrana basilar tem uma forma semelhante ao limiar de mascaramento provocado pelo estímulo.

Os fenómenos de mascaramento no domínio da frequência são largamente explorados pelos codificadores perceptuais actuais. Em todos os casos, a estimação do limiar de mascaramento parte sempre de uma análise espectral do sinal, mas a partir daí os métodos diferem. Um método possível para o cálculo do limiar de mascaramento segue os seguintes passos:

1. Cálculo da intensidade crítica em cada frequência por integração do espectro de potência através de uma “janela” rectangular com 1 Bark de largura.
2. Determinação dos picos espectrais.
3. Colocação de curvas de mascaramento sob os picos, a uma “distância” dada pelo índice de mascaramento.
4. Reunião e arredondamento das várias curvas para formar o limiar de mascaramento final.

Este é basicamente o procedimento usado no Modelo Psico-acústico 1 recomendado em [MPEG 92].

Outra abordagem, adoptada originalmente no PXFMM, mas usada também no Modelo Psico-acústico 2, procede de forma diferente:

³As curvas de excitação foram observadas pela primeira vez em 1928 por Georg von Békésy em cócleas de cadáveres humanos, estimuladas por sons de grande intensidade. Estas experiências têm sido repetidas e aperfeiçoadas ao longo dos anos com cócleas humanas e de outros animais [Allen 85].

1. Determinação do perfil de excitação na membrana basilar, por convolução do espectro do estímulo com uma função espraioamento no domínio Bark.
2. Subtração do índice de mascaramento para obter o limiar espraioado.
3. Conversão do limiar espraioado no limiar de mascaramento. Esta conversão consiste na desconvolução com a função de espraioamento. Na prática porém, devido às dificuldades associadas a este cálculo, a operação de desconvolução é substituída por uma operação de renormalização [Johnston 88].

A função de espraioamento, 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, $z = z_2 - z_1$, e é descrita analiticamente pela expressão:

$$S(z) = 15.81 + 7.5(z + 0.474) - 17.5\sqrt{1 + (z + 0.474)^2} + d(z) \text{ dB}, \quad (\text{A.3})$$

onde

$$d(z) = \min\{0, 8(z - 0.5)(z - 2.5)\}$$

é um termo 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.

A.3.3 Índice de Mascaramento

Chama-se *índice de mascaramento* à margem, medida em dB, entre o nível do sinal mascarante e o limiar de mascaramento à mesma frequência. O índice de mascaramento varia em função da frequência mas praticamente não depende da intensidade do sinal mascarante. Também varia significativamente consoante a natureza tonal dos sinais: na situação de tons a mascarar bandas de ruído, o índice é muito maior do que na situação de ruído a mascarar tons. 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 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 [Schroeder 79]:

$$TMN(z) = 15.5 + z.$$

As tabelas de TMN disponíveis na recomendação do MPEG para o Modelo Psico-acústico 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 [Ferreira 92] que esta expressão levava a uma sobrecodificação das altas frequências em detrimento das baixas e foi proposta uma fórmula alternativa para corrigir esse defeito:

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

Não há portanto consenso quanto à curva exacta do índice TMN em função da frequência. Por esta razão, decidimos basear-nos em dados experimentais obtidos num trabalho recente [Sporer 92], e adoptar uma nova formulação:

$$TMN(z) = \begin{cases} 16\frac{6-z}{6} + 28\frac{z}{6} & \text{se } 0 \leq z < 6 \\ 28 & \text{se } 6 \leq z < 16 \\ 28\frac{20-z}{20-16} + 20\frac{z-16}{20-16} & \text{se } 16 \leq z < 20 \\ 20 & \text{se } 20 \leq z \end{cases} \quad (\text{A.4})$$

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).

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:

$$NMT(z) = 5.5.$$

Ferreira usou um modelo ligeiramente diferente [Ferreira 92]:

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

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 [Johnston 88] 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:

$$MI(z) = \alpha TMN(z) + (1 - \alpha)NMT(z). \quad (\text{A.5})$$

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 [Brandenburg 90]. O princípio assumido 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 A.5. Esta medida de tonalidade é aplicada igualmente no Modelo Psico-acústico 2, recomendado em [MPEG 92].

Apêndice B

Ferramentas Desenvolvidas

Todas as ferramentas criadas ao longo deste trabalho foram desenvolvidas como funções em MATLAB, versão 4.0. 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*.

B.1 Bancos de Filtros e Transformadas

B.1.1 Transformadas em Blocos

Todas estas transformações foram implementadas com a FFT, usando os algoritmos apresentados em [Malvar 92].

```
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.

```

B.1.2 Transformadas Sobrepostas

Implementámos a ELT a partir do programa em C dado em [Malvar 92] 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 Z_i 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 Z_f 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 otimização 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

¹Note-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.

```

B.1.3 Sinais Multiresolução

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 multiresolução (Subsecção 3.1.3). Esta representação baseia-se em duas matrizes: uma transporta o sinal multiresolução 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 multiresolução 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.
```

```

%           01..03 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 multiresolução, 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 multiresolução 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.
```

B.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 multiresolução 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 multiresolução 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 [Y, zf] = 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
```

²Aqui 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 HLTTLBL 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, HLTTLBL.

```

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 [Malvar 92, Section 5.4.4].

```

function [spec, ispec, iform, info, Sdelay] = hltsgn(spec, form)
%HLTSGN Check HLT specification, compute additional delays and other info.
%      HLTSGN 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] = HLTSGN(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].

```

³O 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-ótima 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.
```

B.1.5 Análise de Bancos de Filtros

A rotina FBIR permite a medição das respostas impulsivas 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 registra as saídas de forma a reconstruir as respostas impulsivas 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.
```

B.2 Quantização

B.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.
```

⁴Melhores no sentido da minimização do valor esperado de uma medida de distorção dada por uma função par, crescente no ramo positivo, como por exemplo, $f(x - y_k) = (x - y_k)^2$ [Max 60].

B.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.
```

B.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.
```

QPFD 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.
```

B.3 Codificação Aritmética

As rotinas de codificação e decodificação aritmética foram implementadas em C com programas adaptados a partir dos apresentados em [Witten 87], 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.

```

B.4 Modelo Psico-acústico

A função ZWICKER implementa a Equação A.1.

```
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 A.3 com uma ligeira compressão de 5% no eixo dos xx, como usado no Modelo Psico-acústico 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 psico-acústico retro-adaptativo descrito na Secção 4.4 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.
```

B.5 Codificação Perceptual de Áudio

As rotinas seguintes simulam o sistema de codificação perceptual de áudio com adaptação para trás descrito no Capítulo 4.

```
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

- [Akune 92] M. Akune, R. M. Heddle, and K. Akagiri, "Super Bit Mapping: Psychoacoustically optimized digital recording". In *93rd AES-Convention*, Audio Engineering Society, San Francisco, October 1992. preprint 3371.
- [Allen 85] J. B. Allen, "Cochlear modeling", *IEEE Acoustics, Speech, and Signal Processing Magazine*, 3-29, January 1985.
- [Atal 79] B. S. Atal and M. 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.
- [Beerends 92] J. G. Beerends and J. A. Stemerink, "A perceptual audio quality measure based on a psychoacoustic sound representation", *Journal of the Audio Engineering Society*, 40(12):963-978, December 1992.
- [Berger 68] T. Berger, "Rate distortion theory for sources with abstract alphabets and memory", *Inform. Contr.*, 13:254-273, September 1968.
- [Bodin 95] P. 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.
- [Bosi 92] M. Bosi and G. Davidson, "High quality, low-rate audio transform coding for transmission and multimedia applications". In *93rd AES-Convention*, Audio Engineering Society, San Francisco, October 1992. preprint 3365.
- [Brandenburg 88] K. Brandenburg, "High quality sound coding at 2.5 bit/sample". In *84th AES-Convention*, Audio Engineering Society, Paris, March 1988. preprint 2582.
- [Brandenburg 90] K. Brandenburg and J. D. Johnston, "Second generation perceptual audio coding: The hybrid coder". In *88th AES-Convention*, Audio Engineering Society, Montreux, March 1990. preprint 2937.
- [Brandenburg 92] K. Brandenburg and T. Sporer, "'NMR' and 'Masking Flag': Evaluation of quality using perceptual criteria". In *Proceedings of the 11th International Conference of the AES*, Audio Engineering Society, Portland, 1992.
- [Cassereau 85] P. Cassereau, *A New Class of Optimal Unitary Transforms for Image Processing*. Master's thesis, Mass. Inst. Tech., Cambridge, MA, May 1985.
- [Cox 86] R. 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.

- [Crochiere 83] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, N.J., 1983.
- [De 94] A. De and P. Kabal, "Auditory distortion measure for speech coder evaluation—discrimination information approach", *Speech Communication*, 14(3):205–229, June 1994.
- [EBU-88] *Sound Quality Assessment Material: Recordings for Subjective Tests*. European Broadcasting Union, Brussels, April 1988.
- [Ferreira 92] A. J. S. Ferreira, *Codificação Perceptual de Áudio Digital Estereofónico*. Master's thesis, Faculdade de Engenharia da Universidade do Porto, Porto, January 1992.
- [Gersho 83] A. Gersho and V. Cuperman, "Vector quantization: A pattern-matching technique for speech coding", *IEEE Communications Magazine*, 21:15–21, December 1983.
- [Gersho 91] A. Gersho, S. Wang, and K. Zeger, "Vector quantization techniques in speech coding". In S. Furui and M. M. Sondhi, eds., *Advances in Speech Signal Processing*, chapter 2, Marcel Dekker, Inc., New York, 1991.
- [Gopinath 93] R. A. Gopinath and C. S. Burrus, "Theory of modulated filter banks and modulated wavelet tight frames". In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 169–172, 1993.
- [Gray 84] R. M. Gray, "Vector quantization", *IEEE Acoustics, Speech, and Signal Processing Magazine*, 1:4–29, April 1984.
- [Hamming 80] R. W. Hamming, *Coding and Information Theory*. Prentice-Hall, Englewood Cliffs, N.J., 1980.
- [Houtsma 87] 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 supported by Northern Illinois University (NIU) and the Acoustical Society of America (ASA), 1987.
- [Jain 84] 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.
- [Jayant 84] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Prentice-Hall, Englewood Cliffs, N.J., 1984.
- [Johnston 80] J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks". In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 291–294, Denver, CO, April 1980.
- [Johnston 88] J. D. Johnston, "Transform coding of audio signals using perceptual noise criteria", *IEEE Journal on Selected Areas in Communications*, 6(2):314–323, February 1988.
- [Johnston 91] J. D. Johnston and K. Brandenburg, "Wideband coding—perceptual considerations for speech and music". In S. Furui and M. M. Sondhi, eds., *Advances in Speech Signal Processing*, chapter 4, Marcel Dekker, Inc., New York, 1991.
- [Koilpillai 92] R. D. Koilpillai and P. P. Vaidyanathan, "Cosine-modulated FIR filter banks satisfying perfect-reconstruction", *IEEE Transactions on Signal Processing*, 40:770–783, April 1992.
- [Lyon 88] R. F. Lyon and C. Mead, "An analog electronic cochlea", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1119–1134, July 1988.

- [Maitre 88] X. Maitre, "7 kHz audio coding within 64 kbit/s", *IEEE Journal on Selected Areas in Communications*, 6(2):283–298, February 1988.
- [Malvar 89] H. S. Malvar and D. H. Staelin, "The LOT: Transform coding without blocking effects", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(4):553–559, April 1989.
- [Malvar 90a] H. S. Malvar, "Lapped transforms for efficient transform/subband coding", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38:969–978, June 1990.
- [Malvar 90b] H. S. Malvar, "Modulated QMF filter banks with perfect reconstruction", *Electronics Letters*, 26(13):906–907, June 1990.
- [Malvar 92] H. S. Malvar, *Signal Processing with Lapped Transforms*. Artech House, Norwood, MA, 1992.
- [Max 60] J. Max, "Quantizing for minimum distortion", *IRE Transactions on Information Theory*, IT-6:7–12, March 1960.
- [Morse 91] P. M. Morse, *Vibration and Sound*. Acoustical Society of America, fourth edition, 1991.
- [MPEG 92] *ISO/IEC CD 11172: Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s*. ISO/IEC JTC1/SC29/WG11 (MPEG), 1992.
- [MPEG 94] *ISO/IEC CD 13818: Generic Coding of Moving Pictures and Associated Audio Information*. ISO/IEC JTC1/SC29/WG11 (MPEG), 1994.
- [Nayebi 93] K. Nayebi, T. P. Barnwell, III, and M. J. T. Smith, "Nonuniform filter banks: A reconstruction and design theory", *IEEE Transactions on Signal Processing*, 41(3):1114–1127, March 1993.
- [Noll 93] P. Noll, "Wideband speech and audio coding", *IEEE Communications Magazine*, 34–44, November 1993.
- [Nussbaumer 81] H. J. Nussbaumer, "Pseudo-QMF filter bank", *IBM Tech. Disclosure Bull.*, 24:3081–3087, November 1981.
- [Paillard 92] B. Paillard, P. Mabilieu, S. Morissette, and J. Soumagne, "PERCEVAL: Perceptual evaluation of the quality of audio signals", *Journal of the Audio Engineering Society*, 40:21–31, January 1992.
- [Princen 86] J. P. Princen and A. B. Bradley, "Analysis/synthesis filter bank design based on time domain aliasing cancellation", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34:1153–1161, October 1986.
- [Rothweiler 83] J. H. Rothweiler, "Polyphase quadrature filters—a new subband coding technique". In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1280–1283, Boston, MA, March 1983.
- [Schroeder 79] M. R. Schroeder, B. 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.
- [Schroeder 87] 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.

- [Smith 86] M. J. T. Smith and T. 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.
- [Soman 93] A. K. Soman and P. P. Vaidyanathan, "On orthonormal wavelets and paraunitary filter banks", *IEEE Transactions on Signal Processing*, 41:1170–1183, March 1993.
- [Spille 89] J. Spille and E. F. Schröder, "Design of an optimum filterbank for high quality subband audio coding". In *86th AES-Convention*, Audio Engineering Society, Hamburg, March 1989. preprint 2749.
- [Sporer 92] T. Sporer and H. Schröder, "Measuring tone masking noise". In *93rd AES-Convention*, Audio Engineering Society, San Francisco, October 1992. preprint 3349.
- [Sporer 93] T. Sporer and K. Brandenburg, "Constraints of filter banks used for perceptual measurement". In *95th AES-Convention*, Audio Engineering Society, New York, October 1993. preprint 3703.
- [Stuart 94] J. R. Stuart, "Noise: Methods for estimating detectability and threshold", *Journal of the Audio Engineering Society*, 42(3):124–140, March 1994.
- [Tsutsui 92] K. Tsutsui, H. Suzuki, O. Shimoyoshi, M. Sonohara, K. Akagiri, and R. M. Heddle, "ATRAc: Adaptive transform acoustic coding for MiniDisc". In *93rd AES-Convention*, Audio Engineering Society, San Francisco, October 1992. preprint 3456.
- [Vaidyanathan 88] P. P. Vaidyanathan and P. Q. 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.
- [Vaidyanathan 93] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Prentice-Hall, Englewood Cliffs, N.J., 1993.
- [Vargas 93] L. F. Vargas and H. S. Malvar, "ELT-based wavelet coding of high-fidelity audio signals". In ?, pages 124–127, IEEE?, ?, ? 1993.
- [Witten 87] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression", *Communications of the Association for Computing Machinery*, 30(6):520–540, June 1987.
- [Zelinski 77] R. Zelinski and P. Noll, "Adaptive transform coding of speech signals", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(4):299–309, August 1977.
- [Zeng 91] K. Zeng, C.-H. Yang, D.-Y. Wei, and T. R. N. Rao, "Pseudorandom bit generators in stream-cipher cryptography", *Computer*, 24(2):8–17, February 1991.