

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

**PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**CODIFICAÇÃO CELP COM QUANTIZAÇÃO  
VETORIAL DO FILTRO LPC UTILIZANDO  
BUSCA RÁPIDA EM ÁRVORE K-d**

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE  
SANTA CATARINA PARA A OBTENÇÃO DO GRAU DE MESTRE  
EM ENGENHARIA ELÉTRICA.

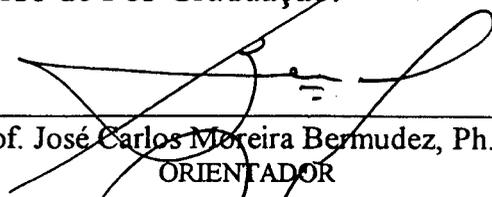
**ALDEBARO BARRETO DA ROCHA KLAUTAU JÚNIOR**

**FLORIANÓPOLIS, NOVEMBRO DE 1993**

**CODIFICAÇÃO CELP COM QUANTIZAÇÃO VETORIAL DO FILTRO LPC  
UTILIZANDO BUSCA RÁPIDA EM ÁRVORE K-d**

**CANDIDATO: ALDEBARO BARRETO DA ROCHA KLAUTAU JÚNIOR**

Esta dissertação foi julgada para obtenção do título de **MESTRE EM ENGENHARIA - ESPECIALIDADE ENGENHARIA ELÉTRICA** e aprovada em sua forma final pelo Curso de Pós Graduação.



---

Prof. José Carlos Moreira Bermudez, Ph. D.  
ORIENTADOR



---

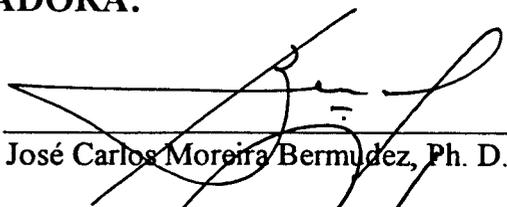
Prof. Rui Seara, Dr. Ing.  
CO-ORIENTADOR



---

Prof. Roberto de Souza Salgado, Ph. D.  
COORDENADOR DO CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**BANCA EXAMINADORA:**



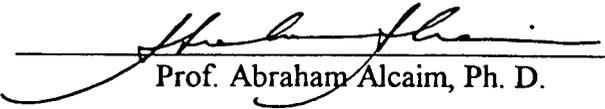
---

Prof. José Carlos Moreira Bermudez, Ph. D. - PRESIDENTE



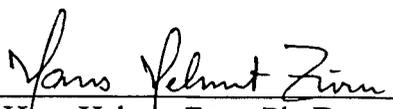
---

Prof. Rui Seara, Dr. Ing.



---

Prof. Abraham Alcaim, Ph. D.



---

Prof. Hans Helmut Zürn, Ph. D.

Uma vez a senhora me disse: - "A vida é luta renhida que aos fracos abates mas aos fortes só faz exaltar..." (GD). Nos momentos em que minhas fraquezas e os infortúnios incomodavam, eu me confortava lembrando o quanto minha tarefa era fácil perante a sua.

Dedico este trabalho à minha avó Eunice Messias.

## A G R A D E C I M E N T O S

Agradeço ao CNPq o apoio financeiro recebido desde a iniciação científica. Agradeço também à UFSC, e particularmente ao LINSE, por toda a infra-estrutura concedida para a realização deste trabalho.

Agradeço aos professores José Carlos Bermudez e Rui Seara pela orientação deste trabalho.

Agradeço aos meus amigos o apoio de todas as horas. As belezas naturais desta região foram realmente suplantadas pela riqueza do material humano com o qual tive o privilégio de conviver durante estes anos. Felizmente, vocês são muitos, e não poderei registrar seus nomes por falta de espaço. Todavia, não poderia deixar de citar as pessoas que constituíram o "lar". Minha gratidão aos amigos Eduardo e Júnior, que foram companheiros irrepreensíveis na difícil tarefa de coabitar. Devo ressaltar também, a ajuda de dois amigos que se assemelham na forma humilde e competente de agir. Meu obrigado ao Márcio Suguieda, "consultor oficial" de informática, e ao Marcus Lamar, o "ombudsman" mais atencioso e participativo que este trabalho poderia ter.

Finalmente, agradeço aos meus pais, irmãos e à minha namorada Polyana. O apoio de vocês foi muito importante para a execução deste trabalho.

# ÍNDICE

LISTA DE FIGURAS.....	vii
LISTA DE TABELAS.....	viii
SIMBOLOGIA.....	x
RESUMO.....	xii
ABSTRACT.....	xiii
<u>CAPÍTULO 1 - INTRODUÇÃO</u> .....	1
<u>CAPÍTULO 2 - CODIFICAÇÃO CELP - REVISÃO BIBLIOGRÁFICA</u> .....	5
2.1 - Introdução.....	5
2.2 - Codificador CELP - Estrutura Básica.....	7
2.3 - Evolução Cronológica do CELP.....	11
2.4 - Conclusões.....	34
<u>CAPÍTULO 3 - QUANTIZAÇÃO VETORIAL DO FILTRO LPC</u> .....	36
3.1 - Introdução.....	36
3.2 - Fundamentos da Quantização Escalar.....	36
3.3 - Fundamentos da Quantização Vetorial.....	39
3.4 - Algoritmos para o Projeto de "Codebooks".....	42
3.4.1 - Considerações sobre o Algoritmo LBG.....	42
3.4.2 - Utilização de Outros Algoritmos.....	44
3.4.2.1 - Algoritmo PNN.....	44
3.4.2.2 - Redes Neurais.....	44
3.4.2.3 - "Simulated Annealing".....	45
3.5 - Quantização Vetorial "Product-code" Generalizada.....	46
3.5.1 - "Product-code".....	46
3.5.2 - Particionamento ou "Split".....	46
3.5.3 - Multi-estágio.....	46
3.5.4 - "Product-code" Generalizada.....	47
3.6 - Medidas de Distorção.....	48
3.7 - Utilização de "Line Spectral Frequencies".....	51
3.8 - Conclusões.....	57
<u>CAPÍTULO 4 - BUSCA RÁPIDA EM ÁRVORE K-DIMENSIONAL</u> .....	59
4.1 - Introdução.....	59
4.2 - Partições de Voronoi.....	59
4.3 - Métodos de Busca Rápida.....	61

4.3.1 - Árvore K-dimensional .....	61
4.3.2 - Outros Métodos de Busca Rápida .....	65
4.4 - Implementação de Árvore K-dimensional .....	67
4.4.1 - Determinação dos Limites das Projeções das Partições de Voronoi ....	69
4.4.2 - Projeto da Árvore K-d Através do Método GOC .....	70
4.5 - Conclusões .....	71

## **CAPÍTULO 5 - AVALIAÇÃO DE MEDIDAS DE DISTORÇÃO PARA A QV**

DAS LSFs .....	72
5.1 - Introdução .....	72
5.2 - Considerações Gerais .....	72
5.3 - Medidas de Ponderação .....	75
5.4 - Estatísticas e Complexidade das Medidas de Ponderação .....	77
5.5 - Avaliação do Efeito da Ponderação .....	81
5.6- Conclusões .....	88

## **CAPÍTULO 6 - CODIFICAÇÃO CELP UTILIZANDO ÁRVORE K-d PARA A**

QV DAS LSFs .....	90
6.1 - Introdução .....	90
6.2 - Avaliação de Métodos para Análise LPC .....	91
6.3 - Resultados Obtidos com o Particionamento .....	92
6.4 - QV das LSFs usando busca rápida em árvore K-d .....	93
6.5 - Avaliação da Voz Sintetizada .....	96
6.6 - Conclusões .....	98

## **CAPÍTULO 7 - CONCLUSÕES**..... 100

## **REFERÊNCIAS BIBLIOGRÁFICAS**..... 104

## LISTA DE FIGURAS

1.1 - Modelo fonte-filtro do processo de produção da voz.....	2
2.1 - Resíduo da predição do PCA .....	6
2.2 - Resíduo da predição conjunta do PCA e do PLA .....	6
2.3 - Diagrama em blocos do Codificador CELP .....	7
2.4- Exemplo de QV (N=4, K=5, b= 2, t=0.4 bits/amostra) .....	10
2.5 - Distribuição dos artigos por assunto.....	11
2.6 - (a) Condições iniciais não-nulas e (b) Representação equivalente.....	12
2.7 - Diagrama em blocos do CELP com a estrutura modificada.....	14
2.8 - Desempenho obtido com "codebooks" esparsos [46].....	16
2.9 - Estrutura múltiplo ganho de [59].....	19
2.10 - Estrutura CMSEC.....	20
2.11 - Exemplo de "codewords" superpostas com deslocamento de uma amostra .....	21
2.12 - Semelhança entre um "codevector" e a memória do PLA.....	24
2.13 - Codificação CELP utilizando um "codebook" adaptativo .....	25
2.14 - Exemplo do sinal de excitação (resíduo do PCA) na transição não-vozeado p/ vozeado .....	26
2.15 - Concepção moderna do CELP .....	27
2.16 - Esquema para diminuição do ruído inter-harmônicas [134].....	30
3.1 - Exemplo de histograma.....	38
3.3 - Complexidade da QV .....	41
3.4- Conceito de equidistância para diferentes normas $L_p$ ([10]).....	49
3.5 - Ilustração gráfica da distorção espectral (SD) [53] .....	51
3.6 - Distribuição das LSFs no plano z para um filtro LPC de ordem 10 .....	56
4.1- Exemplo de diagrama de Voronoi .....	60
4.2 - Exemplo de árvore K-dimensional (adaptada de [171]) .....	63
4.3 - (a) Partições de Voronoi (b) Separação em duas regiões (c) Estrutura da árvore de busca simples .....	63
4.4 - (a) Partições de Voronoi (b) Separação em duas regiões (c) Estrutura da árvore de busca conservativa .....	64
5.1 - Histograma da medida da POTÊNCIA.....	78
5.2 - Histograma da medida da DIFERENÇA .....	79
5.3 - Histograma da medida da SENSIBILIDADE .....	80
6.1 - Comparação entre os algoritmos EOC e GOC.....	95
6.2 - Organização do "codebook" para a utilização de um único nível da árvore .....	96

## LISTA DE TABELAS

2.1 - Parâmetros do codificador CELP original .....	8
2.2- Avaliação da interpolação linear baseada na distorção espectral (SD) [155].....	31
2.3- Características dos codificadores em relação ao PCA .....	32
2.4- Características dos codificadores em relação à excitação .....	33
2.5- Complexidade de cada etapa em MIPS.....	34
3.1 - Exemplo de quantização uniforme.....	37
3.2 - Exemplo de quantização não-uniforme.....	38
3.3 - Exemplo de quantização vetorial.....	40
3.4 - Exemplo da possível diminuição da taxa através da QV.....	40
5.1 - Pesos associados às operações aritméticas [210] .....	73
5.2 - Complexidade do cálculo de um ponto do espectro de potência (Eq. (5.4)) .....	74
5.3 - Estatísticas do vetor de ponderação normalizado p/ a distorção da POTÊNCIA .....	77
5.4 -Estatísticas do vetor de ponderação normalizado p/ a distorção da DIFERENÇA.....	77
5.5 - Estatísticas do vetor de ponderação normalizado para a distorção da SENSIBILI- DADE .....	78
5.6 - Complexidade do cálculo do vetor de ponderação.....	80
5.7 - Desempenho dos "codebooks" projetados c/ diferentes medidas de distorção .....	82
5.8 - Buscas para "codebook" projetado com erro quadrático (CBK_EQ).....	83
5.9 - Buscas para "codebook" projetado com medida da DIFERENÇA (CBK_D).....	83
5.10 - Buscas para "codebook" projetado com medida da POTÊNCIA (CBK_P) .....	84
5.11 - Busca utilizando a medida de distorção da POTÊNCIA-2 .....	85
5.12 - Desempenho da QV por particionamento para CBK_EQ_SPLIT (24 bits) para a codificação através da SD .....	86
5.13 - Busca usando diferentes medidas de distorção p/ o CBK_EQ_SPLIT (24 bits).....	87
5.14 - SD para busca utilizando o erro quadrático .....	88
5.15 - SD para busca utilizando a medida da POTÊNCIA .....	88
5.16 - SD para busca utilizando a medida da DIFERENÇA.....	88
6.1 - Comparação entre métodos de análise LPC para quantização escalar.....	92
6.2 - Quantização escalar da razão log-área dos coeficientes PARCOR.....	92
6.3 - Desempenho da árvore projetada pelo algoritmo EOC (número de "codewords" por nível da árvore).....	94
6.4 - Desempenho da árvore projetada pelo algoritmo GOC (número de "codewords" por nível da árvore).....	94
6.5 - Avaliação da busca em árvore em termos de SD.....	95
6.6 - Distribuição de bits para a excitação.....	97

6.7 - Resultados objetivos (RSRSEG e RSR <sub>w</sub> em dB) util. quantização escalar.....	98
6.8 - Resultados objetivos (RSRSEG e RSR <sub>w</sub> em dB) utilizando QV c/ busca em árvore	
K-d .....	121

## SIMBOLOGIA

$a_i$	i-ésimo coeficiente do filtro PCA
$A(z)$	denominador da função de transferência $H(z)$ do filtro LPC
$B$	coeficiente do filtro PLA
$\bar{b}$	número médio de "codewords" nas folhas para determinado nível da árvore
$b$	número de bits usado pelo quantizador
$c_i(n)$	i-ésima "codeword" do "codebook" fixo
$d(n)$	excitação sintetizada pelo codificador
$d(X, \hat{X})$	distorção entre os vetores $X$ e $\hat{X}$
$E$	duração do sub-quadro de excitação
$e(n)$	resíduo de predição do PCA (excitação ideal)
$e_i(n)$	sinal de erro ponderado
$G$	ganho do "codebook" fixo
$j$	eixo ortogonal ao hiperplano divisor da árvore K-dimensional
$h$	interseção do hiperplano divisor com seu respectivo eixo ortogonal
$h(n)$	resposta ao impulso do filtro PCA <sub>w</sub>
$H$	matriz obtida a partir do truncamento da resposta ao impulso $h(n)$ do filtro PCA <sub>w</sub>
$H(z)$	função de transferência do filtro PCA
$H_w(z)$	função de transferência do filtro PCA <sub>w</sub>
$K$	dimensão do vetor ou do espaço
$M$	atraso do PLA
$MAC$	operação de multiplicação e acumulação, geralmente realizada em um único ciclo de clock.
$MFLOPS$	milhões de operações em ponto flutuante por segundo
$MIPS$	milhões de instruções por segundo
$N$	número de "codewords" do "codebook" fixo
$P$	ordem do filtro LPC (filtro PCA no contexto do CELP).
$PCA$	Filtro Preditor de Curto Atraso.
$PCA_w$	Combinação do filtro PCA e $W(z)$
$PLA$	Filtro Preditor de Longo Atraso.
$P(z)$	função de transferência do filtro PLA
$Q_E^b\{X\}$	quantização escalar da variável $X$ utilizando $b$ bits
$r(n)$	resíduo da predição do conjunto PCA / PLA
$R(m)$	m-ésima amostra da função autocorrelação
$s(n)$	sinal de voz original
$\hat{s}(n)$	sinal de voz sintetizado
$s'(n)$	sinal alvo da primeira etapa de determinação da excitação no CELP

$s''(n)$	sinal alvo da segunda etapa de determinação da excitação no CELP
$S(w)$	espectro de potência do filtro LPC
SD	distorção espectral
t	taxa de bits
T	número de coeficientes do filtro PLA
X	vetor a ser quantizado
$\hat{X}$	vetor quantizado
$W(z)$	função de transferência do filtro ponderador do erro
$w_i$	i-ésima LSF ("Line Spectral Frequency") em radianos
$\gamma$	fator de ponderação do erro
$\delta$	fator de expansão da banda do filtro PCA
$\sigma$	desvio padrão

## RESUMO

O algoritmo de codificação CELP é sem dúvida alguma o mais utilizado atualmente em sistemas de telecomunicações que exigem um sinal de voz sintetizado de boa qualidade e baixas taxas de bits.

Abordaremos inicialmente, de forma geral, alguns aspectos importantes para a implementação dos algoritmos de codificação CELP. Contudo, concentramos nossos esforços na quantização do filtro LPC. Este assunto é objeto de grande interesse atualmente devido à possibilidade de redução da taxa de bits com o uso da quantização vetorial.

Nossa proposta neste trabalho é reduzir a complexidade computacional advinda da quantização vetorial do filtro LPC em codificadores CELP, utilizando uma estrutura de busca rápida baseada em uma árvore K-dimensional. Esta estrutura consiste em uma estratégia de busca que visa diminuir o número de "codewords" a serem pesquisadas na etapa de busca da melhor "codeword".

Serão abordadas as características peculiares que devem ser levadas em conta no projeto da árvore K-dimensional para a quantização vetorial das LSFs utilizando a estrutura com particionamento. Serão também estudadas as influências da ponderação do erro quadrático no processo de busca em árvore, em termos de complexidade e distorção.

Finalmente, serão apresentados os resultados obtidos com a implementação da busca rápida em árvore K-dimensional. Seguindo o mesmo procedimento adotado na literatura, avaliaremos inicialmente, o efeito da quantização do filtro LPC de forma local através da estatística da distorção espectral e, posteriormente, o efeito global da quantização no sinal de voz sintetizado pelo codificador CELP.

## ABSTRACT

CELP coding algorithm is quite sure the most used algorithm in telecommunication systems requiring synthesized speech signal with good quality and low bit rate.

In this work we first have studied the whole algorithm, and afterwards our effort concentrated on the quantization of the LPC filter. This subject has been object of great interest today due to the possibility of reducing the bit rate by using vector quantization.

Our aim is to reduce the computational complexity in the vector quantization of the LPC filter by using a K-dimensional tree structure to implement a fast search. Such a structure consists of a strategy of search which allows to reduce the number of codewords to be scanned in order to find the best codeword.

We have evaluated peculiar characteristics which must be taken into account when designing the K-dimensional tree for the vector quantization of the LSF's using the structure with splitting. Studies have been made on the influence of weighting the squared error in the process.

We have presented results obtained from the implementation of the fast search using a K-dimensional tree, and following common practice, we have evaluated the vector quantization itself through spectral distortion statistics. We also have evaluated the quantization in terms of the synthesized speech signal.

# CAPÍTULO 1

## INTRODUÇÃO

Codificação de voz pode ser entendida como um conjunto de técnicas para a representação eficiente do sinal de voz. O conceito de eficiência depende do objetivo em vista, o qual pode ser a redução da taxa de transmissão, a qualidade do sinal sintetizado, a redução da memória de armazenamento, etc. Aplicações como reconhecimento de voz, síntese de voz, criptologia e outras, geralmente utilizam técnicas de codificação na etapa de pré-processamento do sinal. Neste trabalho, a codificação é abordada no âmbito da transmissão de voz em um sistema de telecomunicações. Nosso principal objetivo é reduzir a taxa de bits necessária para representar a voz, para um especificado nível de qualidade do sinal de voz após a decodificação. Neste contexto, a complexidade computacional também é um fator importante. Quanto mais simples o codificador, menor o custo do "hardware" necessário para sua implementação.

A representação digital mais elementar de um sinal de voz é obtida a partir da simples digitalização das amostras do mesmo. A codificação PCM ("Pulse Code Modulation") [11] implementa este princípio. Esta técnica de codificação serve de ponto de partida para outras técnicas mais elaboradas.

Há alguns anos atrás, as formas usuais de representação do sinal de voz eram classificadas em dois grandes grupos: as representações por forma de onda e as representações paramétricas [11]. Os próprios termos codificadores da forma de onda e codificadores paramétricos indicam a filosofia de cada uma destas classes de codificadores. A representação por forma de onda busca obter a maior semelhança possível entre a forma de onda do sinal sintetizado pelo codificador e a forma de onda do sinal original. Os codificadores da forma de onda alcançam boa qualidade de voz com reduzida complexidade. Todavia, a qualidade da voz sintetizada sofre uma degradação substancial para baixas taxas. Por isto, os codificadores da forma de onda operam geralmente com taxas acima de 16 kbits/s.

A representação paramétrica extrai parâmetros representativos do sinal original. Nesta classe, geralmente procura-se modelar o processo físico de produção da fala humana. O codificador paramétrico mais difundido é o vocoder LPC ("Linear Predictive Coding") [93]. Com mais rigor, o termo vocoder ("voice coder") designaria qualquer codificador de voz. Entretanto, este termo é geralmente associado aos codificadores que utilizam um modelo simplificado fonte-filtro para simular o processo de produção da fala [11]. Neste modelo fonte-filtro, considera-se que o sinal de voz pode ser obtido como a saída de um filtro digital variante no tempo. Para o vocoder LPC, este filtro é excitado por um trem de impulsos para a obtenção de sons vozeados e por ruído para a obtenção de sons não-vozeados (Fig. 1.1).

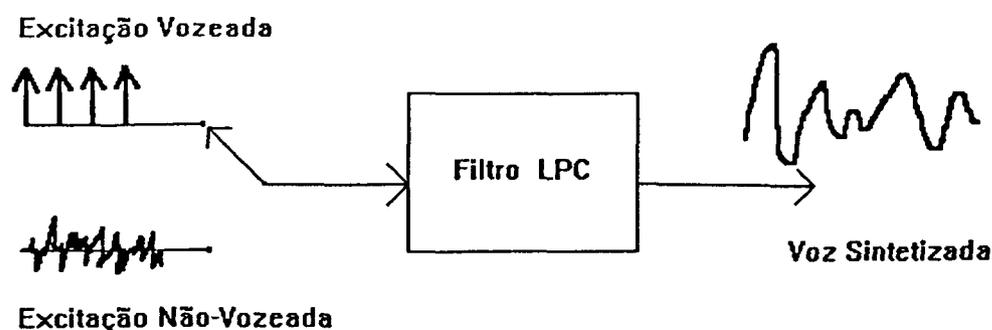


Fig. 1.1 - Modelo fonte-filtro do processo de produção da voz

Em baixas taxas de bits, os codificadores paramétricos possibilitam uma melhor qualidade de voz do que os codificadores da forma de onda. O vocoder LPC, por exemplo, é capaz de garantir boa inteligibilidade da voz a taxas menores do que 2,4 kbits/s [75], [133], [148], [153], [166], [169]. Contudo, a qualidade subjetiva da voz gerada pelo vocoder LPC é inferior à desejada para certas aplicações como, por exemplo, telefonia. O motivo disto é o modelo extremamente simplificado do sinal que excita o filtro LPC.

O desenvolvimento acelerado das técnicas de codificação de voz nas décadas de 80 e 90 estabeleceu uma gama bastante variada de codificadores. Muitos destes, dentre os quais encontra-se o codificador CELP ("Code-Excited Linear Predictive"), apresentam características tanto de codificadores da forma de onda quanto de codificadores paramétricos. Com isto, a separação dicotômica em codificação da forma de onda e paramétrica [11] mostra-se deficiente. Atualmente torna-se necessária uma classificação mais pormenorizada dos codificadores de voz [93], [118]. Todavia, os conceitos de codificação da forma de onda e codificação paramétrica são muito úteis ao nosso propósito de estudar o codificador CELP.

A taxa de bits, a complexidade computacional e a qualidade da voz sintetizada são fatores interdependentes que determinam a eficiência da codificação através de uma relação de compromisso. A quantização vetorial é o caminho natural para melhorarmos o desempenho de um codificador no tocante à taxa de bits e qualidade da voz sintetizada (distorção). Por isso, neste trabalho dedicaremos especial atenção à quantização vetorial do filtro LPC. Para representar o filtro LPC utilizamos as LSFs ("Line Spectral Frequencies"). A estrutura de quantização vetorial adotada foi o particionamento.

Em relação à tríade taxa de bits, complexidade computacional e qualidade da voz sintetizada, a quantização vetorial só apresenta restrições devido ao aumento da complexidade computacional. Todavia, quando a redução da taxa de bits torna-se imperativa, um aumento tolerável da complexidade computacional pode ser aceito. A próxima geração dos codificadores de voz para o sistema de comunicações móveis norte-americano, por exemplo, está sendo especificada de forma que seja capaz de proporcionar uma voz sintetizada de boa qualidade a partir de 6,5 kbits/s, dos quais

aproximadamente 2,5 kbits/s devem ser dedicados ao controle de erros no canal [208]. Neste contexto, a quantização vetorial do filtro LPC é de fundamental importância.

Neste trabalho, nossa proposta é reduzir a complexidade computacional advinda da quantização vetorial do filtro LPC em codificadores CELP, utilizando uma estrutura de busca rápida baseada em uma árvore K-dimensional. Esta estrutura consiste em uma estratégia de busca que visa diminuir o número de "codewords" a serem pesquisadas na etapa de busca da melhor "codeword". Abordaremos as características peculiares que devem ser levadas em conta no projeto da árvore K-dimensional para a quantização vetorial das LSFs utilizando a estrutura com particionamento. Serão estudadas as influências da ponderação do erro quadrático no processo de busca em árvore, em termos de complexidade e distorção.

Este trabalho encontra-se estruturado da forma que descrevemos a seguir. No Capítulo 2 tentaremos organizar de maneira uniforme as informações dispersas na vasta bibliografia referente aos codificadores CELP. Devido à complexidade de seu algoritmo, o CELP só pode ser compreendido através do estudo de diversas técnicas que se encontram embutidas no algoritmo de codificação. Devido à grande variedade destas técnicas, achamos que a maneira mais conveniente de apresentá-las, seria através da evolução cronológica da codificação CELP.

O Capítulo 3 abordará o emprego da quantização vetorial dos parâmetros do filtro LPC na codificação CELP. Apresentaremos a técnica de quantização vetorial como uma generalização da quantização escalar não-uniforme, de maneira que o leitor não familiarizado com a quantização vetorial possa absorvê-la de forma natural. Revisaremos as principais técnicas de estruturação do quantizador vetorial. Apresentaremos de forma sucinta uma revisão dos métodos de projeto de "codebook", dando ênfase ao algoritmo LBG. Julgamos conveniente incorporar também, uma formulação proposta recentemente [173], [175], que consiste em uma generalização da estrutura "product-code". O conceito de quantização vetorial "product-code" generalizada permite-nos uma visão unificada das principais técnicas de estruturação do "codebook". Neste mesmo capítulo, serão discutidas as propriedades gerais das medidas de distorção utilizadas em quantização vetorial. Nosso interesse é fornecer as informações necessárias para o entendimento das peculiaridades das medidas de distorção aplicadas às LSFs, e suas conseqüências no projeto da busca rápida em árvore K-dimensional. Apresentaremos também o conceito e as principais propriedades das LSFs.

No Capítulo 4 será apresentada a proposta de utilização da árvore K-dimensional no intuito de diminuir a complexidade computacional, sem incorrer em degradação significativa do desempenho da quantização vetorial do filtro LPC. Inicialmente, estudaremos as propriedades geométricas dos quantizadores vetoriais seguido do conceito de partições de Voronoi. Posteriormente, definiremos os requisitos necessários para a busca em árvore não acarretar em degradação do desempenho, conceituando a busca simples e a busca conservativa em árvore K-dimensional. Abordaremos a implementação da árvore K-dimensional e seu projeto. Realizaremos também, uma breve revisão dos outros métodos de busca rápida existentes.

O Capítulo 5 será dedicado ao estudo da ponderação do erro quadrático utilizada em sistemas de quantização vetorial das LSFs. Para que pudéssemos comparar cada uma das medidas entre si, normalizaremos os vetores de ponderação. As estatísticas dos vetores de ponderação normalizados, possibilitam-nos maior compreensão das características destas ponderações. Serão apresentados os resultados - em termos de distorção e complexidade - obtidos com cada uma das medidas de ponderação do erro quadrático. Realizaremos também, uma discussão acerca do uso da distorção espectral como medida objetiva para avaliação.

No Capítulo 6 apresentaremos inicialmente os resultados obtidos com o particionamento das LSFs. Estes resultados servem de referência para comparação com os resultados obtidos com a busca em árvore K-dimensional. Serão apresentados também os resultados obtidos com a incorporação dos quantizadores vetoriais projetados ao codificador CELP de referência, comparando-os com alguns sistemas que empregam a quantização escalar.

Este trabalho só pôde ser realizado devido a estarmos partindo de resultados já obtidos por muitos outros pesquisadores que se empenharam em desenvolver trabalhos na área de codificação de voz. A concorrência mundial, estimulada pelo grande interesse comercial neste assunto, torna árduas as pesquisas e muitas vezes as informações não são divulgadas com a transparência desejada. Devido a isto, devemos ressaltar que tivemos a fortuna de ter este trabalho sedimentado em outros, particularmente no "know-how" em codificação CELP desenvolvido a partir da dissertação de mestrado do Prof. Dênis Fernandes [176], que forneceu apoio técnico e os programas por ele desenvolvidos. Desta forma, não abordaremos neste trabalho assuntos como análise LPC, determinação dos filtros PCA e PLA, e outros, pois os mesmos são exauridos na referida dissertação e em outras, desenvolvidas no LINSE / EEL / UFSC [125], [149] e em laboratórios de pesquisa de outras instituições [148], [153], [208].

## CAPÍTULO 2

### CODIFICAÇÃO CELP - REVISÃO BIBLIOGRÁFICA

#### 2.1- INTRODUÇÃO

Os codificadores CELP ("Code-Excited Linear Predictive") [33] pertencem a uma família de codificadores capazes de gerar sinais de voz de boa qualidade a baixas taxas (entre 4 e 16 kbits/s) [68]. Outros codificadores desta família são o Multi-Pulso [32], [125], o Multi-Pulso Regular [49], [50] e o Auto-Excitado [47]. Estes codificadores têm em comum a busca de uma melhor representação da excitação na modelagem fonte-filtro, através da técnica conhecida como análise-por-síntese [42], [68]. Nesta técnica, o codificador realiza sucessivas sínteses, a fim de quantizar a excitação a partir da comparação entre o sinal de voz original e o sinal sintetizado pelo codificador. Por este motivo, talvez o termo codificação-por-síntese fosse mais adequado [56]. Os codificadores desta família baseiam-se no modelo fonte-filtro de produção da voz - característica da codificação paramétrica, e também buscam a semelhança entre as formas de onda dos sinais de voz original e sintetizado - característica da codificação da forma de onda. O compartilhamento destas duas características motiva a classificação destes codificadores como híbridos [93], [128], [182].

Os codificadores híbridos freqüentemente utilizam um segundo filtro denominado filtro Preditor de Longo Atraso (PLA). O objetivo do filtro PLA é explorar a correlação existente entre amostras do sinal de voz separadas por um período de pitch (período fundamental ou tonal) [2], [21], [63], [91]. Neste contexto, o filtro LPC da Fig. 1.1 passa a ser chamado de filtro PCA (Preditor de Curto Atraso). Isto porque o filtro PCA explora a correlação existente entre amostras do sinal de voz separadas por um curto intervalo de tempo.

A ação conjunta dos filtros PCA e PLA, por si só, é capaz de gerar voz sintetizada com qualidade razoável. Esta é a idéia básica do codificador Auto-Excitado [47], [61]. Contudo, quando desejamos alcançar melhor qualidade, devemos buscar formas de representar o resíduo da ação conjunta dos filtros PCA e PLA. Este resíduo corresponde à parcela do sinal de voz que os filtros preditores não conseguiram estimar.

O resíduo da predição do PCA é o sinal de excitação ideal  $e(n)$  indicado na Fig. 2.1. Caso  $e(n)$  fosse aplicado à entrada do filtro PCA, o sinal de saída do filtro seria o próprio sinal de voz original. No vocoder LPC  $e(n)$  é representado grosseiramente através de um trem de impulsos ou de um ruído, conforme o tipo de som a ser sintetizado. Outros codificadores buscam uma representação mais precisa do resíduo  $e(n)$ .

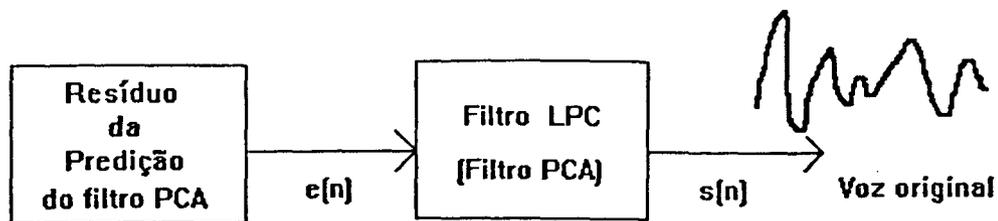


Fig. 2.1 - Resíduo da predição do filtro PCA

Os codificadores de voz que utilizam um segundo filtro preditor (PLA), buscam principalmente extrair a correlação existente entre as amostras do sinal de voz separadas por um período de pitch. No caso de sons não-vozeados, onde não ocorrem pulsos relativos ao período de pitch, o PLA torna-se menos eficiente. O resíduo da predição conjunta dos filtros PCA e PLA,  $r(n)$ , é o sinal que aplicado à entrada do filtro PLA reproduz na saída do filtro PCA o sinal de voz original (Fig. 2.2). No caso ideal, quando toda a correlação existente entre as amostras do sinal de voz tenha sido extraída pelos filtros preditores, o sinal  $r(n)$  seria um ruído branco. Os codificadores da família do CELP que utilizam o PLA e o PCA, buscam representar de forma eficiente o resíduo da predição conjunta destes filtros [68]. Uma característica que freqüentemente diferencia codificadores desta família é justamente a modelagem deste resíduo.

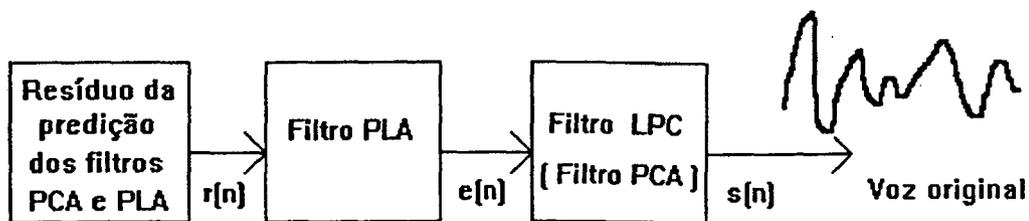


Fig. 2.2 - Resíduo da predição conjunta dos filtros PCA e PLA

O codificador Multi-Pulso busca representar o resíduo  $r(n)$  através da utilização de pulsos de diferentes amplitudes localizados estrategicamente no tempo. O codificador Multi-Pulso Regular restringe a ocorrência destes pulsos a intervalos regulares de tempo. O Auto-Excitado utiliza amostras passadas da própria excitação, através do uso de um filtro PLA adicional. O CELP permite a escolha de um vetor ("codeword") para representar o resíduo  $r(n)$ . Este vetor é selecionado dentre um conjunto pré-estabelecido conhecido como dicionário ou "codebook".

## 2.2- CODIFICADOR CELP - ESTRUTURA BÁSICA

O princípio básico de funcionamento do CELP pode ser descrito através do diagrama em blocos da Fig. 2.3. Cada vetor ("codeword")  $c_i(n)$  é escalado pelo ganho  $G$ , e filtrado pela cascata de filtros formada pelo PLA e PCA, gerando a voz sintetizada  $\hat{s}(n)$ . O sinal  $\hat{s}(n)$  é comparado com o sinal original  $s(n)$  a ser codificado, calculando-se o erro entre estes dois sinais. Este erro é ponderado pelo filtro  $W(z)$  gerando o sinal  $e_i(n)$ . O objetivo do filtro ponderador  $W(z)$  é aumentar a qualidade subjetiva do sinal sintetizado [12]. O mesmo procedimento é repetido para todas as  $N$  "codewords" do dicionário ("codebook"). A melhor "codeword"  $c_i(n)$  será aquela que conduzir ao sinal  $e_i(n)$  com a menor energia.

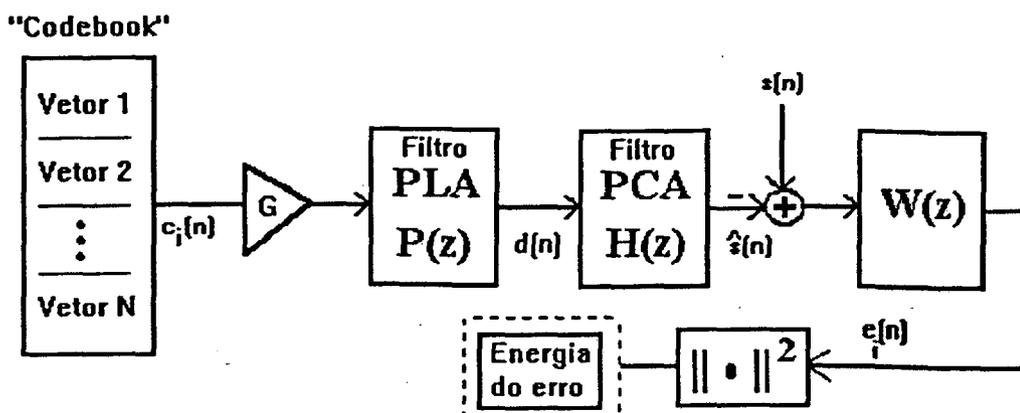


Fig. 2.3 - Diagrama em blocos do Codificador CELP

Vale ressaltar que alguns autores denominam de excitação apenas a contribuição do "codebook", ou seja, a "codeword" escalada  $Gc_i(n)$ . Neste trabalho estaremos considerando que a excitação sintetizada é formada pela contribuição do "codebook" aliada à ação do PLA, ou seja, o sinal  $d(n)$  (Fig. 2.3). Caso a excitação sintetizada  $d(n)$  (Fig. 2.3) correspondesse exatamente ao resíduo de predição  $e(n)$  (Fig. 2.2), o sinal de voz sintetizado  $\hat{s}(n)$  seria idêntico ao sinal de voz original  $s(n)$ . Desta forma podemos interpretar a codificação CELP como uma evolução na modelagem da excitação do vocoder LPC. Além disso, o atual tratamento do PLA como um "codebook" adaptativo [74] (que será abordado oportunamente) reforça esta idéia.

A codificação CELP (assim como outras) baseia-se no tratamento de blocos de amostras, conhecidos como quadros. Em um período de tempo (quadro) suficientemente curto, o sinal de voz não apresenta variação significativa em suas estatísticas, podendo ser considerado estacionário dentro deste período. Cada quadro do sinal de voz é multiplicado por uma janela [11] com duração de aproximadamente 30 ms [176]. Tipicamente, é utilizada a janela de Hamming [176]. O PCA é calculado a partir do sinal de voz passado através de uma janela usando as técnicas de predição linear, sendo atualizado a cada 20 ms. O filtro PCA é responsável pela envoltória espectral do sinal

de voz. A função do filtro PCA é associada à função desempenhada pelo trato vocal humano [11]. Por isto, o intervalo de atualização do PCA é coerente com a velocidade de mudança da anatomia dos elementos do trato vocal. As atualizações do PLA, da "codeword"  $c_i(n)$  e de seu ganho  $G$  são efetuadas a intervalos de 5 ms, denominados de subquadros. Este intervalo é menor do que o utilizado para a atualização do PCA porque no sistema fonador humano as características da excitação mudam mais rapidamente do que se modifica a configuração do trato vocal.

As funções de transferência dos filtros Preditor de Curto Atraso  $H(z)$ ,  $W(z)$  e Preditor de Longo Atraso  $P(z)$  são dadas por [33], [176]:

$$H(z) = \frac{1}{1 - \sum_{i=1}^P a_i z^{-i}} \quad (2.1)$$

$$W(z) = \frac{1 - \sum_{i=1}^P a_i z^{-i}}{1 - \sum_{i=1}^P a_i \gamma^i z^{-i}} \quad (2.2)$$

$$P(z) = \frac{1}{1 + Bz^{-M}} \quad (2.3)$$

Os parâmetros apresentados na Tabela 2.1 foram utilizados no primeiro codificador CELP apresentado em [33]. Este conjunto de parâmetros pode, ainda hoje, ser considerado típico de um codificador CELP. Ressaltamos que durante este trabalho estaremos considerando sinais de voz amostrados a uma taxa de 8 kHz.

Tabela 2.1 - Parâmetros do codificador CELP original.

Ordem do filtro PCA (P)	10
Fator de Ponderação do Erro ( $\gamma$ )	0.9
Fator de expansão da banda do filtro PCA ( $\delta$ )	0.994
Duração do subquadro da excitação (E)	5 ms
Duração do quadro (atualização do PCA)	20 ms
Janela utilizada	Hamming
Duração da janela (análise LPC)	30 ms
Número de coeficientes do filtro PLA (T)	1
Faixa do atraso do PLA (M)	20 a 147
Número de "codewords" do "codebook" fixo (N)	1024

A análise LPC fornece os coeficientes  $a_i$  do filtro PCA. A operação de expansão da largura de banda deste filtro é realizada substituindo-se cada coeficiente  $a_i$  por  $a_i \delta^i$ ,  $\delta < 1$ . Isto equivale a deslocar radialmente os pólos de  $H(z)$  em direção à origem no plano  $z$  [199]. O fator  $\delta = 0.994$  corresponde a uma expansão de cerca de 15 Hz na largura de banda [176]. Calcula-se o denominador do filtro  $W(z)$  usando-se o mesmo conceito de expansão de banda (equação (2)). Neste caso, o fator de expansão  $\gamma$  é chamado de fator ponderador do erro.

O CELP pode ser visto basicamente como o uso conjunto de cinco técnicas importantes: a predição linear [6], a predição de longo atraso [2], a ponderação subjetiva do erro [12], a quantização vetorial [29] e a análise-por-síntese [68]. A integração de cada uma destas técnicas à concepção deste codificador, pode ser acompanhada através da evolução dos trabalhos de Bishnu Atal e de outros pesquisadores [2], [4], [12], [20], [21], [35], até sua formulação definitiva em [33].

A técnica de predição linear é utilizada não só em codificação de voz, mas em diversas outras áreas [6]. Várias bibliografias dedicam atenção especial à predição linear [11], [93], [118]. Por este motivo, não nos deteremos em rerepresentar esta técnica. Um tratamento bastante resumido e didático deste assunto pode ser encontrado nos artigos de L. Rocha [13].

A predição de longo atraso e a ponderação subjetiva do erro são abordadas nos trabalhos de Atal citados acima. Em [116] e [176] podem ser encontradas descrições pormenorizadas destas técnicas em língua portuguesa.

A Quantização Vetorial (QV) vem ganhando cada vez mais importância em sistemas de processamento digital de sinais. Diversos livros e artigos abordam técnicas de QV. Os artigos de revisão [16], [27] e [29] são bons exemplos. Em português temos [115] e as dissertações [153], [148], entre outros.

A idéia básica da QV consiste em aproximar um vetor  $X$  de dimensão  $K$ , por um dos  $N$  vetores  $X_i$  ("codewords") de uma coleção de vetores, chamada "codebook". A busca da melhor "codeword"  $X_i$  é baseada em uma medida de distorção pré-definida [14], [24], [83]. Percorre-se o "codebook" calculando-se uma medida de distorção  $d_i$  entre o vetor  $X$  e cada uma das  $N$  "codewords"  $X_i$ . A "codeword"  $X_i = \hat{X}$  que conduzir à menor distorção  $d$  é então escolhida. A distorção  $d$  entre o vetor  $X$  e a "codeword"  $\hat{X}$  é o erro introduzido pela QV. A taxa de bits  $t$  é dada por  $t = b/K$ , onde  $b = \log_2 N$  é o número de bits necessários para representar os endereços das  $N$  "codewords".

A eficiência da QV reside no fato de que, mantendo cópias do codebook no transmissor e no receptor, basta transmitir o índice da "codeword" escolhida, ao invés de transmitir o vetor. No exemplo da Fig. 2.4, basta o transmissor enviar o índice "01". O receptor buscará em sua cópia do "codebook", quais as 5 amostras correspondentes.

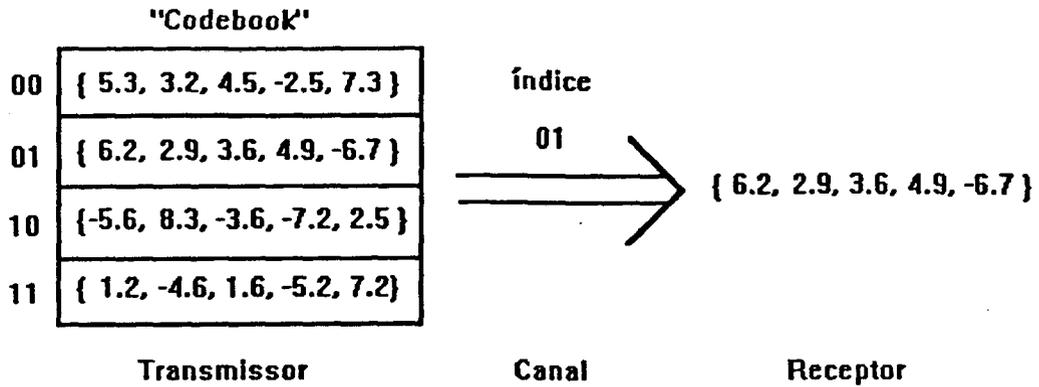


Fig. 2.4- Exemplo de QV ( $N=4$ ,  $K=5$ ,  $b=2$ ,  $t=0.4$  bits/amostra).

Questões como o projeto do "codebook" e técnicas de redução da complexidade computacional são de crucial importância na utilização de QV. Estes assuntos receberão atenção especial nos próximos capítulos. O objetivo desta explanação introdutória é dar a informação necessária para o entendimento da QV no contexto da determinação da excitação no CELP. A QV possui uma característica especial no CELP, pois está inserida em um processo de análise-por-síntese.

Neste ponto é importante fazermos uma observação quanto à necessidade do processo de análise-por-síntese na codificação CELP. Seria bem menos complexo computacionalmente fazer diretamente a QV do resíduo de predição do PLA e PCA [43], [41]. Para isto, bastaria calcularmos o resíduo  $r(n)$  (Fig. 2.2) e determinarmos a "codeword" mais semelhante e seu respectivo ganho. Assim, estaríamos evitando os cálculos requeridos pelo processo de filtragem de cada "codeword".

Em [34] é feita a consideração de que o resíduo  $r(n)$  de predição do PCA e do PLA poderia ser aproximado por um ruído branco. Um sinal com características de ruído branco, ao ser filtrado por um sistema linear com resposta em frequência  $H(z)$ , passa a ter o mesmo espectro  $H(z)$  multiplicado por um fator de escala. No caso de um ruído branco então, minimizar a energia do sinal na entrada do filtro, implicaria na minimização da energia do sinal de erro na saída deste filtro. Assim, a busca da melhor "codeword" em [34] é feita através da quantização vetorial direta de  $r(n)$ , sem utilizar o processo de análise-por-síntese. Como mencionamos, isto representa uma grande redução no esforço computacional. Entretanto, esta técnica conduz a uma degradação substancial na qualidade da voz sintetizada [102]. O resíduo  $r(n)$  da predição conjunta dos filtros PCA e PLA só poderia ser aproximado por um ruído branco caso estes filtros conseguissem extrair toda correlação existente no sinal de voz. Este fato não ocorre na prática.

Sob o ponto de vista da quantização de  $r(n)$ , podemos argumentar também que nos casos em que a operação de quantização é feita com boa precisão, o ruído de quantização resultante realmente aproxima-se de um ruído branco [38]. Isto acontece quando se dispõe de um grande número de bits para a quantização. Quando a quantização é feita com poucos bits (baixas taxas), o ruído de quantização resultante não mais apresenta características de um ruído branco, com seu espectro não

mais sendo plano. Isto acontece quando se dispõe de poucos bits para a quantização. Daí a importância da utilização de sínteses sucessivas na quantização a baixas taxas de bits.

### 2.3 - EVOLUÇÃO CRONOLÓGICA DO CELP

O número de publicações abordando o CELP é significativo. A partir dos trabalhos por nós consultados, realizamos uma estatística baseada nos objetivos perseguidos pelos autores. A Fig. 2.5 mostra a distribuição dos artigos de acordo com algumas classes de objetivos pré-estabelecidas. Podemos observar que a maioria dos artigos preocupam-se com a redução da complexidade do codificador CELP.

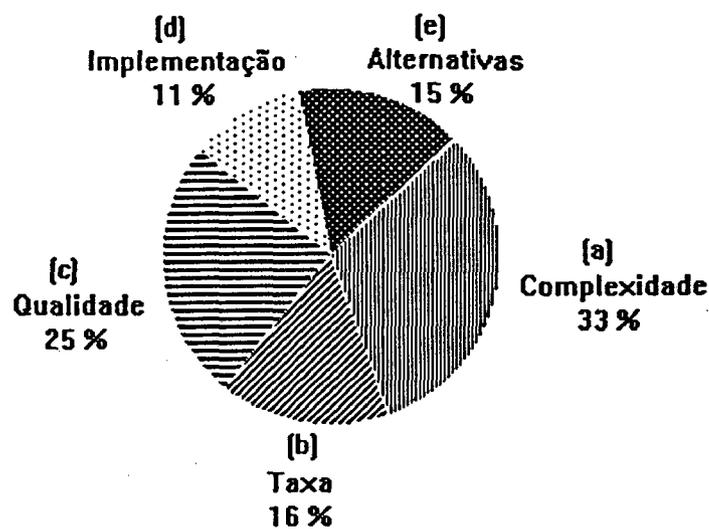


Fig. 2.5 - Distribuição dos artigos por assunto.

(a) Redução de complexidade (inclui redução na memória para armazenamento); (b) Diminuição da taxa (inclui QV do filtro LPC); (c) Avaliação e/ou melhoria da qualidade (inclui alocação dinâmica dos bits e otimização conjunta); (d) Implementação em tempo real (inclui controle do erro de canal); (e) Estruturas alternativas (inclui operação à taxa variável).

Nesta revisão bibliográfica, procuramos apresentar a evolução cronológica das pesquisas em relação aos codificadores CELP. Obviamente, não seria viável considerarmos todas as propostas de melhoramentos. Nosso objetivo é enfatizar os trabalhos que consideramos mais relevantes. Dentro do possível, buscamos situar cada trabalho de acordo com os objetivos apresentados no quadro da Fig. 2.5. Procuramos não subdividir demasiadamente os assuntos para não prejudicarmos a continuidade. Acreditamos que esta metodologia permite apresentar de forma mais clara as soluções já estabelecidas e delinear os avanços futuros.

O CELP original [33], apresentado em 1985, tinha como principal problema a complexidade computacional. Estimativas desta complexidade apresentavam os números de 500 milhões de

operações de multiplicação e acumulação (MAC) por segundo [48] ou 440 milhões de operações em ponto flutuante por segundo (MFLOPS) [46]. Em 1986, os DSPs ("Digital Signal Processors") disponíveis apresentavam capacidade de realizar aproximadamente 4 milhões de MACs por segundo. Isto mantinha o CELP original muito longe de implementações em tempo real. A maioria das pesquisas voltaram-se então para a redução desta complexidade.

Em 1986 foram apresentadas técnicas que reduziram a complexidade do CELP de 400 MFLOPS para algo em torno de 25 a 40 MFLOPS. Os artigos [46] e [48] representam basicamente esta evolução. Ambos baseiam-se em uma modelagem matricial do algoritmo do CELP. O desenvolvimento conceitual do algoritmo, com a realização das filtrações através de equações de recorrência, dificulta as manipulações matemáticas. A modelagem matricial facilitou as manipulações matemáticas. Esta notação matricial só foi possível devido a mudanças na estrutura básica do CELP original (Fig. 2.3).

Existem diversas maneiras de reestruturar o algoritmo do CELP. Duas filosofias são compartilhadas pela quase totalidade das estruturas existentes. Estas consistem na combinação do filtro  $W(z)$  com o filtro PCA, formando o filtro  $PCA_w$ , e no cômputo em separado do efeito da memória dos filtros.

O tratamento da memória dos filtros é extremamente relevante para a qualidade da voz sintetizada [176]. A memória de um filtro é o estado inicial do sistema. No CELP original, por exemplo, ao final de um quadro, o filtro PCA tem armazenadas em sua memória as OP últimas amostras de voz sintetizada, onde OP é a ordem do filtro PCA (equação (2.1)). Na fase de determinação do PLA e da melhor "codeword" (determinação da excitação), o estado inicial do PCA deve ser o mesmo para cada filtração executada. Podemos entretanto evitar a necessidade de recarregar as condições iniciais do PCA para cada filtração. A Fig. 2.6 exemplifica a separação da resposta de um sistema com memória não-nula através da soma da resposta à entrada-zero e da resposta ao estado-zero [38].

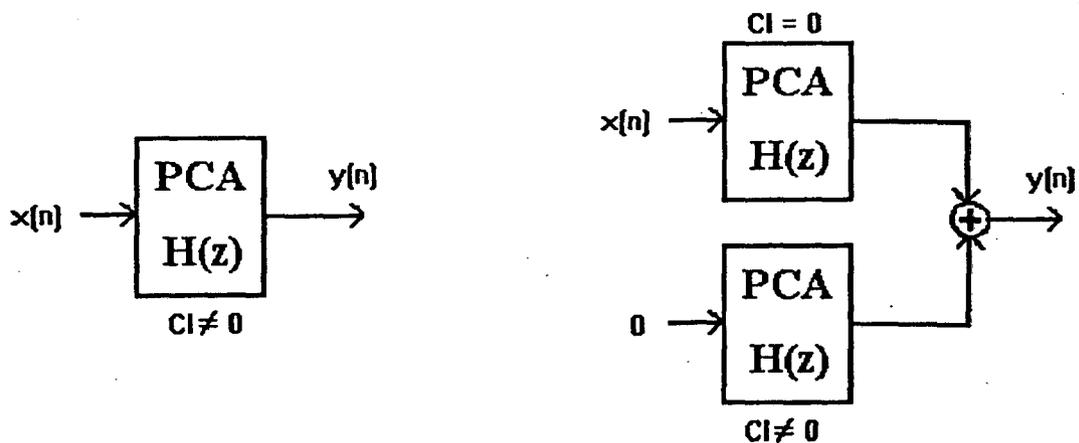


Fig. 2.6(a) - Condições iniciais não-nulas

Fig. 2.6(b) - Representação equivalente

O objetivo do filtro ponderador é concentrar o erro nas regiões de maior energia da envoltória espectral (região dos formantes), onde o sistema auditivo não o distingue devido ao mascaramento auditivo [12], [21], [176]. A forma mais utilizada para a função de transferência do filtro ponderador do erro possui numerador igual ao denominador da função de transferência  $H(z)$  do filtro PCA [176]. Retirando a influência da memória dos filtros PCA e ponderador, podemos tratá-los como sistemas lineares rearranjando o diagrama de blocos da Fig. 2.3 de forma que o filtro  $W(z)$  seja antecipado para os dois ramos anteriores. A composição do PCA com  $W(z)$  resulta na função de transferência  $H_w(z)$ :

$$H_w(z) = H(z)W(z) = \frac{1}{\sum_{i=1}^p a_i \gamma^i z^{-i}} \quad (2.4)$$

Uma das estruturas mais empregadas na implementação de codificadores CELP [129], [135], [138], [143], [144], [158], [206] é apresentada na Fig. 2.7. Nesta estrutura o filtro ponderador  $W(z)$  é transportado para os dois ramos anteriores do diagrama. Este filtro é combinado com o PCA resultando no filtro  $PCA_w$ . A voz a ser codificada  $s(n)$  é ponderada por  $W(z)$  a cada quadro (aproximadamente 20 ms). O efeito da memória do filtro  $PCA_w$  é computado, subtraindo-se sua resposta à entrada-zero do sinal de voz ponderado. O sinal alvo resultante é  $s'(n)$ . A determinação da excitação será feita então de forma a produzir o sinal  $\hat{s}(n)$  que tenha o menor erro quadrático em relação a  $s'(n)$ .

Observamos da Fig. 2.7 que o filtro  $PCA_w$ , resultante do cascadeamento do filtro PCA e  $W(z)$ , passa a ser utilizado para sintetizar a voz durante todo o processo de codificação. Na decodificação, o filtro  $W(z)$  não existe e a voz é sintetizada filtrando-se a excitação através do próprio filtro PCA.

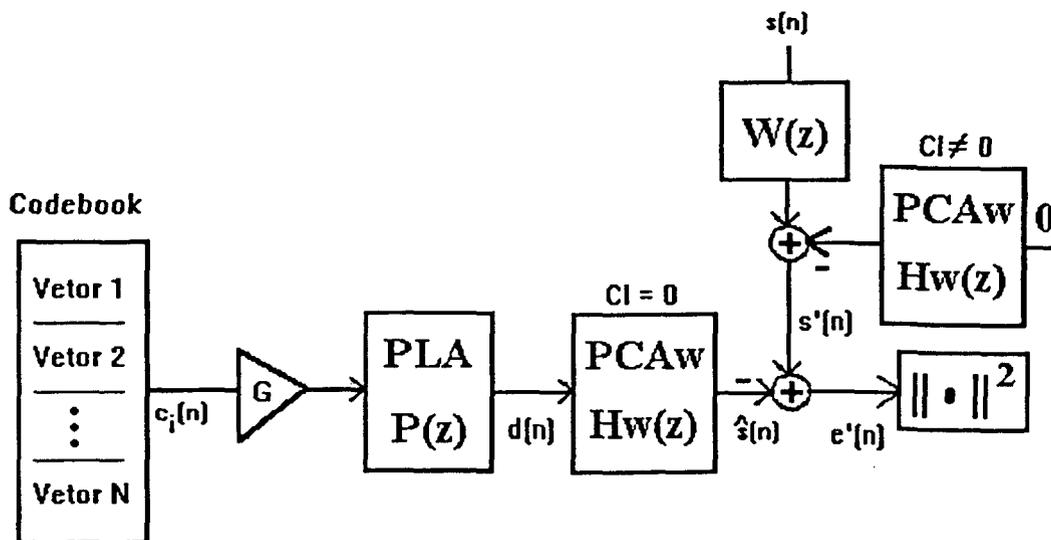


Fig. 2.7 Diagrama em blocos do CELP com a estrutura modificada

A determinação da excitação  $d(n)$  no CELP é um processo de duas etapas. Inicialmente, determina-se o filtro PLA. Depois determina-se a codeword e o ganho. Para se obter os parâmetros do PLA, considera-se  $G=0$  (Fig. 2.7). Os valores passados da excitação  $d(n)$  encontram-se armazenados na memória do PLA. O atraso  $M$  do PLA (equação (2.3)) assume tipicamente valores na faixa de 20 a 147 (Tabela 2.1). A cada um destes valores corresponde uma seqüência candidata obtida a partir das amostras armazenadas na memória do PLA. Estas amostras são filtradas pelo  $PCA_w$  com memória nula e comparadas com  $s'(n)$ . Este processo é conhecido como cálculo do PLA em laço-fechado, pois é baseado na análise-por-síntese. Outra alternativa para determinar o PLA é conhecida como cálculo em laço aberto. Nesta técnica, determinamos a melhor seqüência candidata, ou seja o valor de  $M$ , comparando-a diretamente com o resíduo da predição do PCA (sinal  $e(n)$  na Fig. 2.1). Esta comparação é feita através da correlação normalizada [176]. O cálculo do PLA em laço-aberto é mais simples computacionalmente pois evita a filtragem das seqüências candidatas para cada valor de  $M$ . A desvantagem deste método é que o mesmo conduz a uma qualidade de voz inferior à obtida pelo cálculo em laço-fechado [176]. A determinação do filtro PLA, tanto em laço aberto quanto em laço fechado, é bastante conhecida [176] e não a representaremos em detalhes neste trabalho.

A determinação do PLA com um único coeficiente em laço-fechado é análoga à determinação da melhor codeword e seu ganho. O valor de  $M$  pode ser visto como um índice de codeword e o valor do coeficiente  $B$  (Eq. 2.3) como um ganho. Neste caso, as duas etapas de determinação da excitação tornam-se similares. Desta forma, algumas técnicas de redução de complexidade podem ser aplicadas nas duas etapas de determinação da excitação. Isto será abordado oportunamente neste trabalho.

Em 1986, a determinação do PLA em laço aberto ainda era bastante usada. Frente à complexidade computacional da determinação total da excitação, o cálculo do PLA em laço aberto representava uma pequena parcela. Por este motivo, os trabalhos [46] e [48] preocupam-se apenas com o custo computacional da determinação da melhor "codeword"  $c_i(n)$  e de seu ganho  $G$ .

Determinado o PLA, devemos extrair sua contribuição do sinal alvo de voz  $s'(n)$  [102]. Esta contribuição equivale à resposta do filtro PLA à entrada-zero, pois este foi determinado considerando  $G=0$  (Fig. 2.7). O resultado desta subtração é o novo sinal alvo de voz  $s''(n)$ .

A segunda etapa da determinação da excitação consiste na filtragem de cada codeword candidata  $c_i(n)$  através dos filtros PLA e  $PCA_w$  com memórias nulas. Pode-se observar que, dependendo do valor de  $M$ , o filtro PLA pode vir a não influenciar esta filtragem. A filtragem através da cascata composta pelos filtros PLA e  $PCA_w$  pode ser realizada através de equações de diferenças ou através de convolução, calculada a partir do produto entre matrizes. A modelagem matricial [46], [48] parte do cálculo das primeiras  $E$  amostras da resposta ao impulso conjunta  $h(n)$  dos filtros PLA e  $PCA_w$ , onde  $E$  é a duração do subquadro da excitação (Tabela 2.1). A convolução de  $h(n)$  com a codeword escalada  $Gc_i(n)$  é expressa pelo produto matricial  $HC$ , onde  $H$  é a matriz composta a partir do truncamento de  $h(n)$  em  $E$  amostras e  $C$  corresponde ao vetor coluna  $Gc_i(n)$ . Por exemplo, para um subquadro com  $E=4$  amostras, a matriz  $H$  seria constituída por:

$$H := \begin{bmatrix} h_0 & 0 & 0 & 0 \\ h_1 & h_0 & 0 & 0 \\ h_2 & h_1 & h_0 & 0 \\ h_3 & h_2 & h_1 & h_0 \end{bmatrix}$$

Devemos ressaltar que o uso de convolução, por si só, não apresenta vantagens em relação ao uso das equações de diferenças. Em geral, a formulação matricial requer  $E(E+1)/2$  MACs para calcular  $HC$ . Este cálculo pode ser feito com  $E(P+T)$  MACs através de equações de diferenças, onde  $T$  é o número de coeficientes ("taps") do PLA. Para valores usuais ( $E=40$ ,  $P=10$ ,  $T=1$ ),  $(E+1)/2$  é maior do que  $(P+T)$ . Assim, o uso de convoluções implicaria em maior número de operações. O motivo da ampla utilização das convoluções é o fato das "codewords" geralmente serem vetores esparsos. Utilizando vetores esparsos, pode-se reduzir o número de multiplicações das equações de diferença. Todavia, neste caso o número de comparações aumenta consideravelmente, superando o número de MACs economizadas. Por outro lado, os produtos matriciais (convolução) obtém grande benefício da esparsidade das "codewords".

A idéia apresentada por Gersho e Davidson em [46] baseia-se na exploração das características dos "codebooks" esparsos. Podemos obter "codebooks" esparsos realizando a ceifagem central de um "codebook" gaussiano [147]. Para o CELP, o uso de "codebooks" esparsos possibilita redução na complexidade computacional, pode melhorar a qualidade subjetiva da voz sintetizada e reduz a

memória necessária para o armazenamento do "codebook" por um fator que pode chegar até a sete vezes [46].

Utilizando-se "codewords" com apenas  $N_p$  amostras não-nulas, resulta em um número médio de  $N_p(E+1)/2$  operações de MAC para o cálculo da convolução HC. A Fig. 2.8 [46] mostra que podemos utilizar apenas 4 amostras não-nulas por "codeword" (considerando cada "codeword" com 40 amostras) sem introduzir degradação significativa na RSRSEG média. Desta forma, o ganho computacional deste método é substancial em relação à utilização das equações de diferenças nas filtragens.

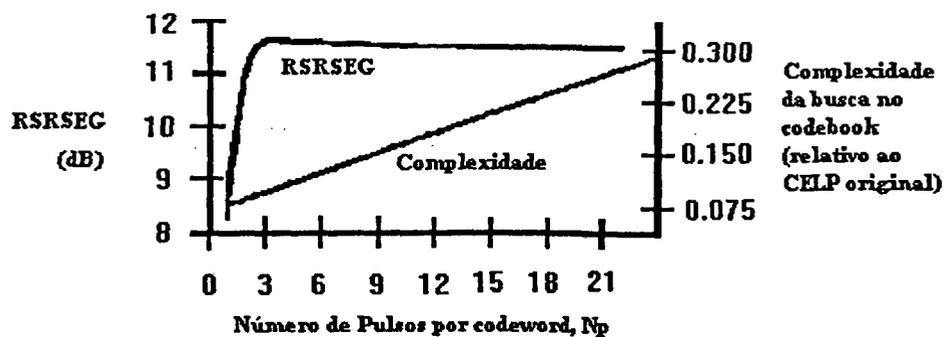


Fig. 2.8- Desempenho obtido com "codebooks" esparsos [46].

Devemos ressaltar que a técnica apresentada em [46] não permite a utilização de instruções de MAC, disponíveis em um único ciclo como, por exemplo, as existentes em DSPs da família TMS320. As operações de multiplicação e de adição devem ser efetuadas individualmente a partir de suas instruções respectivas, o que é uma desvantagem.

Descrevemos então, a formulação matricial para a segunda etapa do cálculo da excitação no CELP. A figura de mérito a ser minimizada é a energia do erro entre o sinal alvo de voz da segunda etapa  $s''(n)$  e o sinal sintetizado nesta etapa  $\hat{s}''(n)$ . O sinal  $s''(n)$  é obtido subtraindo-se a contribuição do PLA do sinal  $s'(n)$  (Fig. 2.7). O sinal  $\hat{s}''(n)$  é obtido realizando-se a convolução da "codeword" escalada  $Gc_i(n)$ , com a resposta ao impulso  $h(n)$  da cascata dos filtros PLA e PCA. A energia do erro  $e_i(n)$  é dada por:

$$\|e_i(n)\|^2 = \|s''(n) - \hat{s}''(n)\|^2 \quad (2.5)$$

$$\|e_i(n)\|^2 = \|s''(n) - HGc_i(n)\|^2 \quad (2.6)$$

Para encontrar o ganho  $G$  que minimiza a energia do erro, impomos:

$$\frac{\partial \|e_i(n)\|^2}{\partial G} = 0 \quad (2.7)$$

o que leva a:

$$G_{\text{ótimo}} = \frac{s''(n)^T Hc_i(n)}{\|Hc_i(n)\|^2} \quad (2.8)$$

A substituição do valor de  $G_{\text{ótimo}}$  em (2.6), conduz a

$$\|e_i(n)\|^2 = \|s''(n)\|^2 - \frac{(s''(n)^T Hc_i(n))^2}{\|Hc_i(n)\|^2} \quad (2.9)$$

O primeiro termo do lado direito da equação (2.9) representa a energia do sinal  $s''(n)$  e não depende da "codeword"  $c_i(n)$ . Com isto, a minimização de (2.9) equivale à maximização de seu segundo termo. Este termo equivale à correlação cruzada entre  $s''(n)$  e  $Hc_i(n)$ , elevada ao quadrado e normalizada pela energia da "codeword" filtrada  $Hc_i(n)$ .

$$\|e_i(n)\|^2 \Big|_{\text{mínimo}} \Leftrightarrow \frac{(s''(n)^T Hc_i(n))^2}{\|Hc_i(n)\|^2} \Big|_{\text{máximo}} \quad (2.10)$$

Uma consideração deve ser feita em relação à equação (2.9). Esta equação foi obtida a partir da substituição da expressão de  $G_{\text{ótimo}}$  em (2.6). Para fins de codificação, o ganho  $G$  deve ser quantizado com um número finito de bits. Sendo assim, seria mais preciso determinarmos (2.6) utilizando o valor quantizado de  $G_{\text{ótimo}}$ . Isto corresponde à inserção da quantização do ganho  $G$  no processo de busca da melhor "codeword" [102]. Esta alternativa possui maior importância à medida que  $G$  é quantizado com poucos bits (menos do que 5 bits). Quando a quantização de  $G$  é feita de forma razoavelmente precisa, a complexidade adicional da inserção da quantização de  $G$  no laço de busca da "codeword" pode ser dispensada.

O esforço computacional relativo ao cálculo da convolução  $Hc_i(n)$  em (2.9) pode ser substancialmente reduzido usando-se a técnica de decomposição do valor singular ("Singular-Value Decomposition") [48]. Esta técnica consiste na decomposição da matriz  $H$  em duas matrizes unitárias reais  $U$  e  $V$  e uma matriz diagonal  $D$ :

$$H = UDV^T \quad (2.11)$$

Esta decomposição permite manipularmos a equação (2.9) de forma a obter uma redução na complexidade de um fator  $P+T$  [48]. O problema deste método é o custo adicional do cálculo da decomposição.

Outro método proposto em [48] utiliza a busca da "codeword" no domínio da frequência. Neste método, despreza-se a influência do PLA na busca da melhor "codeword". Calcula-se a resposta ao impulso  $h_w(n)$  do filtro  $PCA_w$ . Considera-se que a saída do filtro  $PCA_w$ , excitado por uma

"codeword" de comprimento equivalente a 5 ms, desvanece de tal maneira que pode ser considerada nula após 10 ms. Assim, uma DFT ("Discrete Fourier Transform") de 80 pontos (considerando-se  $f_s=8$  kHz) é suficiente para representar a resposta deste filtro  $S_i(k)$  no domínio da frequência. Neste método, o "codebook" é formado pelas DFTs das "codewords". A convolução entre  $h_w(n)$  e  $Gc_i(n)$  pode ser representada no domínio da frequência, como:

$$S_i(k) = F(k)GC_i(k) \quad (2.12)$$

onde:

$$F(k) = \text{DFT}\{h_w(n)\} \quad (2.13)$$

$$C_i(k) = \text{DFT}\{c_i(n)\} \quad (2.14)$$

O erro a ser minimizado é dado por:

$$E_i = \sum_{i=1}^{40} |S_i(k) - F(k)GC_i(k)|^2 \quad (2.15)$$

A terceira proposta apresentada em [48] é um esquema eficiente para calcular o termo referente à energia de  $Hc_i(n)$ . O método baseia-se no fato de que a soma dos quadrados da convolução de duas seqüências é igual à correlação cruzada entre as autocorrelações destas duas seqüências. Sendo assim, as autocorrelações de cada "codeword" são armazenadas e calcula-se o termo referente à energia na equação (2.10) através de:

$$\|Hc_i(n)\|^2 = R_{c_i}(0)R_{h_w}(0) + 2 \sum_{m=1}^E R_{c_i}(m)R_{h_w}(m) \quad (2.16)$$

onde  $R_{c_i}(m)$  e  $R_{h_w}(m)$  são as m-ésimas amostras das funções de autocorrelação de  $c_i(n)$  e  $h_w(n)$  respectivamente.

As autocorrelações  $R_{c_i}(m)$  podem ser pré-calculadas e armazenadas, fazendo com que o esforço computacional deste método seja reduzido. O único cálculo adicional seria o cálculo da autocorrelação  $R_{h_w}(m)$ . A desvantagem deste método é o aumento na memória necessária para armazenamento. Além do "codebook" com as "codewords"  $c_i(n)$ , devemos armazenar também suas correlações  $R_{c_i}(m)$ .

Uma característica importante de [48] é que nenhum dos métodos apresentados faz restrições em relação à estrutura do "codebook". Quando Atal e Schoroeder apresentaram o CELP [33],

fizeram a ressalva de que o "codebook" gaussiano utilizado não era a opção ótima. Isto permite supor que o uso de um "codebook" melhor projetado poderia conduzir a resultados superiores [147].

Surgiram outros trabalhos abordando a redução de complexidade do CELP em 1986 (ex: [41], [43], [45]). Entretanto, [46] e [48] formaram a base para diversos outros desenvolvimentos referentes à redução da complexidade computacional do CELP.

O CELP continuou a amadurecer em 1987. Os trabalhos [58] e [59], em relação à alocação dos bits no CELP (distribuição dos bits entre os parâmetros) e à qualidade da voz sintetizada, estabeleceram melhor as relações de compromisso entre taxa e qualidade.

Em [59], Atal e Kroon propõem também uma alternativa para a redução da complexidade computacional. Esta alternativa baseia-se no seccionamento das "codewords" e no uso de mais de um ganho para o "codebook" gaussiano. Na estrutura de múltiplos ganhos de [59], cada "codeword" é dividida em  $L$  partes. Cada uma destas partes corresponde a  $E/L$  amostras. Cada uma destas parcelas é escalada por um ganho  $g_L$  e a "codeword" resultante é obtida a partir da soma de todas as parcelas. Este processo é ilustrado simbolicamente na Fig. 2.9, para  $L=2$  e  $E=10$  amostras.

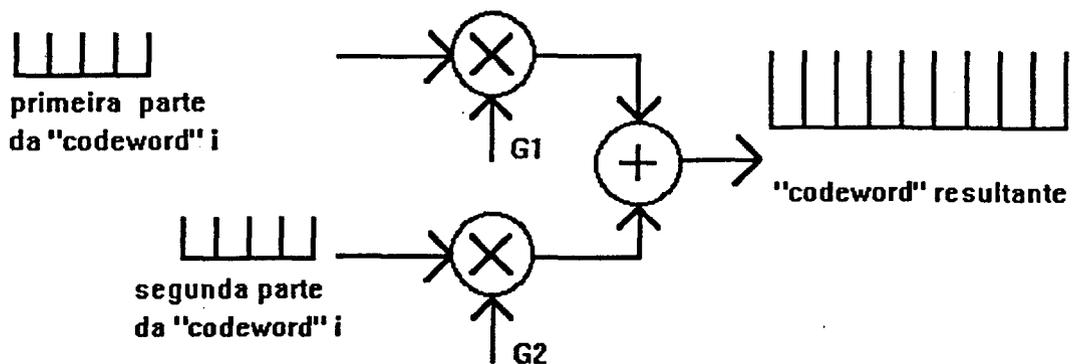


Fig. 2.9 - Estrutura múltiplo ganho de [59]

Em [176] é apresentada uma extensa avaliação de uma outra estrutura de múltiplos ganhos, chamada CMSEC ("Constrained Multiple-Stage Excitation Coding"). Na estrutura CMSEC, a "codeword" resultante é formada pela soma de  $L$  seqüências, cada uma ponderada por um ganho  $g_L$ . Para obter estas seqüências são utilizados  $L$  "codebooks" diferentes. Na composição da "codeword" resultante, utiliza-se sempre "codewords" de mesmo índice para todos os "codebooks" [176], [188]. Na Fig. 2.10 mostramos o processo de formação da "codeword" resultante para  $L=2$  e  $E=10$  amostras.

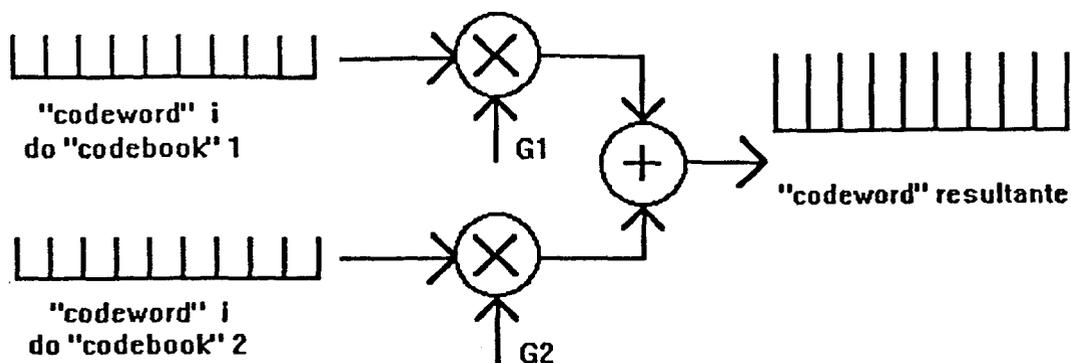


Fig. 2.10- Estrutura CMSEC

De forma geral, o uso de uma combinação linear entre vetores para formar a "codeword" resultante, conduzirá a um sistema de equações lineares em função dos ganhos. Assim, para ambas as estruturas, os valores ótimos dos ganhos são obtidos simultaneamente através da resolução de um sistema de  $L$  equações lineares [59], [176].

O uso de múltiplos ganhos permite reduzir significativamente a complexidade da codificação CELP. A determinação da "codeword" e de seu ganho no CELP corresponde a uma quantização ganho-forma [31] (esta interpretação para a codificação CELP será aprofundada oportunamente). A utilização de um único "codebook" corresponde tipicamente à atribuição de 10 bits para a forma (1024 "codewords") e 5 bits para o ganho. As estruturas múltiplos ganhos possibilitam maior flexibilidade na alocação de bits entre a forma ("codebook") e os ganhos. Uma vez que a determinação da melhor "codeword" está inserida em um processo de elevada complexidade computacional (análise-por-síntese), esta complexidade é diminuída à medida em que se reduz o número de bits associado à forma, repassando-os para os ganhos. Desta forma pode-se alcançar configurações que apresentem um compromisso razoável entre complexidade computacional, taxa de bits e qualidade da voz sintetizada [176], [188].

As evoluções obtidas pela equipe de Gersho são apresentadas em [64]. Este importante trabalho alia o uso eficiente de "codebooks" esparsos [46] ao método da autocorrelação [48]. Desta forma, a complexidade de busca da "codeword" foi reduzida para 0.55 MFLOPS. Isto corresponde a uma redução na complexidade de um fator de 800 vezes em relação ao CELP original [33]. O codificador proposto opera à taxa de 4,8 kbit/s e necessita de 16 Kbytes de memória de dados e de programa. Os autores mostram em [64] sua expectativa de implementar o CELP proposto em tempo real, utilizando um DSP de ponto flutuante como o AT&T DSP32 de 16 MHz. Estes dados referem-se ao uso de um "codebook" com apenas 256 "codewords".

Em relação ao projeto do "codebook", ainda em [64] foi desenvolvida uma expressão para o cálculo de centróide que leva em conta a estrutura da codificação CELP. O cálculo de centróides é uma das etapas dos algoritmos de projeto de "codebooks" como, por exemplo, o LBG. Assim, pode-se projetar um "codebook" para o CELP usando-se a expressão de centróide definida em [64].

Todavia, em [64] não são apresentados resultados da utilização desta técnica, ou seja, não são feitas comparações entre "codebooks" projetados a partir da técnica apresentada e outros "codebooks" como, por exemplo, o gaussiano.

Outro trabalho importante desta época é o de Lin [62]. Lin apresentou alguns "codebooks" com estruturas especiais que chamou de "codebooks" pseudo-estocásticos. Um destes, o "codebook" com deslocamento circular, alcançou grande aceitação (Fig. 2.11). Neste "codebook" as "codewords" são superpostas, compartilhando amostras entre si [147]. O "codebook" com deslocamento circular é um vetor, podendo ser chamado de "codevector" [149]. Para este tipo de "codebook", Lin apresentou relações recursivas que tornam o cálculo da convolução  $Hc_i(n)$  mais eficiente. Esta técnica é conhecida como "correção dos pontos extremos" [102]. Devido à sua importância na atualidade, a técnica de "correção dos pontos extremos" é descrita a seguir.

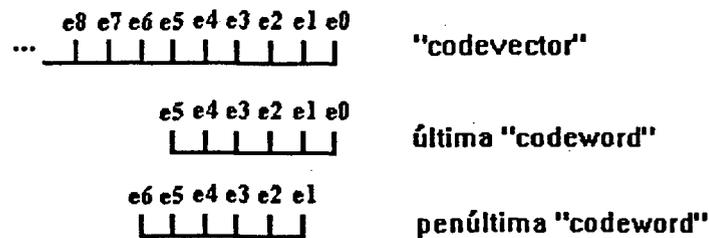


Fig. 2.11 - Exemplo de "codewords" superpostas com deslocamento de uma amostra

Ao calcularmos a convolução  $y_i(n) = Hc_i(n)$  utilizando um "codebook" com deslocamento circular, podemos usar o resultado da convolução da "codeword" anterior  $c_{i-1}(n)$ . Retiramos a influência da amostra que saiu da convolução, acrescentamos o efeito da nova amostra e ajustamos as novas posições das amostras de  $y_i(n)$ . As convoluções são calculadas iniciando-se pelas últimas amostras do "codebook". A primeira convolução a ser calculada é referente à última "codeword" do "codevector". Para o exemplo da Fig. 2.11, o resultado seria:

$$\begin{bmatrix} h0 & 0 & 0 & 0 & 0 & 0 \\ h1 & h0 & 0 & 0 & 0 & 0 \\ h2 & h1 & h0 & 0 & 0 & 0 \\ h3 & h2 & h1 & h0 & 0 & 0 \\ h4 & h3 & h2 & h1 & h0 & 0 \\ h5 & h4 & h3 & h2 & h1 & h0 \end{bmatrix} \begin{bmatrix} e5 \\ e4 \\ e3 \\ e2 \\ e1 \\ e0 \end{bmatrix} = \begin{bmatrix} h0 \cdot e5 \\ h1 \cdot e5 + h0 \cdot e4 \\ h2 \cdot e5 + h1 \cdot e4 + h0 \cdot e3 \\ h3 \cdot e5 + h2 \cdot e4 + h1 \cdot e3 + h0 \cdot e2 \\ h4 \cdot e5 + h3 \cdot e4 + h2 \cdot e3 + h1 \cdot e2 + h0 \cdot e1 \\ h5 \cdot e5 + h4 \cdot e4 + h3 \cdot e3 + h2 \cdot e2 + h1 \cdot e1 + h0 \cdot e0 \end{bmatrix}$$

O resultado da convolução  $y_i(n)$  pode beneficiar-se do resultado da convolução anterior  $y_{i-1}(n)$ , apresentado acima. Cada amostra de  $y_i(n)$  pode ser calculada como a soma de  $y_{i-1}(n-1)$  com o produto referente à nova amostra da convolução  $y_i(n)$ . Por exemplo,  $y_i(3)$  é calculada como

a soma de  $y_{i-1}(2)$  com o produto "h2 e6". Desta forma, a influência da amostra "e6" que está entrando na convolução é contabilizada. A retirada do efeito da amostra "e0", que está saindo da convolução, é realizada naturalmente pelo deslocamento de  $y_i(n)$ . Assim, para a penúltima "codeword" teríamos:

$$\begin{bmatrix} h0 & 0 & 0 & 0 & 0 & 0 \\ h1 & h0 & 0 & 0 & 0 & 0 \\ h2 & h1 & h0 & 0 & 0 & 0 \\ h3 & h2 & h1 & h0 & 0 & 0 \\ h4 & h3 & h2 & h1 & h0 & 0 \\ h5 & h4 & h3 & h2 & h1 & h0 \end{bmatrix} \cdot \begin{bmatrix} e6 \\ e5 \\ e4 \\ e3 \\ e2 \\ e1 \end{bmatrix} = \begin{bmatrix} h0 \cdot e6 \\ h1 \cdot e6 + h0 \cdot e5 \\ h2 \cdot e6 + h1 \cdot e5 + h0 \cdot e4 \\ h3 \cdot e6 + h2 \cdot e5 + h1 \cdot e4 + h0 \cdot e3 \\ h4 \cdot e6 + h3 \cdot e5 + h2 \cdot e4 + h1 \cdot e3 + h0 \cdot e2 \\ h5 \cdot e6 + h4 \cdot e5 + h3 \cdot e4 + h2 \cdot e3 + h1 \cdot e2 + h0 \cdot e1 \end{bmatrix}$$

Outra linha de pesquisa relevante apresentada em 1987 foi desenvolvida por Adoul et al [56], [60]. Nestes trabalhos, alcança-se uma redução efetiva na complexidade restringindo-se as "codewords" a terem energia unitária e amplitudes -1 e 1 apenas. As N "codewords" de dimensão K podem ser interpretadas geometricamente como vetores no espaço K-dimensional. Fazendo-se com que estas "codewords" tenham energia unitária, as mesmas podem ser vistas como um conjunto de N pontos de uma hiper-esfera K-dimensional. O projeto do "codebook" deve preocupar-se com a melhor distribuição possível destes N pontos na hiper-esfera. A alternativa proposta é a utilização de códigos binários, derivados da teoria dos códigos corretores de erro [126]. Estes códigos conduzem a um conjunto de pontos na hiper-esfera que buscam guardar a maior distância possível uns dos outros. A vantagem computacional desta técnica é que os códigos binários permitem o uso de algoritmos rápidos para o cálculo dos produtos internos, necessários na etapa de busca da melhor "codeword". O codificador apresentado por Adoul et al pode ser implementado em um simples DSP TMS32010.

Em 1988, Kroon e Atal aprofundaram seus estudos em relação à distribuição de bits no CELP [71]. Em termos de quantização do PCA, as LSFs ("Line Spectral Frequencies") ganharam maior importância com o trabalho de Soong e Juang [69]. Neste artigo, o intuito dos autores era alcançar uma quantização transparente, ou seja, não introduzir degradação subjetivamente audível por ocasião da quantização. O leitor interessado pode encontrar em [118, pp. 143] mais detalhes sobre o processo de detecção de limiares de percepção. A medida de distorção utilizada em [69], foi a distorção espectral (SD) média dada abaixo:

$$\overline{SD} = \sqrt{\frac{1}{Q} \sum_{n=1}^Q SD_n^2} \tag{2.17}$$

onde:

$$SD_n^2 = \frac{1}{\pi} \int_0^{\pi} \left( 10 \log \frac{S_n(w)}{\hat{S}_n(w)} \right)^2 dw \quad (2.18)$$

Em (2.17) e (2.18)  $S_n(w)$  é o espectro de potência do filtro LPC original,  $\hat{S}_n(w)$  é o espectro de potência do filtro LPC quantizado,  $Q$  é o número de quadros e  $\overline{SD}$  é a distorção espectral média em dB.

Atualmente, a distorção espectral é a medida objetiva mais utilizada na avaliação da quantização do PCA. Em [69] define-se a transparência como uma distorção espectral média menor do que 1 dB. Foi observado em [92] que, por mais que a distorção média seja baixa, a ocorrência esporádica de quadros com distorção elevada (denominados "outliers") provoca degradação audível. O conceito de transparência foi aperfeiçoado em [146], passando a ser amplamente utilizado desde então. Adiantaremos aqui este conceito, que se baseia no atendimento de três condições:

- i) a distorção espectral média deve ser aproximadamente 1 dB.
- ii) não devem haver quadros com distorção espectral maior do que 4 dB.
- iii) o número de quadros apresentando distorção espectral entre 2 e 4 dB deve ser menor do que 2 %.

A maioria dos codificadores CELP implementados até então operavam à taxa de 4,8 kbits/s. A quantização do PCA nestes codificadores era realizada de maneira relativamente grosseira. A exigência de uma melhor qualidade de voz no CELP tornou necessária uma quantização mais precisa do PCA. Neste contexto, o conceito de transparência descrito acima passou a ser muito útil na avaliação da quantização do PCA.

Uma característica importante da voz é sua natureza dinâmica. Um exemplo disso é a diferença significativa nas características dos sons vozeados e não-vozeados. Por causa desta característica podemos atingir um melhor desempenho realizando uma alocação dinâmica dos bits no CELP [78]. Outra abordagem possível para esta questão seria reduzir a taxa média de transmissão, permitindo que o codificador trabalhe com taxas variáveis [77]. Esta segunda linha de pesquisa vem ganhando maior espaço atualmente devido à concepção das redes ATM ("Asynchronous Transfer Mode") [104]. O sistema ATM é parte integrante das redes de telecomunicações modernas e absorve naturalmente codificadores operando com taxas variáveis.

Apesar dos avanços já obtidos, as implementações do CELP em tempo real geralmente exigiam a limitação do número de "codewords" a um valor menor do que 1024. O CELP operando a 4,8 kbits/s apresentava uma qualidade de voz bem superior ao do vocoder LPC. Entretanto, para melhorar ainda mais esta qualidade, necessitava-se de "codebooks" de dimensões maiores. A utilização de estruturas em múltiplos estágios é uma conhecida técnica de redução de complexidade da quantização vetorial [36]. Em [72], Gersho e Davidson incorporam ao CELP a utilização de

estruturas em múltiplos estágios. Posteriormente, esta filosofia foi aplicada com grande sucesso pela Motorola no codificador VSELP [113]. O VSELP é uma derivação do CELP que se tornou famosa por ser escolhido pela TIA ("Telecommunications Industries Association") como padrão para telefonia móvel celular nos EUA [172], [208].

Ainda em termos de redução de complexidade, nesta época ocorreu um desenvolvimento muito importante no que se refere ao cálculo do PLA [74], [70]. Podemos observar que um "codebook" com deslocamento circular ("codevector") de uma única amostra é similar às seqüências que são obtidas da memória do filtro PLA. A Fig. 2.12 ilustra simbolicamente esta semelhança.

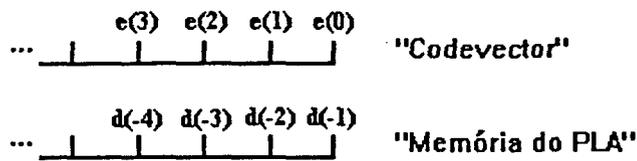


Fig. 2.12- Semelhança entre um "codevector" e a memória do PLA.

Relembramos então, uma analogia já citada. Considerando-se que o filtro PLA tenha apenas um coeficiente B (equação (2.3)), o valor do atraso M pode ser visto como o índice de um "codebook", enquanto o coeficiente B do filtro PLA é análogo a um ganho. Assim, o filtro PLA desempenha tarefa similar a um "codebook" adaptativo, pois a memória deste filtro é atualizada a cada subquadro. Pode-se obter uma redução da complexidade computacional caso se realize a determinação dos parâmetros do PLA como um "codebook" adaptativo de fato. Um exemplo seria a aplicação do método da correção dos pontos extremos [102], descrito anteriormente. Na Fig. 2.13 ilustramos a atuação de um "codebook" adaptativo.

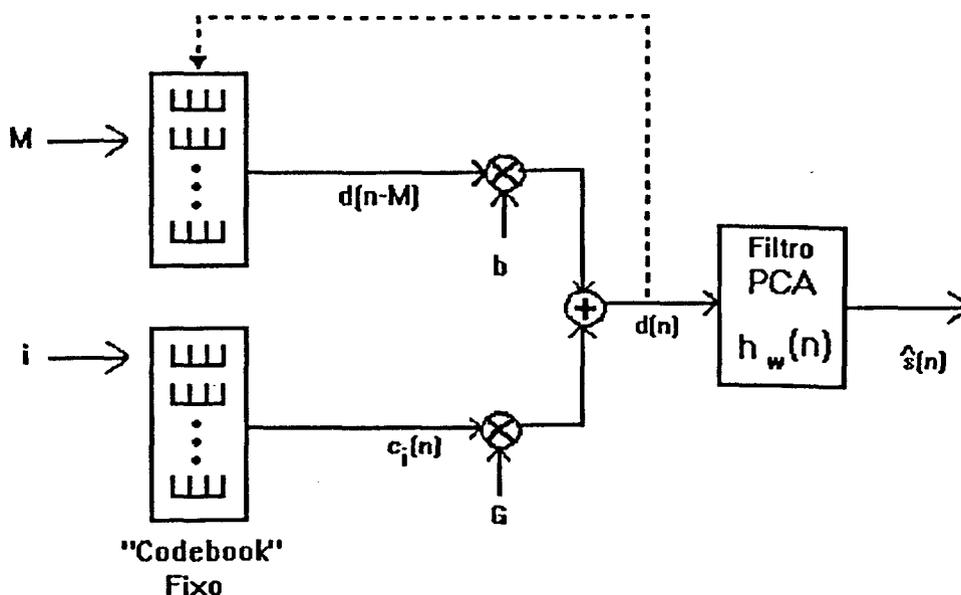


Fig. 2.13- Codificação CELP utilizando um "codebook" adaptativo.

O impedimento de realizarmos uma analogia irrestrita do PLA de um único coeficiente com um "codebook" adaptativo é que as "codewords" correspondentes a  $M < E$  (duração do subquadro de excitação) não teriam todas as suas amostras conhecidas. Estas amostras seriam dependentes da saída do filtro PLA no próprio subquadro em questão.

Para contornar este problema, pensou-se inicialmente em utilizar o valor de  $E$  como limite mínimo para a faixa de variação de  $M$ . Desta forma, podemos determinar os parâmetros do PLA, utilizando o mesmo algoritmo de correção dos pontos extremos [102] desenvolvido para "codebooks" com deslocamento circular [62].

Discutamos as implicações da restrição  $M \geq E$ . A duração do subquadro é tipicamente em torno de 5 ms, o que corresponde a 40 amostras para  $f_s=8$  kHz. Para sons vozeados, o valor de  $M$  é correlacionado com o período de pitch. O fato de restringirmos o valor de  $M$  a ser maior ou igual a 40, portanto, prejudica as vozes que tenham frequência de pitch alta. Isto acontece principalmente com as vozes femininas e infantis, que corresponderiam a valores pequenos de  $M$ .

A degradação do desempenho do "codebook" adaptativo com  $M \geq E$  é exemplificada na Fig. 2.14. A memória do filtro PLA é composta pelos valores da excitação sintetizada nos subquadros passados. Na transição de um som não-vozeado para um som vozeado, a maioria das "codewords" do "codebook" adaptativo, extraídas da memória do filtro PLA, não possuem pulsos relativos ao pitch. Na Fig. 2.14, apenas um pulso está presente na memória do filtro PLA. Este filtro implementado da forma original, através da filtragem recursiva, seria capaz de gerar os dois pulsos de pitch necessários para compor de maneira mais adequada a excitação do subquadro a ser codificado.

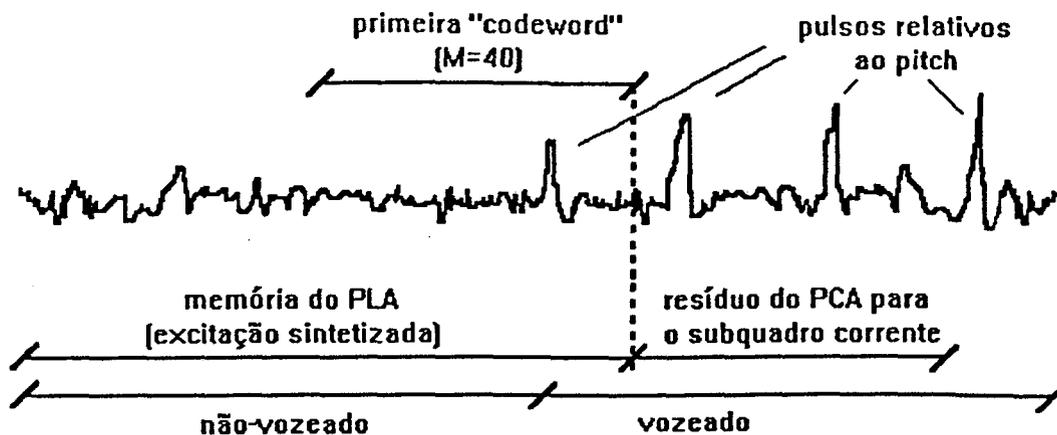


Fig. 2.14-Exemplo do sinal de excitação (resíduo do filtro PCA) na transição não-vozeado p/ vozeado

Em [70] é proposto um artifício para contornar a restrição  $M \geq E$ . A idéia consiste em definir o "codebook" adaptativo da seguinte forma:

$$d(n) = d(n-M) \quad , \quad 0 \leq n < M \quad (2.19a)$$

$$d(n) = d(n-2M) \quad , \quad M \leq n < E \quad (2.19b)$$

onde  $d(n)$  é a excitação sintetizada (Fig. 2.7).

Uma solução mais eficiente para este problema foi apresentada em [74]. A proposta consiste na criação de "codewords" virtuais para  $M < E$ . Por exemplo, para  $M=p$ , onde  $p < E$ , apenas as  $p$  primeiras amostras estão definidas *a priori*. Repete-se estas amostras com periodicidade  $p$  até formar-se as  $E$  amostras necessárias. A partir deste trabalho [74], o filtro PLA passou a ser tratado como um "codebook" adaptativo. Isto permitiu reduzir os cálculos tanto na determinação dos parâmetros do filtro PLA como na determinação da "codeword". Isso porque este filtro implementado como um "codebook" adaptativo não mais interfere na segunda etapa de determinação da excitação.

Daqui em diante, estaremos utilizando o termo "codebook" fixo para diferenciá-lo do "codebook" adaptativo, correspondente ao PLA.

Na concepção moderna do CELP (Fig. 2.15), a quantização da excitação  $e(n)$  (Fig. 2.1) passa a ser vista como um processo de quantização vetorial em dois estágios [29], onde cada estágio utiliza a técnica ganho-forma [31], [206]. Devemos levar em conta que estas técnicas no CELP estão embutidas em um processo de análise-por-síntese. O trabalho de Kleijn et al [102] é muito útil para consolidar esta idéia.

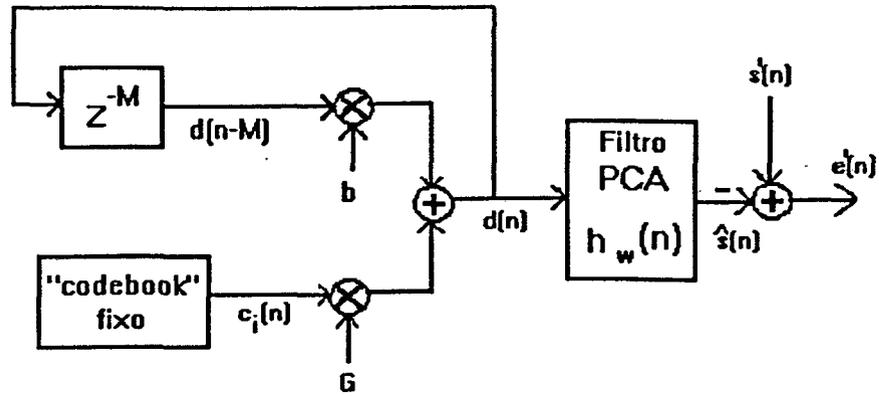


Fig. 2.15- Concepção moderna do CELP

Em 1989 continuaram a ser apresentados trabalhos concernentes à alocação dinâmica dos bits no CELP [87], [88].

Uma discussão importante acerca da qualidade da voz sintetizada pelo CELP é encontrada em [85]. Neste trabalho, é abordada uma deficiência do CELP que já acarretou um considerável esforço por parte dos pesquisadores [44]. O CELP introduz uma distorção na estrutura harmônica dos sons vozeados. Comparando-se os espectrogramas [3] do sinal de voz original com o da voz sintetizada [85], verifica-se que o CELP insere ruído em frequências entre as harmônicas. Com isso, a eficiência da codificação dos sons vozeados é menor do que a dos sons não-vozeados. O leitor deve atentar para esta deficiência do CELP, que será objeto de maiores investigações nos próximos anos.

Em relação à quantização do PCA, Atal et al [92] fazem uma comparação entre as LSFs, o seno inverso dos coeficientes de reflexão e a razão log-área. Uma conclusão inusitada deste trabalho é que o desempenho das LSFs foi o pior dos três. Este resultado foi refutado com o decorrer das pesquisas [155], [168] e pelo próprio Atal [146].

A robustez do CELP a ruídos é comprovada em [90]. Este fator mostrou-se importante na comparação dos desempenhos de diversos codificadores na presença de ruído, onde o CELP mostrou-se superior [89].

Para alcançar uma redução na taxa de bits, mantendo-se a qualidade, são feitas duas propostas em [84] e [86]. Estes trabalhos baseiam-se em uma classificação do sinal de voz. Para cada classe é estabelecido um esquema de codificação diferente. Desta forma, Wang e Gersho [86] diminuem a taxa de 4.8 kbits/s para 3.6 kbits/s sem degradação na qualidade.

Na década de 80, o Departamento de Defesa dos EUA (DoD) já havia estabelecido um padrão para comunicações de segurança. Este codificador consiste em um vocoder LPC, operando à taxa de 2,4 kbits/s, conhecido mundialmente como vocoder LPC-10e. Em 1987, o DoD buscou aperfeiçoar a qualidade de voz de seu sistema de telefonia e iniciou um processo de seleção com o objetivo de definir um codificador operando à taxa de 4,8 kbits/s. As conclusões desta seleção são apresentadas

pela equipe de Tremain em [89] e soaram bastante encorajadoras em relação ao potencial do CELP. Dos dez codificadores apresentados, os três melhores classificados foram justamente os codificadores baseados no CELP. O codificador escolhido é detalhado em [95]. Basicamente, este CELP utiliza técnicas de redução de complexidade já mencionadas: tratamento do filtro PLA como "codebook" adaptativo, "codebooks" esparsos com deslocamento circular e utilização da técnica de correção dos pontos extremos. Este algoritmo é chamado de PFS-1016 ("Proposed Federal Standard 1016") ou CELP do DoD [95], [154], [181].

Em 1990 continuaram a surgir trabalhos abordando a redução de complexidade no CELP [100], [111], [105]. O aumento na qualidade da voz sintetizada também foi abordado [109], [110], [117]. Em [117] temos novamente críticas quanto ao desempenho do CELP na representação de sons vozeados a baixas taxas. Isto fez com que Ozawa [108] sugerisse utilizar o CELP apenas para sons não-vozeados e uma variante do codificador Multi-Pulso [32] para sons vozeados. Esta abordagem vem sendo seguida por outros grupos de pesquisa [131].

A deficiência do CELP em capturar a estrutura espectral harmônica decorre em parte do fato de que nem sempre o período de pitch é múltiplo do período de amostragem. Uma vantagem de utilizarmos o filtro PLA com mais de um coeficiente é a possibilidade de interpolação proporcionada pelos coeficientes [176]. Isto minimiza o problema em questão. Uma nova alternativa, apresentada em 1990 por dois grupos distintos, é o uso de valores fracionários para o atraso  $M$  do filtro PLA [106], [107]. Isto corresponde à criação de novas "codewords" para o "codebook" adaptativo. As novas "codewords" são obtidas através da interpolação de amostras da memória do filtro PLA. Esta técnica aumenta a qualidade da voz sintetizada para sons vozeados, principalmente para vozes com frequência de pitch alta. Atualmente, diversos codificadores incorporam esta técnica [158], [160].

Em relação à quantização dos coeficientes do PCA, 1990 foi marcado pela consolidação das LSFs como parâmetro para quantização destes coeficientes. Isto pode ser atestado pelos trabalhos [121], [122], [123]. Popularizou-se também o conceito de quantização transparente [146] já mencionado anteriormente. A quantização vetorial ainda não tinha amadurecido o suficiente para ser utilizada na quantização dos parâmetros do filtro PCA no CELP. A maioria dos trabalhos versavam sobre quantizadores escalares.

Dois trabalhos que refletem o sucesso que o CELP vem acumulando são [114] e [113]. Em [113] é detalhado o codificador VSELP ("Vector Sum Excited Linear Prediction"), vencedor da concorrência estabelecida pela TIA ("Telecommunications Industry Association") para escolha de um codificador para uso no sistema de telefonia móvel celular norte-americano [99], [172], [208]. Em [114] o CELP do DoD é aperfeiçoado, sendo descrita sua implementação em um DSP32C da AT&T. O VSELP já se encontra em operação nos telefones celulares norte-americanos, enquanto que o CELP do DoD é parte integrante de diversos sistemas de telecomunicações.

Segundo nosso conhecimento, a única concorrência para codificadores a baixas taxas perdida pelo CELP nos últimos anos foi na escolha de um codificador para o sistema INMARSAT [132]. O

codificador escolhido foi um vocoder com excitação por bandas de frequência chamado IMBE ("Improved Multi-Band Excitation"), proposto por Griffin e Lim [75], [164]. Existem diversos grupos de pesquisa aperfeiçoando esta idéia [152], [169], [207].

Três dos mais importantes artigos publicados em 1990 são [101], [102] e [103]. Em [103] é apresentada uma visão unificada dos codificadores híbridos. Os autores deste trabalho são os idealizadores do codificador Auto-Excitado [47]. Neste trabalho, Rose e Barnwell III concluem que o codificador Auto-Excitado apresenta desempenho similar ao CELP. Este resultado pode ser questionado, quando comparado com a conclusão a que chegou a equipe do DoD [89].

Os outros dois artigos referem-se a métodos de redução da complexidade computacional. Em [101], Atal e Trancoso aprofundam o trabalho apresentado em [48], já comentado anteriormente. O trabalho de Kleijn et al [102] é uma excelente revisão dos métodos de redução de complexidade. Sua leitura é indispensável aos que estudam o CELP. Em [102] é apresentada uma valiosa comparação entre os diversos métodos de redução de complexidade propostos até 1988.

Em 1991, o VSELP é aperfeiçoado no trabalho de Gerson e Jasiuk [138]. O ruído inter-harmônicas, já mencionado como uma das deficiências do CELP, é combatido em [138] utilizando-se um filtro para enfatizar o erro nas frequências harmônicas [117]. Os autores incorporam também o uso de valores fracionários para o atraso  $M$  [106], [107]. As alterações propostas, aumentam a qualidade da voz e a complexidade computacional desta nova versão, em relação ao VSELP original. Por isto, em [138] também são feitas propostas para a redução desta complexidade.

A questão do ruído inter-harmônicas é abordado de forma elucidativa em [134]. Por razões práticas, a excitação ideal  $e(n)$  (Fig. 2.2) nunca será alcançada pelo codificador. O erro entre a excitação sintetizada  $d(n)$  (Fig. 2.15) e a excitação ideal  $e(n)$ , inevitavelmente surgirá como um ruído de quantização. Como a memória do filtro PLA ("codebook" adaptativo) é atualizada a partir da excitação sintetizada  $d(n)$ , haverá conseqüentemente uma realimentação do ruído de quantização no sistema. Esta realimentação do ruído de quantização corresponde a uma parcela considerável do ruído inter-harmônicas.

Quando o "codebook" adaptativo contém uma boa representação da periodicidade da excitação, devemos proteger este "codebook" da indesejável realimentação do ruído de quantização. Isto poderia ser feito, minimizando-se a ação do "codebook" fixo. Por outro lado, se a representação da excitação proporcionada pelo "codebook" adaptativo é deficiente (como em uma transição de um som não-vozeado para um som vozeado), desejamos que este "codebook" se adapte rapidamente, de forma a incorporar os pulsos referentes ao pitch. Para que isto aconteça, a ação do "codebook" fixo deve ser enfatizada. Observamos que a discussão acima conduz a duas atitudes contraditórias.

Em [134] é apresentada uma proposta para contornar esta dificuldade. Esta proposta consiste em atualizar a memória do "codebook" adaptativo, utilizando um ganho  $G'$  para a "codeword" do "codebook" fixo, diferente do ganho  $G$  empregado na geração da excitação (Fig. 2.16). Este método conduz a uma degradação do desempenho do codificador em altas frequências. Por outro lado, as

baixas frequências, que são mais importantes para a percepção humana, alcançam melhor representação. Assim, a razão sinal/ruído segmentada (RSRSEG) média [176] decai com a aplicação deste método, enquanto a qualidade subjetiva aumenta [134].

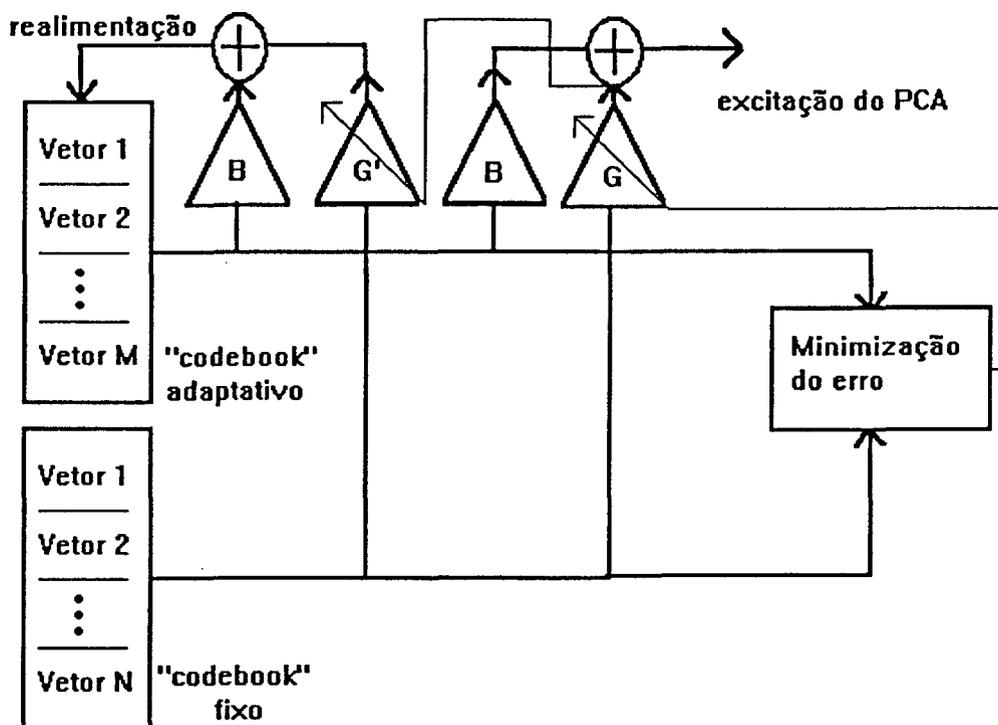


Fig. 2.16- Esquema para diminuição do ruído inter-harmônicas [134]

A segunda idéia apresentada em [134] consiste na ceifagem central do "codebook" adaptativo, de forma similar à utilizada na geração de "codebooks" fixos esparsos [147]. Segundo estes autores, pode-se adotar 50 % de amostras nulas com redução de apenas 0.5 dB na RSRSEG. Esta técnica não altera a qualidade subjetiva da voz; entretanto conduz a uma redução na complexidade computacional.

Em 1991, surgiram vários outros trabalhos apresentando propostas para aumentar a qualidade da voz sintetizada pelo CELP [140], [141], [143]. Alguns destes, buscam o aumento de qualidade através da alocação dinâmica de bits [144], [145], [150]. O sistema ATM, já mencionado, continua a estimular a operação com taxa variável. Propostas neste sentido são formuladas em [145] e [150].

Achamos que uma questão que merece maior esclarecimento refere-se à utilização de "codebooks" treinados [129] e sua comparação com o uso de "codebooks" simplificados, tais como os baseados em códigos algébricos [56], [60], [111]. Seria o resíduo da predição conjunta do PCA e do PLA próximo o bastante de um ruído branco para tornar desnecessária qualquer preocupação em sofisticar o "codebook" fixo? Em [129], Ozawa e Miyano relatam uma melhoria na qualidade da voz com a utilização de um "codebook" obtido com o algoritmo LBG [15]. No projeto deste

"codebook", utilizaram a definição de centróide apresentada em [64]. Os autores utilizaram também um método específico de busca da melhor excitação. Este método baseia-se em uma decisão atrasada na determinação da excitação [109]. Infelizmente, os autores não fazem uma diferenciação clara da importância de cada uma das duas técnicas no citado aumento da qualidade subjetiva da voz sintetizada.

Ainda em relação à obtenção de boa qualidade a baixas taxas, vale a pena ressaltar a abordagem de Atal et al em [131]. O codificador apresentado classifica a voz em intervalos (quadros) periódicos e não-periódicos. Para os intervalos não-periódicos é utilizado o CELP. Para os periódicos, a excitação é composta por pulsos. Este trabalho mostra uma forte tendência para a utilização de estruturas alternativas (diferentes do CELP) para melhorar a codificação de sons vozeados.

Em [172], Boisson de Marca e Eliana de Martino concluem o trabalho ressaltando duas linhas de pesquisa para o CELP: a codificação em classes e a diminuição da taxa relativa ao PCA através do uso da quantização vetorial. A primeira observação vai ao encontro do trabalho da equipe dos Laboratórios da Bell [131]. O direcionamento relacionado à segunda colocação pode ser observado em [146], [155], [156], [157]. Estes quatro últimos trabalhos abordam a quantização dos parâmetros do filtro PCA. Ambos têm em comum o uso de técnicas de quantização vetorial das LSFs para representação do filtro e a aplicação do conceito de quantização transparente [146].

Em [155] são feitas duas comparações importantes. A primeira delas é feita entre uma estrutura de quantização vetorial sub-ótima do tipo particionamento [146] e um quantizador vetorial sem restrições quanto à estrutura. O particionamento é utilizado para diminuir a complexidade computacional, mas aumenta a distorção média. Os autores chegam à conclusão de que a utilização do particionamento acarreta uma degradação equivalente a um bit na taxa de bits para a mesma distorção. Outra comparação importante diz respeito à aplicação de interpolação ao PCA. A Tabela 2.2 transcrita abaixo, indica a superioridade das LSFs em relação aos outros parâmetros.

Tabela 2.2- Avaliação da interpolação linear baseada na distorção espectral (SD) [155]

Parâmetro de representação do PCA	SD média	SD máxima
Coeficientes do filtro PCA	1.10	18.64
PARCOR	1.21	9.31
Seno inverso dos PARCOR	1.20	9.34
Autocorrelação normalizada	1.12	9.83
LSFs	1.05	9.25
Coeficientes cepstrais	1.05	31.20

Um trabalho interessante baseado na Teoria da Informação, é o de Hedelin e Bergström [136]. Os autores concluem que, através de um método de elevado custo computacional, pode-se quantizar

o ganho G do "codebook" fixo com um único bit. Isto é possível utilizando-se informações provenientes da primeira etapa de determinação da excitação, ou seja, da determinação dos parâmetros do filtro PLA ("codebook" adaptativo). Apesar do acentuado enfoque teórico, este trabalho é útil para a compreensão das características adequadas ao "codebook" fixo.

Nenhum dos trabalhos de 1991 citados até aqui aborda especificamente a redução de complexidade. Este assunto é objeto de atenção exclusiva de alguns poucos trabalhos como, por exemplo, [137]. A maioria dos trabalhos deste ano (relacionados à redução de complexidade) aborda a implementação do codificador como um todo [130], [139], [142], [158], [159], [160] englobando simultaneamente o aumento da qualidade da voz sintetizada, a diminuição de taxa e a redução da complexidade. Isto deve-se primordialmente ao fato de que o ETSI ("European Telecommunications Standards Institute") havia iniciado há alguns anos um processo de seleção de um codificador que opere à taxa de 6,5 kbits/s. Este codificador objetiva modernizar o sistema europeu GSM ("Groupe Speciale Mobile") de telefonia móvel celular, permitindo duplicar sua capacidade. Atualmente, o sistema GSM utiliza um codificador operando à taxa de 13 kbits/s [79]. Motivados por esta concorrência, os trabalhos citados (com exceção de [139]), apresentam codificadores candidatos a esta padronização. Estes artigos são repletos de informações de ordem prática, que buscam otimizar a implementação de codificadores CELP. Por constituírem-se de trabalhos voltados à implementação, os mesmos preocupam-se com o controle de erros na transmissão e incluem a codificação de canal.

Seria cansativa uma descrição detalhada de cada um destes trabalhos. Além disso, as técnicas utilizadas são, em sua grande maioria, derivações de técnicas já comentadas. Assim, relacionamos apenas as características mais relevantes de cada codificador nas Tabelas 2.3 e 2.4.

Tabela 2.3- Características dos codificadores em relação ao PCA

Quantização do PCA ****	Dinamarca [130]	Itália [142]	França [160]	Suécia [158]
Interpolação	Sim	Não	Sim *	Sim
Quantizador	Escalar	Escalar	Escalar	Escalar
Parâmetro	LSF	PARCOR **	PARCOR ***	LSF
Duração (quadro)	20 ms	20 ms	30 ms	20 ms
Bits p/ o PCA	40	38	37	34

\* Utiliza interpolação apenas para o primeiro subquadro, \*\* Seno Inverso do PARCOR, \*\*\* Razão log-área do PARCOR, \*\*\*\* Todos os filtros PCA são de décima ordem.

Tabela 2.4- Características dos codificadores em relação à excitação

Excitação	Dinamarca [130]	Itália [142]	França [160]	Suécia [158]
Bits do "codebook" fixo	10	7	24	11 bits
Tipo do "codebook" fixo	Esparso com deslocamento circular	Gaussiano	Pulsos binários regulares e pulsos simples	Pulsos binários regulares
Método de cálculo do "codebook" fixo	Correção dos pontos extremos	Autocorrelação	Próprio	Próprio
Duração do subquadro	5 ms	5 ms	7.5 ms	5 ms
Bits para o atraso M	7	7	8 bits	8 e 6 bits
Atrasos fracionários	Não	Não	Sim	Sim
Bits para o ganho do "codebook" adaptativo	4 bits	3 bits	4 bits	4 bits
Bits para o ganho do "codebook" fixo	5 bits	5 bits	5 bits	4 bits
Método de cálculo do "codebook" adaptativo	Correção dos pontos extremos	Correção dos pontos extremos	Próprio	Correção dos pontos extremos modificada
Taxa	7,2 kbit/s	6,3 kbits/s	6,7 kbits/s	6,9 kbits/s
RSRSEG média	-	9,07 dB	-	-
Complexidade	12,4 MACS	3,5 x GSM antigo	10,1 MIPS	13,8 MIPS
Otimização conjunta	Não	Não	Não	Sim

Gostaríamos de ressaltar duas características da Tabela 2.4. A primeira delas refere-se à utilização de "codebooks" formados por pulsos binários regulares. As "codewords" destes "codebooks" são compostas apenas de elementos -1, 0 e 1. O uso deste tipo de "codebook" faz com que a complexidade da análise relativa ao "codebook" adaptativo seja bem maior do que a relativa ao "codebook" fixo. Isto pode ser verificado na Tabela 2.5, onde a complexidade respectiva aos codificadores que utilizam Pulsos Binários Regulares é descrita em MIPS (milhões de instruções por segundo). Podemos observar também a maior complexidade do codificador sueco na determinação dos parâmetros do filtro PCA. Isto pode ser associado ao uso das LSFs.

Tabela 2.5- Complexidade de cada etapa em MIPS

<b>Etapas:</b>	<b>Suécia [158]</b>	<b>França [160]</b>
Cálculo do PCA	1,9	0,9
Codebook adaptativo	8,5	5,4
Codebook fixo	3,4	1,0
Cálculo do resíduo	-	1,1
Decodificador	1,5	1,7

A segunda característica da Tabela 2.4 que gostaríamos de ressaltar refere-se à otimização conjunta. O codificador sueco, que usa um "codebook" composto por pulsos binários regulares, realiza uma otimização conjunta das duas etapas de determinação da excitação. A otimização conjunta torna-se viável computacionalmente porque a busca da melhor "codeword" do "codebook" fixo apresenta complexidade reduzida para "codificadores" que utilizam pulsos binários regulares. Assim, determina-se o melhor par de "codewords" dos "codebooks" fixo e adaptativo, realizando-se buscas no "codebook" fixo para um pequeno número das melhores "codewords" do "codebook" adaptativo.

A avaliação de técnicas recentes ocasionalmente apresenta fragilidades. Isto pode ser observado, por exemplo, nas opiniões divergentes em relação ao desempenho proporcionado pela otimização conjunta. Em [139], os autores comentam que a otimização conjunta conduz a um significativo aumento na qualidade da voz sintetizada. Contudo, em [160] os autores expressam a opinião de que o pequeno aumento na qualidade na voz sintetizada devido à otimização conjunta não compensa o aumento da complexidade computacional.

## 2.4 - CONCLUSÕES

O estudo do codificador CELP não é uma tarefa fácil. O próprio algoritmo deste codificador incorpora diversas técnicas, cujas programações são deveras complexas. Outro problema decorre das diferentes implementações pelos diversos grupos de pesquisa. A divulgação de alguns padrões nos últimos anos como, por exemplo, o VSELP [99] e o CELP do DoD [154], facilitam um pouco a tarefa de implementação de um codificador CELP de referência.

Atualmente, o CELP encontra-se consagrado. Contudo, existem diversas vertentes buscando outras soluções. Dentre estas, uma das mais importantes para aplicações que exijam baixas taxas, é o

vocoder multi-bandas (IMBE) [164]. O IMBE apresenta-se como um sério concorrente para o CELP, sendo merecedor da atenção dos pesquisadores.

Podemos delinear claramente uma linha de pesquisa interessada em utilizar o CELP apenas na codificação de sons não-vozeados [101], [131], [196]. Para codificar os sons vozeados seriam utilizadas outras técnicas de codificação. Esta estratégia requer a classificação do sinal de voz em sons vozeados e não-vozeados [198]. Para esta tarefa, podem ser utilizados os conhecimentos adquiridos durante as pesquisas em relação ao vocoder LPC. Uma abordagem mais ambiciosa é a classificação do sinal de voz em um número maior de classes [197]. As pesquisas nesta direção referem-se a este tipo de classificação utilizando o termo classificação fonética [86], [117], [189], [195].

Um dos aspectos importantes que podem ser vislumbrados a partir desta revisão bibliográfica, diz respeito à utilização da quantização vetorial na codificação do filtro LPC no CELP (filtro PCA). Frequentemente o preço advindo do aumento da complexidade computacional torna-se um fator secundário face à necessidade de redução da taxa de bits. A telefonia móvel celular é um exemplo de aplicação que exige de forma imperiosa a diminuição na taxa de bits. O interesse comercial fomentado pelos sistemas norte-americano e europeu de telefonia celular gera uma evolução vertiginosa das pesquisas em codificação de voz a baixas taxas de bits com qualidade a nível de telefonia. Os norte-americanos, por exemplo, desejam uma voz sintetizada de boa qualidade a partir de aproximadamente 4 kbits/s, uma vez que a taxa total, incluindo o controle de erros no canal, deve ser de apenas 6,5 kbits/s. Neste contexto, a quantização escalar do filtro LPC mostra-se inadequada. Este fato impulsiona as pesquisas em relação à quantização vetorial do filtro LPC para o CELP.

# CAPÍTULO 3

## QUANTIZAÇÃO VETORIAL DO FILTRO LPC

### 3.1 - INTRODUÇÃO

Nosso objetivo neste capítulo é apresentar os conceitos básicos da quantização vetorial (QV). Enfatizaremos a quantização vetorial dos parâmetros do filtro LPC no codificador CELP. Existem diversas outras referências bibliográficas que complementam as informações apresentadas neste capítulo. Os leitores interessados em um embasamento teórico mais aprofundado a respeito da QV podem recorrer ao livro de Berger [5]. No contexto de codificação de voz, os artigos [16], [22], [27] e [29] permitem uma visão geral da aplicação da QV nesta área. Em língua portuguesa, temos o texto [115], que faz uma revisão geral sobre a QV. Além deste, temos outros trabalhos enfocando a QV como, por exemplo, as dissertações de mestrado [148] e [153].

Este capítulo está organizado da seguinte forma: na seção 3.2 apresentaremos os fundamentos básicos de quantização escalar. O motivo para isto é que a QV pode ser vista como uma generalização da quantização escalar. Na seção 3.3 apresentaremos os principais fundamentos da quantização vetorial. As questões concernentes ao projeto dos "codebooks" são tratadas na seção 3.4. Na seção 3.5 abordaremos a quantização vetorial "product-code" generalizada. Esta formulação teórica foi proposta recentemente e possibilita desenvolver um conceito unificado de diversos métodos de QV. Na seção 3.6 será apresentada uma breve revisão acerca das principais medidas de distorção utilizadas no processo de codificação. Na seção 3.7 apresentaremos o conceito de "Line Spectral Frequencies". As seções 3.2, 3.3, 3.5 e 3.7 são basicamente revisões de assuntos bem estabelecidos na literatura. O motivo da inclusão dessas seções é tornar este trabalho autocontido. As conclusões deste capítulo serão apresentadas na seção 3.8.

### 3.2 - FUNDAMENTOS DA QUANTIZAÇÃO ESCALAR

Dada uma variável  $X$ , um quantizador escalar deve representá-la utilizando  $b$  bits. Esta representação equivale à determinação do valor quantizado  $\hat{X}$ , onde  $\hat{X} = Q_E^b\{X\}$ . A partir dos  $b$  bits disponíveis, dispomos de  $2^b$  níveis de quantização para  $\hat{X}$ . O erro quadrático  $e^2 = (\hat{X} - X)^2$  é uma das medidas mais utilizadas para avaliar a precisão desta quantização.

A quantização escalar uniforme é uma forma simples de representar  $X$  através de  $b$  bits. No quantizador uniforme, os  $2^b$  valores de  $\hat{X}$  apresentam o mesmo espaçamento  $\Delta$  entre si. Cada um destes  $2^b$  valores é representado por um índice binário de  $b$  bits. Para projetarmos um quantizador uniforme, devemos inicialmente determinar o intervalo de ocorrência da variável  $X$ , ou seja, seus

valores máximo  $X_{\text{máx}}$  e mínimo  $X_{\text{mín}}$ . O valor de  $\Delta$ , conhecido como passo de quantização, é dado por  $\Delta = (X_{\text{máx}} - X_{\text{mín}}) / (2^b - 1)$ . O numerador desta expressão corresponde à faixa dinâmica de  $X$ . A subtração no denominador indica que devemos reservar um dos  $2^b$  índices disponíveis para representar a referência. Normalmente, a referência inicial corresponde ao valor de  $X_{\text{mín}}$ . Os  $(2^b - 1)$  índices restantes correspondem ao número de passos, contabilizados a partir do valor da referência inicial. Por exemplo, seja  $X$  um número real na faixa de  $1 \leq X \leq 2$ . Um quantizador uniforme de 2 bits dispõe das seguintes combinações de bits: "00", "01", "10" e "11". Poderíamos atribuir o valor  $X_{\text{mín}} = 1$  ao índice "00". Considerando  $\Delta = 0,33$ , o quantizador corresponderia à Tabela 3.1.

Tabela 3.1 - Exemplo de quantização uniforme

Índice	00	01	10	11
Valor literal de $\hat{X}$	$X_{\text{mín}}$	$X_{\text{mín}} + \Delta$	$X_{\text{mín}} + 2\Delta$	$X_{\text{mín}} + 3\Delta$
Valor numérico de $\hat{X}$	1	1,33	1,66	1,99

Podemos supor que este quantizador é utilizado em um sistema de telecomunicações, onde o transmissor deve enviar o valor  $X=1,4$  ao receptor. O primeiro passo seria encontrarmos o índice correspondente ao valor numérico de  $\hat{X}$  que mais se aproxima de  $X$ . A Tabela 3.1 poderia ter sido pré-calculada e armazenada em memória ROM. Muitas vezes o mesmo processador que executa o "software" de transmissão, executa também o "software" de recepção. Assim, esta tabela poderia ser compartilhada, de forma a economizar memória de armazenamento. Neste tipo de implementação, bastaria percorrermos a tabela para determinarmos o índice  $i$  que acarrete o menor erro quadrático  $e_i^2 = (\hat{X}_i - X)^2$ . Em nosso exemplo, seria transmitido o índice "01" para representar  $X=1,4$ . O receptor faria a operação inversa, atribuindo o valor 1,33 a este índice. Esta operação é uma simples consulta a uma tabela. Nestas condições, a tabela é conhecida como "look-up table".

Para quantizadores com um número elevado de bits, o armazenamento desta tabela exige uma quantidade significativa de memória. Uma forma alternativa de implementação seria armazenar apenas o valor de referência (geralmente  $X_{\text{mín}}$ ) e o valor de  $\Delta$ . O valor de  $\hat{X}_i$ , associado ao índice  $i$ , é então calculado tanto no codificador quanto no decodificador. Nesta implementação, o sistema deveria determinar o número  $n$  de passos que conduzisse a  $X = X_{\text{mín}} + n\Delta$ . Contudo, geralmente não existe um valor inteiro de  $n$  que leve exatamente ao valor  $X$ . O valor inteiro mais próximo é dado por  $n = \text{TRUNC}\{0,5 + (X - X_{\text{mín}}) / \Delta\}$ , onde TRUNC representa a função de truncamento. A soma de 0,5 é um artifício para que este truncamento corresponda a um arredondamento [38]. O receptor determina o valor de  $\hat{X}$  a partir de  $\hat{X} = X_{\text{mín}} + n\Delta$ .

A construção da Tabela 3.1 pode suscitar diversos questionamentos. No caso de um quantizador escalar que utilize uma "look-up table", por que não organizar esta tabela de forma independente de  $X_{\text{mín}}$  e  $\Delta$ ? Esta é a idéia básica da quantização escalar não-uniforme. A Tabela 3.2

mostra um exemplo de quantização não-uniforme, onde os valores de  $\hat{X}$  são propositadamente aleatórios.

Tabela 3.2 - Exemplo de quantização não-uniforme

Índice	00	01	10	11
Valor numérico de $\hat{X}$	1,2	1,665	1,38	1,9999

A idéia da quantização não-uniforme é reforçada pelo fato de que para fins de taxa de transmissão, não importa o valor numérico de  $\hat{X}$ , pois é transmitido apenas o índice correspondente. Tanto para a Tabela 3.1 quanto para a Tabela 3.2, a taxa seria de 2 bits para cada valor de  $\hat{X}$  transmitido, ou seja,  $t=2$  bits/amostra. Ao utilizarmos "look-up tables", a precisão nos valores de  $\hat{X}$  deve respeitar apenas condições de armazenamento de memória. Já ressaltamos anteriormente que muitas vezes isto é um fator significativo.

O projeto de um quantizador não-uniforme freqüentemente corresponde à construção destas tabelas de acordo com a função densidade de probabilidade (FDP) [1] da variável  $X$ . Por exemplo, caso  $X$  apresente uma FDP gaussiana, devemos utilizar um quantizador não-uniforme gaussiano [167]. Neste caso, concentramos a maior parte dos pontos da tabela em torno do valor médio destes pontos.

Na prática, muitas vezes não dispomos da FDP de  $X$ . Daí, podemos utilizar um histograma de ocorrências para estimar a FDP. Vamos supor que o levantamento do histograma de ocorrências da variável  $X$  corresponda ao gráfico apresentado na Fig.3.1. É razoável aceitar que um quantizador uniforme como o da Tabela 3.1, não represente uma alocação adequada para os níveis do quantizador. Na verdade, o quantizador uniforme só irá apresentar bom desempenho quando  $X$  tiver uma FDP aproximadamente uniforme.

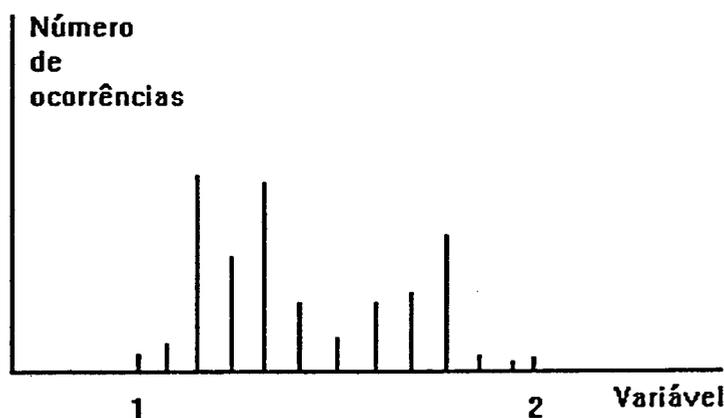


Fig. 3.1 - Exemplo de histograma

Quando não dispomos da FDP de  $X$ , devemos basear o projeto do quantizador apenas em uma seqüência de ocorrências de  $X$ . O trabalho de Lloyd (republicado em [25]) apresenta justamente uma solução para esta questão. O algoritmo de Lloyd determina a tabela de quantização a partir da seqüência de ocorrências (seqüência de treino) disponível. Veremos adiante que a partir do trabalho de Lloyd, formulou-se o algoritmo mais utilizado atualmente em quantização vetorial: o LBG [15].

### 3.3 - FUNDAMENTOS DA QUANTIZAÇÃO VETORIAL

Na discussão sobre quantização escalar, exemplificamos a transmissão da variável quantizada  $\hat{X}$ . Todavia, os sistemas de telecomunicações não transmitem apenas uma variável, mas sim uma seqüência  $\hat{X}(n)$  das mesmas. Esta seqüência pode representar um sinal de voz digitalizado (PCM), os coeficientes de um polinômio, etc. De forma geral, estes dados podem ser vistos como vetores. Os vetores são a forma natural de manusearmos os sinais em sistemas de processamento digital e em simulações dos mesmos [39].

Em uma transmissão ininterrupta de voz, por exemplo, estes vetores poderiam ter um número infinito de elementos. Entretanto, podemos seccionar este sinal em um determinado número  $K$  de amostras e representar cada um destes segmentos por vetores de dimensão  $K$ . Este é o passo inicial para que possamos submeter cada um destes segmentos a uma quantização vetorial (QV).

O motivo principal para utilizarmos a QV é a inerente diminuição da taxa de bits que esta técnica pode proporcionar. A Teoria da Distorção pela Taxa indica que a QV é o caminho natural para a obtenção de desempenhos, em termos de taxa e distorção, próximos aos limites teóricos [5]. Já citamos no Capítulo 2 que para uma dada distorção, a QV resultará em uma taxa de bits menor do que a taxa respectiva à quantização escalar. Nosso objetivo nesta seção é justificar esta afirmação através de exemplos simples.

Podemos considerar a QV como a extrapolação da idéia da quantização escalar não-uniforme. Assim, cada elemento da tabela de quantização representa agora um vetor de  $K$  amostras ao invés de um escalar. Um quantizador vetorial mapeia um vetor  $X$  de entrada com dimensão  $K$ , através de um vetor reprodução  $\hat{X}_i$ , também de dimensão  $K$ . O vetor  $\hat{X}_i$  é escolhido dentre  $N$  vetores ("codewords" ou palavras-códigos) de um dicionário de códigos também chamado "codebook". A eficiência do sistema de QV reside no fato de ser necessário transmitir apenas o índice  $i$  da "codeword" escolhida.

Em nosso sistema de telecomunicações fictício, devemos agora quantizar um vetor  $X$  de duas amostras (dimensão  $K=2$ ); por exemplo,  $X=\{1,4;1,7\}$ . Um quantizador para vetores deve ser definido através de uma tabela semelhante às Tabelas 3.1 e 3.2. A Tabela 3.3 apresenta uma alternativa possível, onde são realizadas todas combinações possíveis entre os elementos da Tabela 3.1.

Tabela 3.3 - Exemplo de quantização vetorial

Índice	0000	0001	0010	0011	0100	0101	0110	0111
Valor de $\hat{X}_i$	{1;1}	{1;1,33}	{1;1,66}	{1;1,99}	{1,33;1}	{1,33;1,33}	{1,33;1,66}	{1,33;1,99}
Índice	1000	1001	1010	1011	1100	1101	1110	1111
Valor de $\hat{X}_i$	{1,66;1}	{1,66;1,33}	{1,66;1,66}	{1,66;1,99}	{1,99;1}	{1,99;1,33}	{1,99;1,66}	{1,99;1,99}

Para determinarmos qual a "codeword"  $\hat{X}_i$  que melhor representa  $X_k$ , definimos uma medida de distorção  $d(X_k, \hat{X}_i)$ . Esta medida indica o grau de semelhança entre dois vetores [14], [24], [83]. A "codeword" escolhida é a que conduzir à menor distorção. A medida de distorção mais utilizada em QV é o erro quadrático (EQ), dado por:

$$EQ = \|X_k - \hat{X}_k\|^2 = \sum_{j=1}^K (X_j - \hat{X}_j)^2.$$

onde  $\| \cdot \|$  representa a norma euclidiana de um vetor.

Suponhamos que a quantização escalar de um número elevado de amostras a partir da Tabela 3.1 corresponda a um erro quadrático médio (EQM),  $\bar{E}$ . Podemos afirmar que a QV através da Tabela 3.3 conduzirá a um EQM menor ou igual a  $\bar{E}$ . Isto ocorre porque a Tabela 3.1 está contida na Tabela 3.3. Neste caso, a taxa de bits da QV (Tabela 3.3) é de 4 bits para 2 amostras ( $t=2$  bits/amostra). Esta é a mesma taxa da Tabela 3.1. Podemos concluir desta argumentação que a QV obtém um desempenho, em termos da relação entre taxa e distorção, no mínimo equivalente à quantização escalar.

Façamos outras considerações no intuito de definir o motivo pelo qual a QV sobrepuja a quantização escalar. Suponhamos que o vetor  $X$  corresponda a amostras de um sinal discreto  $X(n)$ , cuja natureza física imponha que não haja variações bruscas entre amostras consecutivas. Por simplicidade, suponhamos que a variação entre amostras consecutivas deste sinal (e conseqüentemente do vetor  $X$ ) é sempre de  $\pm 0,33$ . Isto implica em uma probabilidade nula de ocorrência para diversos vetores da Tabela 3.3. Vetores como  $X=\{1;1\}$  e  $X=\{1;1,66\}$ , por exemplo, não ocorreriam jamais. A Tabela 3.4 corresponde à eliminação dos vetores da Tabela 3.3 que não ocorrem em  $X(n)$ . Observamos que a taxa foi reduzida de  $t=2$  bits/amostra (Tabela 3.3) para  $t=1,5$  bits/amostra (Tabela 3.4), com a garantia de que a distorção não se alterou.

Tabela 3.4 - Exemplo da possível diminuição da taxa através da QV

Índice	000	001	010	011	100	101	110	111
Valor de $\hat{X}_i$	{1;1,33}	{1,33;1}	{1,33;1,66}	{1,66;1,33}	{1,66;1,99}	{1,99;1,66}	-	-

Os exemplos acima apresentam de maneira informal a vantagem que a QV obtém ao explorar a correlação [1] existente entre as amostras do sinal  $X(n)$ . A QV sempre apresenta desempenho superior à quantização escalar em termos do compromisso entre taxa de bits e distorção [5]. Contudo, devemos atentar para o fato de que a QV acarreta sempre uma maior complexidade computacional.

A complexidade da QV pode ser classificada de acordo com a Fig. 3.3. A etapa de projeto do quantizador é realizada "off-line". Quando estamos realizando simulações experimentais que envolvam o projeto de diversos quantizadores, a complexidade desta etapa torna-se um fator significativo. Contudo, uma vez projetado o quantizador, o custo de sua implementação é determinado apenas pela memória e pelo número de cálculos necessários para selecionar a melhor "codeword" (vizinho mais próximo) [175]. Com isto, a complexidade da etapa de operação do quantizador é geralmente a mais relevante.

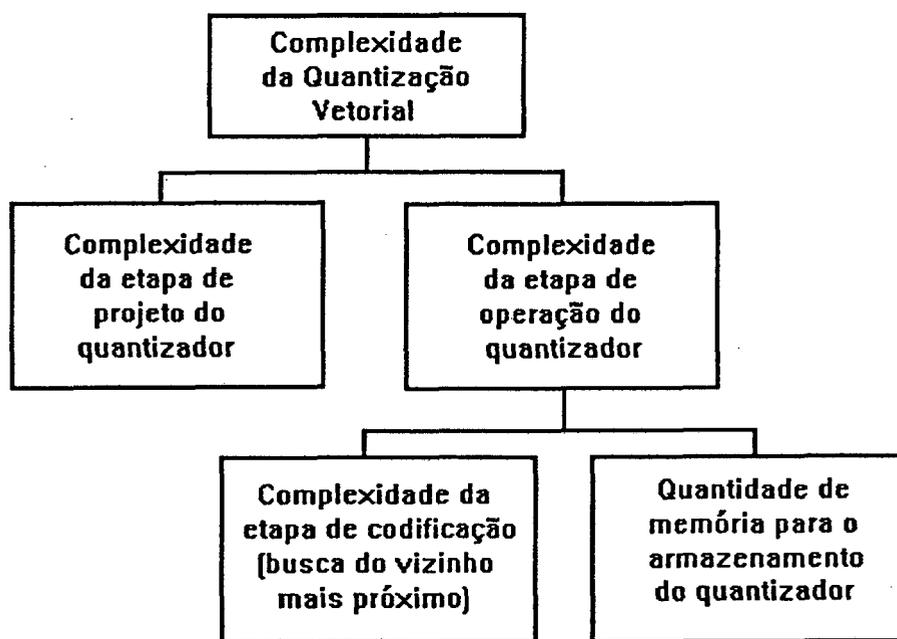


Fig. 3.3 - Complexidade da QV

Para o armazenamento de um "codebook" com  $N$  "codewords" de dimensão  $K$  necessitamos de  $NK$  endereços de memória. Quando utilizamos o erro quadrático como medida de distorção, a busca exaustiva da melhor "codeword" requer  $NK$  operações de multiplicação e acumulação (MAC). São necessários  $b = \log_2 N$  bits para representar os índices das "codewords". A taxa de bits por amostra do sinal codificado a partir deste "codebook" é  $t=b/K$ . Para uma determinada taxa de bits por

amostra, podemos reduzir a distorção aumentando  $K$  e  $N$ . Contudo, para manter  $t$  constante, o número de vetores  $N = 2^k$  aumenta exponencialmente. O aumento de  $N$  implica em aumento de complexidade computacional na etapa de busca da melhor "codeword" e um aumento também na quantidade de memória necessária para o armazenamento do "codebook".

### 3.4 - ALGORITMOS PARA O PROJETO DE "CODEBOOKS"

Até este momento ainda não discutimos a respeito dos algoritmos utilizados para o projeto do "codebook". Na maior parte dos casos, não dispomos da FDP conjunta do vetor  $X$  a ser quantizado. Geralmente, os algoritmos de projeto do "codebook" utilizam um número razoavelmente grande de vetores para estimar a estatística de  $X$ . Este conjunto de vetores é conhecido como seqüência de treinamento (ST). Estima-se que para cada "codeword" do "codebook", devemos dispor de 20 a 1000 vetores na ST [190]. Além da ST, os algoritmos de projeto do "codebook" exigem a definição de uma medida de distorção [14], [24], [83]. A partir disto, os algoritmos determinam o "codebook" através de processos iterativos.

#### 3.4.1 - CONSIDERAÇÕES SOBRE O ALGORITMO LBG

O algoritmo mais utilizado em QV para o projeto de "codebooks" é o LBG [15]. Seu nome corresponde às iniciais dos autores do artigo [15]: Linde, Buzo e Gray. O algoritmo LBG é uma extrapolação do trabalho de Lloyd [25], para o caso de vetores de dimensão  $K$  [177]. Por este motivo, o LBG também é conhecido como algoritmo "K-means" ou GLA. O termo GLA ("Generalized Lloyd Algorithm") é utilizado principalmente pela equipe de Allen Gersho [178], [179], [180].

Faremos, a seguir, uma descrição sucinta de uma implementação do algoritmo LBG. Nosso objetivo é apresentar uma visão prática do algoritmo de forma a simplificar a tarefa daqueles que queiram, porventura, implementá-lo. Além disso, desejamos estabelecer o importante conceito de centróide. Apesar da apresentação do algoritmo estar estruturada na forma de um programa, sua descrição é puramente conceitual. Outros trabalhos podem complementar as informações acerca do LBG [15], [148], [153], [179].

#### **Algoritmo LBG**

##### *Passo 1- Inicialização:*

$N$  - número de "codewords" do "codebook".

$K$  - dimensão das "codewords" e dos vetores da seqüência de treinamento.

$LIMAR$  - limiar de parada do algoritmo, por exemplo, 1 %.

*DIS* - distorção média. Calculada a partir da distorção de cada vetor da seqüência de treinamento.

*DISANT*- distorção média da iteração anterior. Assume um valor inicial bastante elevado (ex.:  $10^{30}$ ).

*CBK[N][K]* - armazena o "codebook" :  $N$  "codewords" de dimensão  $K$ . Pode ser inicializado com números randômicos.

*SEQ[VT][K]* - seqüência de treinamento:  $VT$  vetores de dimensão  $K$ .

*AUX[VT]* - vetor auxiliar: armazena o índice da "codeword" correspondente a cada vetor da seqüência de treinamento.

#### *Passo 2- Codificação da seqüência de treinamento:*

De acordo com a medida de distorção definida, buscar para cada um dos  $VT$  vetores qual a "codeword" de *CBK* que melhor o representa. Armazenar esta informação no vetor auxiliar *AUX*. Calcular o valor da distorção média e armazenar este valor em *DIS*.

#### *Passo 3- Teste de parada do algoritmo:*

A partir do valor de *LIMIAR*, determinar se o algoritmo não está mais apresentando diminuição significativa na distorção média. Este teste é feito através da comparação de  $(DISANT - DIS) / DIS$  com *LIMIAR*. Se o primeiro termo for maior do que *LIMIAR*, significa que o algoritmo ainda está convergindo e deve continuar com o passo 4. Caso contrário, o processo termina com o "codebook" final sendo dado pelas  $N$  "codewords" de *CBK* neste momento.

#### *Passo 4- Atualizar o "codebook":*

De acordo com a informação de *AUX*, sabe-se quais os vetores que cada "codeword" representa. Para cada "codeword", corresponde um grupo de vetores de *SEQ*. Determinar para cada um destes grupos, uma nova "codeword". Esta nova "codeword" é a centróide deste grupo de vetores. A centróide deve ser o vetor que minimiza a distorção média para o respectivo grupo de vetores. A definição da expressão da centróide é vinculada à medida de distorção utilizada. No caso do *EQ*, a centróide é o centro de gravidade dos vetores do grupo. Uma vez calculado o novo "codebook", atualizar *CBK* e retornar ao passo 2.

Existem diversas formas de determinar o "codebook" inicial para o LBG [15], [97], [192]. A forma mais utilizada é a apresentada em [15] e conhecida como "splitting".

A principal responsável pela complexidade computacional do LBG é a etapa de codificação da ST (passo 2). O processo iterativo no qual se baseia o LBG, exige que a cada iteração do algoritmo

toda a ST seja codificada. Quando a medida de distorção é o erro quadrático, esta codificação requer  $VT \cdot N \cdot K$  operações MAC, onde VT é o número de vetores da ST. Por exemplo, para  $N=1024$ ,  $K=10$  e  $VT=20.480$  vetores, seriam necessárias 209 milhões de MAC's para codificar a ST em cada iteração. Esta estimativa não inclui o "overhead" associado.

Quando se usa o erro quadrático como medida de distorção, podemos reduzir o número de cálculos necessários, incorporando o uso de distâncias parciais [40] ao LBG. Isto permite reduzir o tempo de projeto em mais de 50 %.

### 3.4.2 - UTILIZAÇÃO DE OUTROS ALGORITMOS

Apesar do LBG ser o algoritmo mais utilizado em QV, existem diversos outros algoritmos propostos na literatura. A seguir descreveremos os principais métodos de projeto de "codebooks".

#### 3.4.2.1 - Algoritmo PNN:

O PNN ("Pairwise Nearest Neighbor") é um algoritmo para o projeto de "codebooks" apresentado em [57], [97]. Em [177], é feita uma revisão sobre a perspectiva histórica do PNN. Relata-se que o conceito no qual se baseia o PNN é similar ao de um método de agrupamento estatístico, publicado em 1963.

A principal vantagem do PNN em relação ao LBG é que sua complexidade computacional é significativamente menor, quando se utiliza o erro quadrático como medida de distorção [192]. Por outro lado, o "codebook" projetado pelo PNN normalmente acarreta uma distorção média um pouco maior do que a do LBG. O PNN é muito útil na fase de testes que envolvam um grande número de simulações. Quando desejamos um resultado superior, devemos utilizar o LBG para refinar o "codebook" gerado pelo PNN [97].

Devemos ressaltar que nem sempre a medida de distorção utilizada consegue expressar adequadamente a qualidade subjetiva. A partir deste enfoque, torna-se mais difícil a comparação entre o PNN e o LBG. Em [84], o PNN é utilizado para gerar o "codebook" correspondente à excitação no CELP. Os autores argumentam que o LBG acarreta uma suavização excessiva das "codewords" que representarão o sinal. Enquanto isso, o PNN consegue preservar melhor as características impulsivas do sinal, importantes para o objetivo perseguido pelos autores.

#### 3.4.2.2 - Redes Neurais:

Uma rede neural artificial é capaz de executar as funções de um quantizador vetorial [94], [170]. As redes neurais são muito utilizadas em classificação de padrões [94]. Todavia, sua aplicação em QV ainda não foi muito explorada. Este tema vem merecendo bastante atenção atualmente [120], [124], [161], [193], [195], [200], [201].

Existem basicamente duas técnicas para o treinamento de redes neurais: o treinamento supervisionado e o não-supervisionado [94]. Em algumas aplicações, podemos obter compressão de taxa de bits utilizando redes com treinamento supervisionado. Em [124] são utilizadas redes "perceptron" e o método de treinamento supervisionado "back-propagation", para a compressão de sinais de eletrocardiogramas. Todavia, as redes neurais mais utilizadas em QV são as que dão suporte ao treinamento não-supervisionado [94]. Estas redes são capazes de realizar as três etapas da QV: o projeto do "codebook", a codificação e a decodificação. Destas redes, a mais popular é a rede de Kohonen ou SOM ("Self-Organizing Maps") [65], [94], [170].

A etapa de treinamento da rede de Kohonen corresponde à etapa de projeto do "codebook". Uma vez treinada, a rede fornece em sua saída a "codeword" mais semelhante ao vetor apresentado em sua entrada. Esta "codeword" corresponde aos valores das sinapses referentes ao neurônio ativado [65], [170].

Em [161] é feita uma abordagem comparativa entre o LBG, a rede de Kohonen e as redes propostas pelos autores. Os resultados obtidos mostram que o LBG apresenta menor distorção do que a obtida com a rede de Kohonen, enquanto a técnica proposta situa-se próxima ao LBG em termos de distorção. Em [119] os resultados obtidos com redes neurais são tidos como comparáveis aos obtidos com o LBG. Em [180] são propostas modificações na rede de Kohonen e os resultados obtidos são superiores aos obtidos com o LBG.

Uma observação importante é a potencialidade do uso de redes neurais em QV adaptativa [119], [120], [184], [185]. Em [195], a rede de Kohonen é utilizada no contexto do codificador CELP. Nesta aplicação a rede executa três tarefas: classifica foneticamente o sinal, diminui a taxa de bits e diminui a complexidade da etapa de codificação da QV. Em [151], os autores utilizam uma rede de Hopfield na etapa de seleção da melhor "codeword" do "codebook" fixo no codificador CELP.

#### 3.4.2.3 - "Simulated Annealing":

O algoritmo LBG não garante a determinação do mínimo global da função de distorção, ou seja, não atinge garantidamente o "codebook" ótimo [192]. Podemos tentar contornar este problema, inicializando o algoritmo LBG com diferentes "codebooks" iniciais e selecionando o resultado correspondente à menor distorção média. Contudo, a complexidade computacional desta técnica é bastante elevada. Além disso, a determinação do mínimo global continua a não ser garantida [192]. Devido a esta limitação do LBG surgiram métodos que buscam obter o "codebook" ótimo [80], [179], [180].

Em [192] é feita uma comparação entre diversos métodos para o projeto de "codebooks", entre eles o LBG, o PNN e o "simulated annealing" [80]. Uma conclusão importante de [192] é que a melhoria obtida com o "simulated annealing" [80], em relação ao LBG, não parece justificar o significativo aumento na complexidade computacional.

### 3.5- QUANTIZAÇÃO VETORIAL "PRODUCT-CODE" GENERALIZADA

A utilização de quantizadores vetoriais com estruturas restritivas como, por exemplo, multi-estágio, degrada o desempenho da QV em termos de taxa de bits e distorção [178]. Isto porque qualquer uma das estruturas de codificação de um vetor randômico de dimensão  $K$ , usando um número determinado de  $b$  bits, são equivalentes a casos especiais da QV não-estruturada. Assim, os métodos de estruturação do quantizador são geralmente sub-ótimos [178]. Todavia, em alguns casos a utilização de quantizadores não estruturados é inviável devido à complexidade computacional como, por exemplo, na quantização vetorial do filtro LPC no codificador CELP. Contornamos a excessiva complexidade computacional impondo restrições às estruturas de quantização vetorial [178], e assim, diminuimos o número de cálculos da etapa de codificação e a quantidade de memória requerida [190]. Dentre as estruturas mais importantes podemos citar a árvore binária [36], [153], os quantizadores "product-code" [36], [31] e os quantizadores multi-estágio [36], [72]. A seguir descreveremos de forma sucinta cada uma destas técnicas.

#### 3.5.1 - "Product-code":

A quantização "product-code" é estruturada a partir de um produto de "codebooks" [153], [165]. Nesta técnica, o quantizador vetorial é organizado como o produto cartesiano de "codebooks" menores. Normalmente o termo "product-code" é associado com a QV ganho-forma ("shape-gain"), que é um caso particular do "product-code" [31]. O quantizador ganho-forma normaliza o vetor (representando o filtro LPC) a ser quantizado. A partir daí, o filtro LPC passa a ser representado por um escalar (ganho) e um vetor normalizado (forma). O ganho e o vetor-forma são quantizados a partir de "codebooks" próprios. A estrutura "product-code" permite redução na complexidade e na quantidade de memória para armazenamento do "codebook". No entanto, esta estrutura acarreta em aumento da distorção média em relação ao "codebook" não-estruturado [153].

#### 3.5.2 - Particionamento ou "Split":

Em [146], a técnica "product-code" é utilizada para quantizar vetorialmente os parâmetros do filtro LPC. O vetor correspondente ao filtro LPC é particionado em dois vetores menores. Cada um destes vetores é quantizado de forma independente. Para isto, projetamos dois "codebooks" diferentes. Esta técnica é conhecida pelo nome de QV "split" ou particionamento, constituindo-se em um caso particular da QV "product-code" [146].

#### 3.5.3 - Multi-estágio:

A QV multi-estágio pode ser vista como uma técnica de aproximações sucessivas. Para isto são projetados "codebooks" respectivos a cada estágio da quantização. Cada um destes "codebooks" é

formado por  $2^{b_i}$  "codewords" de dimensão K. Assim, os b bits disponíveis para a quantização, são distribuídos entre os L estágios:  $b = \sum_{i=1}^L b_i$ . O vetor X de entrada é quantizado a partir do primeiro estágio. Para isto, seleciona-se a melhor "codeword" do primeiro "codebook"  $\hat{y}_1 = Q_1\{X\}$ , onde  $Q_i\{\cdot\}$  simboliza a operação de quantização do estágio i. Determina-se o erro  $E_1$  (resíduo) desta primeira quantização:  $E_1 = \hat{y}_1 - X$ . O resíduo  $E_1$  é então quantizado usando o segundo "codebook". Desta forma, a cada estágio (com exceção do último) o resíduo da quantização é calculado e passado para o próximo estágio, onde será quantizado de forma independente dos outros estágios. O decodificador recebe para cada estágio o índice da "codeword" respectiva e determina o vetor  $\hat{X}$  a partir da soma destas "codewords".

A QV multi-estágio exige o armazenamento de  $\sum_{i=1}^L 2^{b_i}$  "codewords". Este número pode ser bem menor do que as  $2^b$  "codewords" relativas a um "codebook" não-estruturado [178]. Realizando-se uma busca sequencial, a complexidade de codificação, em termos de operações MAC, é similar à necessidade de memória de armazenamento:  $\sum_{i=1}^L 2^{b_i}$ .

### 3.5.4 - "Product-code" Generalizada:

As técnicas acima são casos particulares da QV "product-code" generalizada [175], [173]. A generalização do conceito de QV "product-code", possibilita enfocarmos diversas estruturas de quantização de acordo com esta terminologia. Dentre estas, estão as estruturas de QV mais utilizadas: árvore binária [36], [153], ganho-forma [36], [31], multi-estágio [36], [72] e particionamento [146]. O codificador CELP também pode ser interpretado a partir do enfoque proporcionado pela QV "product-code" generalizada [175]. Apresentaremos a seguir a generalização do conceito de QV "product-code" [175].

A QV "product-code" generalizada baseia-se na representação do vetor X a ser quantizado, por um conjunto de s características:  $f_1, f_2, \dots, f_s$ . Cada uma destas  $f_i$  características têm dimensão  $k_i$ , podendo ser um escalar ou um vetor. O vetor  $\hat{X}$  é sintetizado a partir das características através de uma função de síntese  $g$ :  $X = g(f_1, f_2, \dots, f_s)$ . Para cada característica  $f_i$  são alocados  $b_i$  bits, de

forma que  $b = \sum_{i=1}^s 2^{b_i}$ , onde b é o número de bits utilizados pelo quantizador. Cada característica  $f_i$  é quantizada com  $b_i$  bits ( $\hat{f}_i = Q^{b_i}\{f_i\}$ ) a partir de um "codebook" contendo  $2^{b_i}$  "codewords" de características. Estas "codewords" permitem sintetizar o vetor  $\hat{X}$  (que deve representar X) a partir de  $g$ :  $\hat{X} = g(\hat{f}_1, \hat{f}_2, \dots, \hat{f}_s)$ .

Na QV multi-estágio a função de síntese  $g$  é dada pela soma das "codewords" de características, onde os vetores de características são os resíduos dos estágios anteriores. A QV por particionamento é um caso particular da QV multi-estágio, onde os vetores de características possuem algumas componentes nulas. Desta forma, o somatório da função de síntese pode ser substituído por uma concatenação e a extração das características reduz-se ao particionamento do vetor a ser quantizado [173].

Assim, qualquer QV por particionamento pode ser realizada através de uma QV multi-estágio. Todavia, o contrário não é verdadeiro. Atal e Paliwal afirmaram que a QV por particionamento apresenta menor distorção do que a QV multi-estágio [146], [194]. Entretanto, esta afirmação é contestada em outros artigos [173], [183]. Em termos do compromisso entre taxa, distorção e complexidade, a QV por particionamento é uma opção bastante razoável em muitos casos [173]. A QV das LSFs ("Line Spectral Frequencies") é um bom exemplo. Constata-se isto através da ampla utilização da QV por particionamento na quantização das LSFs no codificador CELP [146], [155], [156], [157], [194].

### 3.6- MEDIDAS DE DISTORÇÃO

Especificadas a taxa de bits e a complexidade computacional, o objetivo dos métodos de compressão é minimizar a distorção entre a informação original e aquela reconstruída na etapa de decompressão. Todavia, torna-se muito difícil definir a distorção quando estão envolvidos fatores de percepção humana como a visão e a audição. Este é um grande problema em processamento de voz [14], [83], de imagens [55], de sinais biomédicos [82], [127] e outros.

De fato, existem dois meios diferentes de definir e medir distorções [22]. O primeiro deles fundamenta-se em uma noção subjetiva, perceptual ou qualitativa, baseada em distorções relativas obtidas a partir das opiniões de pessoas. Como exemplos, temos os testes MOS ("Mean Opinion Score"), DAM ("Diagnostic Acceptability Measure") e DRT ("Diagnostic Rhyme Test") [83], [202], amplamente utilizados em processamento de voz. A distorção subjetiva é de grande importância na avaliação de sistemas de compressão. Contudo, existe uma dificuldade muito grande em definir figuras de mérito objetivas que possam ser empregadas no projeto de um sistema e que garantam a minimização das distorções subjetivas [22].

O estabelecimento de uma figura de mérito objetiva requer a definição de uma medida quantitativa de distorção. Esta medida é uma função que atribui um valor não-negativo a um par de vetores em comparação. Baseando-nos nesta medida objetiva, podemos comparar quantitativamente sistemas diferentes. Além disso, dada uma medida que seja simples de manipular matematicamente, podemos projetar um sistema de forma a minimizar a distorção. Um exemplo do aproveitamento desta vantagem é o algoritmo LBG abordado anteriormente. Podemos concluir que os dois requisitos

básicos para uma medida quantitativa de distorção são a sua simplicidade matemática e sua consistência, ou seja, sua correlação com testes subjetivos.

Devido à sua simplicidade matemática, a medida quantitativa de distorção mais utilizada é o tradicional erro quadrático ou energia do erro [14]. Contudo, para sistemas de compressão de voz a baixas taxas de bits, a energia do erro não atende ao segundo requisito, ou seja, não é consistente com os resultados subjetivos. Para esta aplicação, surgiram diversas outras medidas alternativas [14], [83]. Estas medidas possuem as seguintes propriedades. Dado que  $d(X,Y)$  corresponde à distorção entre o vetor  $X$  e o vetor  $Y$ ,  $d(X,Y)$  não pode ser negativa e  $d(X,Y)=0$  se  $X=Y$  [14], [5]. Estas duas propriedades são gerais para qualquer medida de distorção sob o ponto de vista da teoria da distorção pela taxa [5].

Uma classe particular das medidas de distorção é formada pelas medidas baseadas em distâncias. Estas medidas, também chamadas de métricas de Minkowski [10], correspondem a normas  $L_p$  do vetor  $E$  obtido a partir da diferença entre  $X$  e  $Y$  [153]. O vetor  $E$  corresponde ao erro entre  $X$  e  $Y$ .

No espaço euclidiano  $K$ -dimensional, as normas  $L_p$  de um vetor  $E$  são dadas por:

$$\|E\|_p = \left( \sum_{i=1}^K |e(i)|^p \right)^{1/p} \tag{3.1}$$

onde  $e(i)$  é a  $i$ -ésima amostra do vetor  $E$ , e  $K$  é a dimensão deste vetor.

O vetor  $E$  corresponde a um determinado ponto no espaço  $K$ -dimensional. A Fig. 3.4, mostra o lugar geométrico dos pontos equidistantes de um determinado ponto no espaço tridimensional para diversos valores de  $p$ .

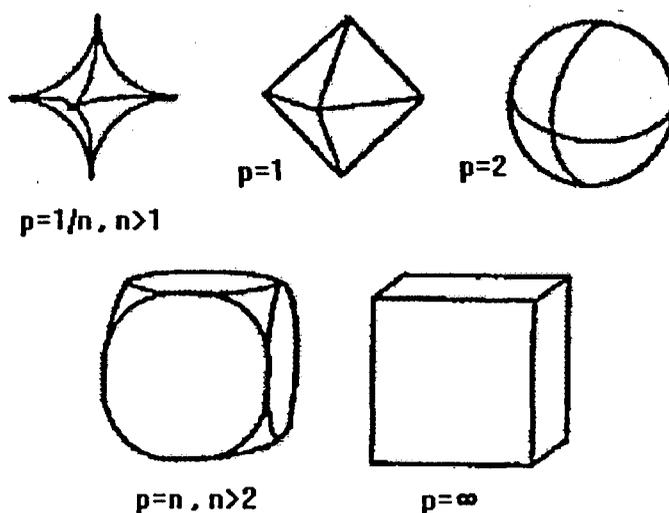


Figura 3.4- Conceito de equidistância para diferentes normas  $L_p$  [10].

Normalmente, utilizamos as medidas de distorção de forma comparativa. A etapa de busca da melhor "codeword" em QV é um exemplo. Assim, não estamos interessados no valor absoluto da distorção e podemos desprezar a potência  $1/p$  da norma  $L_p$ . Devemos ressaltar que, obviamente, este artifício não é válido para  $p=\infty$ . Assim, as medidas de distorção baseadas em distâncias são dadas por:

$$d_p(X, Y) = (\|E\|_p)^p = \sum_{i=1}^K |x(i) - y(i)|^p \quad (3.2)$$

Podemos observar da equação (3.2) que o erro quadrático corresponde a  $d_2(X, Y)$ .

Em certas aplicações é conveniente ponderar de forma diferente cada amostra do vetor erro  $E$ . Nestes casos, as medidas de distorção utilizadas são chamadas de medidas ponderadas de distorção. A ponderação pode ser tratada matematicamente através da notação matricial definindo-se a matriz de ponderação  $W$ . A distorção ponderada é dada por:

$$d_w(X, Y) = (X - Y)^T \cdot W \cdot (X - Y) \quad (3.3)$$

O erro quadrático é obtido a partir da equação (3.3) fazendo-se  $W = I$ , a matriz identidade.

Buscando obter maior correlação com a percepção humana, foram propostas várias medidas ponderadas de distorção que possuem caráter dinâmico, ou seja, a matriz  $W$  não é fixa. Nestas medidas, a matriz  $W$  depende do vetor  $X$  a ser quantizado. Estas medidas de ponderação são dadas por [209]:

$$d_w(X, Y) = (X - Y)^T \cdot W(X) \cdot (X - Y) \quad (3.4)$$

onde a notação  $W(X)$  indica que a matriz de ponderação é função do vetor a ser quantizado.

Esta classe de medidas é chamada em [209] de "input weighted squared error per sample". Neste trabalho serão denominadas de medidas dinâmicas de distorção. Os sistemas de QV das LSFs normalmente utilizam medidas dinâmicas.

Em QV, a medida de distorção é utilizada tanto na fase do projeto do "codebook" quanto na etapa de busca da melhor "codeword". Já citamos anteriormente que os algoritmos de projeto de "codebook" exigem o cálculo da expressão da centróide. Relembramos que a centróide é o vetor que minimiza a distorção média total para um dado conjunto de vetores. No caso do erro quadrático, o vetor centróide  $C$  equivale ao centro euclidiano de gravidade ou baricentro dos  $P$  vetores em questão, sendo dada por [153]:

$$C = \frac{1}{P} \sum_{i=1}^P X_i \quad (3.5)$$

onde  $X_i$  representa o  $i$ -ésimo vetor deste conjunto.

Esta expressão é válida também para as medidas ponderadas de distorção, desde que não sejam dinâmicas. As medidas dinâmicas exigem uma outra expressão para a centróide: para cada um dos  $P$  vetores do grupo, teremos uma matriz  $W(X_i)$  correspondente. Definindo a matriz  $M$  como sendo:

$$M = \sum_{i=1}^P W(X_i) \quad (3.6)$$

teremos a seguinte expressão para a centróide relativa à distorção dinâmica [209]:

$$C = M^{-1} \cdot \sum_{i=1}^P \{W(X_i) \cdot X_i\} \quad (3.7)$$

onde  $M^{-1}$  corresponde à matriz inversa de  $M$ .

Já apresentamos no Capítulo 2 a determinação da distorção espectral (equação (2.17)). Esta medida de distorção é apresentada de forma gráfica através da Fig. 3.5. A distorção espectral (SD) é atualmente a medida de distorção mais utilizada para avaliação da quantização do filtro LPC [146]. Contudo, a distorção espectral não pode ser usada no projeto dos "codebooks", pois não existe ligação direta entre a mesma e os parâmetros que representam o filtro LPC. A SD pode ser utilizada na etapa de codificação, de forma que a "codeword" escolhida seja aquela que conduzir à menor SD. Todavia, a complexidade computacional desta busca seria muito grande. Uma alternativa intermediária é a utilização da SD apenas na decisão entre um pequeno número de "codewords" pré-selecionadas [183]. Por causa da sua complexidade matemática, a distorção espectral geralmente é utilizada apenas na avaliação "off-line" do desempenho do quantizador.

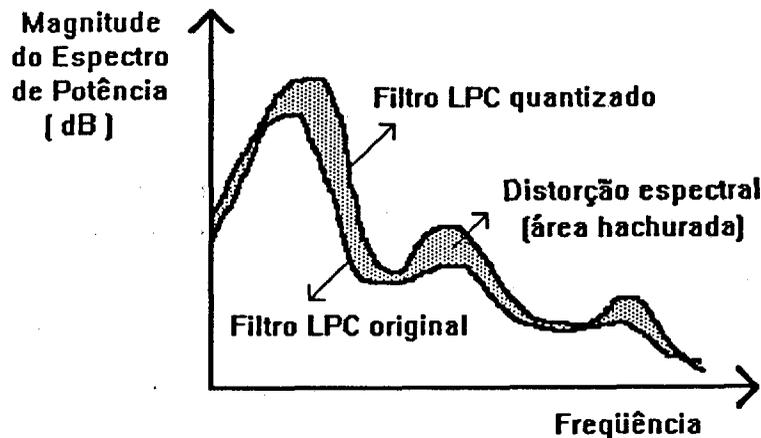


Fig. 3.5 - Ilustração gráfica da distorção espectral (SD) [53].

### 3.7- UTILIZAÇÃO DE "LINE SPECTRAL FREQUENCIES"

A análise LPC fornece o polinômio  $A(z)$  não-quantizado. O filtro LPC corresponde à função  $H(z)=1/A(z)$  (equação (2.1)). A precisão de  $A(z)$  é limitada apenas pelo tamanho da palavra do sistema de processamento. Atualmente, dispomos de DSPs ("Digital Signal Processors") que operam com palavras de 32 bits. Implementações de linguagens de alto nível, em estações de trabalho ou em

computadores pessoais, suportam palavras de 64 bits. Neste caso, um só filtro LPC com 10 coeficientes corresponderia a 640 bits.

Todavia, em sistemas de telecomunicações, reconhecimento de voz e outras aplicações, desejamos minimizar o número de bits necessários para representar  $A(z)$ . Nestas aplicações,  $A(z)$  deve ser representado por um polinômio  $\hat{A}(z)$ . O polinômio  $\hat{A}(z)$  é codificado a partir de um reduzido número de bits e deve desempenhar adequadamente as funções atribuídas a  $A(z)$ .

Já abordamos a quantização escalar na seção 3.2 deste capítulo. A maneira mais óbvia de calcularmos  $\hat{A}(z)$  é quantizarmos escalarmente cada coeficiente de  $A(z)$  com  $b_i$  bits ( $\hat{a}_i = Q_E^{b_i}\{a_i\}$ ). Contudo, os resultados de anos de pesquisa mostraram que os coeficientes  $a_i$  não apresentam características adequadas para quantização. Listamos abaixo algumas destas características, juntamente com uma breve explicação das mesmas:

*(a) - Estabilidade do filtro:*

É muito importante que o filtro LPC quantizado seja estável. Caso contrário, o sinal de voz sintetizado apresentará saturação [176, pp.61] e degradação audível. A operação de quantização pode gerar um filtro instável a partir de um filtro estável. Desta forma, devemos dispor de um método para testar a estabilidade do filtro quantizado. No caso da quantização dos próprios coeficientes de  $A(z)$ , devemos testar se as raízes do filtro quantizado encontram-se no interior do círculo unitário no plano  $z$  [38]. A substituição dos coeficientes por parâmetros alternativos facilita os testes de estabilidade por evitar a necessidade de cálculo das raízes de  $\hat{A}(z)$ .

A questão da estabilidade sempre é citada como uma desvantagem da quantização dos próprios coeficientes do filtro. Devemos mencionar que quando utilizamos a QV com "codebooks" não-estruturados, este problema não existe. Neste caso, basta garantirmos que o "codebook" seja composto apenas por "codewords" que correspondam a filtros estáveis. A própria operação de QV encarregar-se-ia de evitar filtros instáveis. Todavia, o problema do teste da estabilidade do filtro quantizado volta a ganhar importância quando são utilizados quantizadores como o multi-estágio e o "split".

*(b) - Faixa dinâmica:*

A faixa dinâmica consiste na faixa de valores assumidos por determinado parâmetro. Quando buscamos eficiência na quantização, devemos optar por parâmetros que apresentem reduzida faixa dinâmica e pequena variância. Uma faixa dinâmica reduzida levará a "codebooks" menores com a mesma eficiência de codificação.

(c) - *Sensibilidade espectral:*

A sensibilidade espectral é uma medida de quanto o espectro de potência do filtro LPC se modifica a partir da alteração de um determinado parâmetro. É importante que a sensibilidade espectral em relação a este parâmetro seja predizível. Desde que isto ocorra, podemos otimizar o quantizador de forma a considerar a importância individual dos parâmetros [37], [69], [122]. Esta medida de importância é dada pela sensibilidade espectral relativa a cada parâmetro.

(d) - *Boas características de interpolação:*

Nos codificadores paramétricos e híbridos, um novo filtro LPC é transmitido a um dado intervalo de tempo pré-definido (duração do quadro). Quanto maior o intervalo de atualização, menor será a taxa de bits correspondente ao filtro LPC. Uma das alternativas para aumentarmos este intervalo de atualização é utilizarmos a interpolação [53]. Na maior parte dos casos utiliza-se uma interpolação linear. O parâmetro a ser interpolado deve apresentar pequena variação no decorrer do tempo, permitindo que os valores obtidos a partir da interpolação conduzam a uma boa aproximação dos valores corretos. Neste caso, estamos considerando que os valores corretos são aqueles obtidos a partir de um pequeno intervalo de atualização, e que correspondam às transições no espectro do sinal de voz ocorridas no decorrer do tempo.

(e) - *Suporte a uma medida de distorção eficaz:*

Quando utilizamos QV, necessitamos da definição de uma medida eficaz de distorção. Esta medida deve ser significativa subjetivamente. Além disso, deve apresentar reduzida complexidade computacional. A medida de distorção mais utilizada, devido principalmente à sua baixa complexidade, é o erro quadrático (EQ). No caso da QV dos coeficientes LPC, o EQ não apresenta bom desempenho. Neste caso, a distorção de Itakura-Saito conduz a um desempenho melhor que o EQ [153].

Devemos ressaltar que a estrutura do quantizador impõe restrições à formulação de medidas de distorção. Já abordamos o motivo pelo qual a QV do filtro LPC no CELP exige a utilização de "codebooks" estruturados. Assim, é importante que a implementação da medida de distorção seja compatível com a estrutura do quantizador.

Como os coeficientes  $a_i$ , em geral, não são eficientes em relação às características citadas acima, surgiram diversas formas alternativas de representação do filtro LPC [11]. Uma vez que a representação de um dado polinômio  $A(z)$  deve ser única, a determinação destes parâmetros alternativos devem ser funções biunívocas dos coeficientes  $a_i$ . Dentre estes parâmetros, os PARCORs ("Partial Correlation Coefficients") [11], ou coeficientes de reflexão, foram os mais utilizados durante muito tempo.

Os PARCORs são um subproduto do algoritmo de Durbin para análise LPC [11]. Sendo assim, sua determinação não representa complexidade adicional. Para garantirmos a estabilidade do filtro, basta assegurarmos que os PARCORs encontram-se na faixa de -1 a 1. A sensibilidade espectral do filtro LPC em relação ao valor do PARCOR, aumenta muito quando o módulo deste PARCOR aproxima-se da unidade [6]. Isto exigiria que a quantização do PARCOR fosse realizada de maneira não uniforme, com maior número de bits alocados para os valores próximos de -1 e 1. Entretanto, para contornar a quantização não uniforme (maior complexidade - seção 3.2), são utilizadas transformações não-lineares. Estas transformações visam tornar mais uniforme a sensibilidade espectral em relação a cada PARCOR  $k_i$ . As duas transformações comumente utilizadas são o seno inverso e as razões log-área [6]. A primeira corresponde ao cálculo do seno inverso hiperbólico de cada PARCOR:  $S_i = \sinh^{-1}(k_i)$ . A razão log-área é dada por:  $K_i = \ln[(1+k_i)/(1-k_i)]$ . Estas duas transformações conduzem a bons resultados e não existe supremacia de uma em relação à outra [176]. Após a transformação, os coeficientes  $K_i$  e  $S_i$  podem ser quantizados uniformemente.

Atualmente a representação do filtro LPC através dos PARCORs tem sido substituída pela representação através das LSFs ("Line Spectral Frequencies"). As LSFs foram propostas por Itakura em 1975 [8]. A popularização das LSFs não foi imediata. Em [92], Atal et al concluem que as LSFs apresentam desempenho inferior ao seno inverso dos PARCORs ( $S_i$ ). Contudo, os japoneses continuaram as pesquisas e conseguiram difundir as LSFs [53], [146], [190]. O próprio Atal em [146] e [194] utiliza as LSFs na QV dos parâmetros do filtro LPC.

As LSFs apresentam diversas características desejáveis, tais como reduzida faixa dinâmica, conservação da estabilidade do filtro quantizado e interpolação eficiente [53]. Além disso, suas sensibilidades espectrais são localizadas - uma mudança em determinada LSF altera o espectro de potência do filtro LPC apenas em frequências próximas a esta LSF [146]. Para a QV, as LSFs possibilitam o uso de medidas de distorção relativamente simples e subjetivamente significativas [173], [194]. A desvantagem da utilização das LSFs é a complexidade computacional para sua obtenção. A determinação das LSFs requer o cálculo das raízes de dois polinômios de ordem igual à metade da ordem do filtro LPC (para ordens pares). A seguir, descrevemos o processo de obtenção das LSFs.

A análise LPC fornece o polinômio  $A(z)$  dado por:

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_p z^{-p} \quad (3.8)$$

onde  $P$  é a ordem do filtro LPC (filtro PCA).

Para calcularmos as LSFs, formamos dois polinômios auxiliares:

$$Q_{P+1}(z) = A(z) + z^{-(P+1)} A(z^{-1}) \quad (3.9)$$

$$P_{P+1}(z) = A(z) - z^{-(P+1)} A(z^{-1}) \quad (3.10)$$

onde  $Q_{P+1}(z)$  tem simetria par e  $P_{P+1}(z)$  tem simetria ímpar. Esta operação equivale à decomposição da resposta ao impulso correspondente ao polinômio  $A(z)$  em suas partes par e ímpar, com o eixo de simetria deslocado para  $(P+1)/2$  [37].

Para  $P$  par,  $Q_{P+1}(z)$  tem uma raiz em  $z=-1$  e  $P_{P+1}(z)$  tem uma raiz em  $z=1$ . Para  $P$  ímpar,  $Q_{P+1}(z)$  não tem raiz em  $z=-1$  e  $P_{P+1}(z)$  tem raiz de multiplicidade 2 em  $z=1$ .

O próximo passo para a determinação das LSFs é a extração das raízes em  $z=\pm 1$ , gerando dois novos polinômios  $P(z)$  e  $Q(z)$ . Quando  $P$  é par, a ordem de  $P(z)$  e  $Q(z)$  é  $P$ . Quando  $P$  é ímpar, a ordem de  $P(z)$  é  $P-1$  e a ordem de  $Q(z)$  é  $P+1$ . Assim,  $P(z)$  e  $Q(z)$  são sempre de ordem par.

As LSFs são as raízes dos polinômios  $Q(z)$  e  $P(z)$ . As LSFs ocorrem em pares complexos conjugados e pertencem ao círculo unitário do plano  $z$  [93]. Por ocorrerem aos pares, as LSFs são conhecidas também como LSPs ("Line Spectral Pairs"). Assim, basta calcularmos as LSFs correspondentes a um dos semi-círculos. Além disto, devemos calcular apenas suas posições angulares, que variam de 0 a  $\pi$  radianos, uma vez que as LSFs têm sempre módulo unitário [30].

Devemos ressaltar que a representação através das LSFs parte do pressuposto que o filtro LPC é estável, ou seja, que  $A(z)$  é de fase mínima [38]. Neste caso, as propriedades das raízes de  $Q(z)$  e  $P(z)$  são demonstradas em [30]. As três mais importantes são:

i) As LSFs estão sempre sobre o círculo unitário.

ii) As LSFs possuem um padrão de ordenação onde a frequência da seguinte nunca é maior do que a da anterior. As LSFs correspondentes a  $Q(z)$  são alternadas com as correspondentes a  $P(z)$  no círculo unitário. Chamando de  $w_{p_i}$  e  $w_{q_i}$ , os ângulos das  $i$ -ésimas raízes de  $P(z)$  e  $Q(z)$  respectivamente, teremos:  $w_{p_1} < w_{q_1} < w_{p_2} < w_{q_2} < \dots < w_{p_P}$ .

iii) Mantendo-se a ordenação na quantização das LSFs, o filtro quantizado resultante é estável.

A Fig. 3.6 apresenta o círculo unitário do plano  $z$ . São mostrados os pólos do filtro LPC e as LSFs respectivas. Podemos notar que para pólo próximo do círculo unitário, geralmente existe um par de LSFs correspondente.

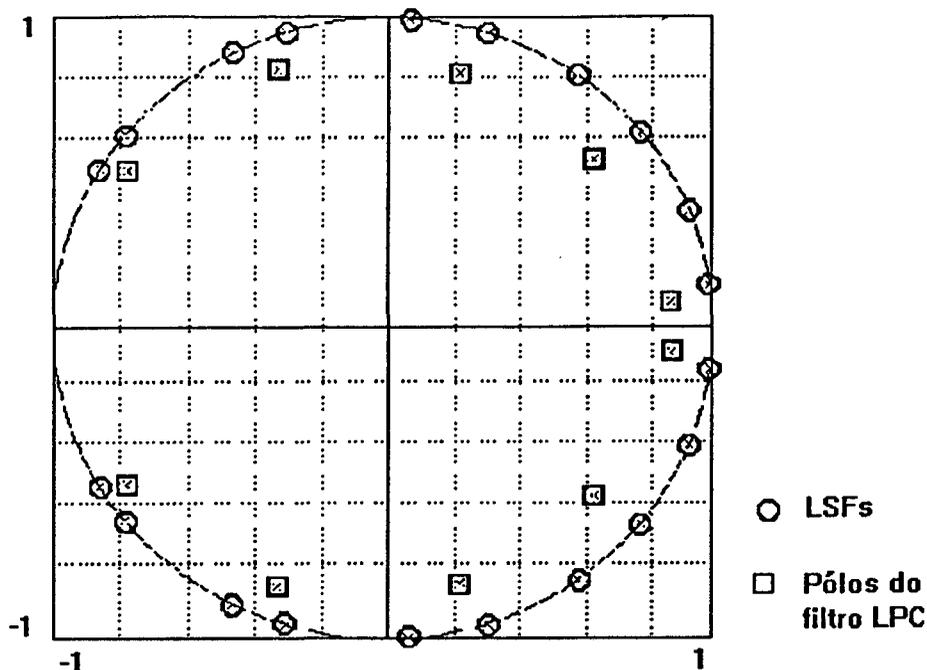


Fig. 3.6 - Distribuição das LSFs no plano z para um filtro LPC de ordem 10

A partir do conhecimento das propriedades das LSFs, podemos calculá-las de forma eficiente. No caso de um filtro LPC de ordem  $P=8$ , a partir do procedimento descrito anteriormente, determinamos  $P(z)$  e  $Q(z)$ :

$$P(z) = 1 + p_1 \cdot z^{-1} + p_2 \cdot z^{-2} + p_3 \cdot z^{-3} + p_4 \cdot z^{-4} + p_5 \cdot z^{-5} + p_6 \cdot z^{-6} + p_7 \cdot z^{-7} + p_8 \cdot z^{-8} \quad (3.11)$$

$$Q(z) = 1 + q_1 \cdot z^{-1} + q_2 \cdot z^{-2} + q_3 \cdot z^{-3} + q_4 \cdot z^{-4} + q_5 \cdot z^{-5} + q_6 \cdot z^{-6} + q_7 \cdot z^{-7} + q_8 \cdot z^{-8} \quad (3.12)$$

Podemos substituir  $z$  por  $e^{j\omega}$ , pois sabemos que as LSFs encontram-se no círculo unitário. A simetria dos coeficientes de  $P(e^{j\omega})$  e  $Q(e^{j\omega})$  permite que arranjemos os polinômios em cossenos. Para  $P=8$ , temos a seguinte expressão para  $P(e^{j\omega})$ :

$$P(e^{j\omega}) = e^{-j4\omega} [16 \cos^4(\omega) + 8p_1 \cdot \cos^3(\omega) + 4(p_2 - 4) \cos^2(\omega) + 2(p_3 - 3p_1) \cos(\omega) + p_4 - 2p_2 + 2] \quad (3.13)$$

Representando  $\cos(\omega) = x$  e desprezando o termo correspondente à fase linear, temos para o caso da ordem do filtro LPC ser  $P=8$ :

$$P(x) = 16x^4 + 8p_1 \cdot x^3 + 4(p_2 - 4)x^2 + 2(p_3 - 3p_1)x + p_4 - 2p_2 + 2 \quad (3.14)$$

$$Q(x) = 16x^4 + 8q_1 \cdot x^3 + 4(q_2 - 4)x^2 + 2(q_3 - 3q_1)x + q_4 - 2q_2 + 2 \quad (3.15)$$

As ordens dos polinômios  $P(x)$  e  $Q(x)$  correspondem à metade da ordem dos polinômios  $P(z)$  e  $Q(z)$ , respectivamente. Para um filtro LPC de ordem  $P=10$ , temos:

$$P(x) = 32x^5 + 16p_1 \cdot x^4 + 8(p_2 - 5)x^3 + 4(p_3 - 4p_1)x^2 + 2(p_4 - 3p_2 + 5)x + p_5 + 2p_1 - 2p_3 \quad (3.16)$$

$$Q(x) = 32x^5 + 16q_1 \cdot x^4 + 8(q_2 - 5)x^3 + 4(q_3 - 4q_1)x^2 + 2(q_4 - 3q_2 + 5)x + q_5 + 2q_1 - 2q_3 \quad (3.17)$$

Devemos calcular as raízes de  $P(x)$  e  $Q(x)$ . As LSFs são então determinadas através do cosseno inverso destas raízes. Existem métodos analíticos para o cálculo das raízes de polinômios de ordem inferior a 5. Isto indica que só precisamos recorrer a métodos numéricos quando a ordem do filtro LPC for maior do que 8 [30]. Neste caso, qualquer método numérico de cálculo de raízes poderia ser utilizado. Todavia, algoritmos específicos exploram melhor as peculiaridades das LSFs e são mais eficientes para implementações em tempo real [30], [51]. Neste trabalho, utilizamos o algoritmo proposto em [51], baseado em polinômios de Chebyshev [208].

Na etapa de descodificação, devemos calcular os coeficientes do filtro LPC a partir das LSFs. Esta operação é mais simples do que a transformação inversa. Devemos apenas remontar o polinômio  $\hat{A}(z)$  a partir das LSFs quantizadas. A forma mais trivial de fazer isto seria remontar  $P(z)$  e  $Q(z)$  a partir de suas respectivas LSFs, incorporar as raízes em -1 e 1 de forma a obter  $P_{p+1}(z)$  e  $Q_{p+1}(z)$ , e finalmente, fazer:  $\hat{A}(z) = (P_{p+1}(z) + Q_{p+1}(z))/2$ . Contudo, preferimos adotar neste trabalho o algoritmo descrito em [51] por ser mais eficiente computacionalmente.

### 3.8- CONCLUSÕES

Neste capítulo apresentamos os fundamentos da QV como uma extensão da quantização escalar não-uniforme. Buscamos detalhar alguns procedimentos de quantização que são necessários na etapa de implementação do quantizador.

Fizemos também uma pequena revisão dos métodos de projeto de "codebooks". O algoritmo LBG é indiscutivelmente o mais utilizado e por sua importância o descrevemos detalhadamente. Por outro lado, como muitas coisas em engenharia, relações de compromisso podem fazer com que outros métodos se mostrem mais adequados em determinadas situações. O algoritmo PNN, por exemplo, é uma boa alternativa quando se necessita de menor complexidade computacional no projeto. Enfocamos também a utilização de redes neurais para o projeto de "codebooks". Acreditamos que ainda existe muita coisa a ser pesquisada em relação à aplicação de redes neurais no projeto de "codebooks", principalmente quando se exige adaptabilidade.

Dedicamos atenção especial às medidas de distorção, organizando algumas informações dispersas na literatura acerca de suas propriedades matemáticas e geométricas. Buscamos definir uma nomenclatura que será seguida no decorrer deste trabalho. As medidas dinâmicas de distorção são bastante utilizadas na QV das LSFs e serão estudadas detalhadamente no Capítulo 5.

Apresentamos o conceito de LSFs que atualmente encontra-se bastante difundido. Expusemos apenas as idéias básicas acerca destes parâmetros. Existem diversos outros artigos citados no texto que também enfocam as LSFs (ex: [208]) e podem complementar nossas informações.

Nosso objetivo neste capítulo foi apresentar os subsídios teóricos básicos da QV das LSFs para que o leitor que não tenha experiência neste assunto possa acompanhar adequadamente o

desenvolvimento deste trabalho. A QV por si só é um assunto bastante vasto e não pretendemos esgotá-la. Tentamos apenas ressaltar os conceitos inseridos no escopo deste trabalho, procurando apresentá-los de maneira didática tendo em mente o leitor que não tem experiência em QV, uma vez que existem inúmeras referências para aqueles que desejam se aprofundar neste assunto.

# CAPÍTULO 4

## BUSCA RÁPIDA EM ÁRVORE K-DIMENSIONAL

### 4.1- INTRODUÇÃO

Um quantizador vetorial pode ser visto como a cascata de um codificador e um decodificador. O codificador identifica em qual região do espaço o vetor a ser quantizado se encontra, associando-lhe a "codeword" correspondente à região identificada. O decodificador gera o vetor de saída a partir do índice desta "codeword". Essencialmente, o codificador é definido pelo mapeamento do espaço K-dimensional em um número finito de regiões. Neste capítulo aprofundaremos o estudo das propriedades destas regiões. Podemos utilizar o conhecimento acerca da geometria destas regiões na elaboração de métodos de busca rápida que permitam diminuir a complexidade da QV.

Este capítulo encontra-se organizado da seguinte forma. Na seção 4.2 apresentamos o conceito de partições de Voronoi. Na seção 4.3 abordamos os métodos de busca rápida, dando ênfase à estrutura conhecida como árvore K-dimensional (K-d). Na seção 4.4 enfocamos a implementação e o projeto da árvore K-dimensional através do algoritmo GOC. Na seção 4.5 teremos as conclusões deste capítulo.

### 4.2- PARTIÇÕES DE VORONOI

Consideremos um conjunto ("codebook")  $Z = \{\hat{X}_1, \hat{X}_2, \dots, \hat{X}_N\}$  de N vetores ("codewords") de dimensão K. As "codewords"  $\hat{X}_1, \hat{X}_2, \dots, \hat{X}_N$  correspondem a pontos no espaço K-dimensional. Consideremos também o erro quadrático  $d_2(X, Y)$  como medida de distância (distorção) entre dois vetores X e Y. As "codewords"  $\hat{X}_i$  são comumente chamadas de vizinhos. Na técnica de QV, a etapa de busca da melhor "codeword" é também mencionada na literatura de busca do vizinho mais próximo ("nearest neighbor search"). O conjunto de pontos (vetores) P que estão mais próximos à "codeword"  $\hat{X}_i$  do que a qualquer "codeword"  $\hat{X}_j \in Z$ , com  $j \neq i$ , determina uma região no espaço chamada de partição de Voronoi [203] da "codeword"  $\hat{X}_i$ . A partição de Voronoi de  $\hat{X}_i$  é definida como:

$$V(\hat{X}_i) = \{X \in \mathfrak{R}^K : d_2(X, \hat{X}_i) < d_2(X, \hat{X}_j) \text{ para todo } j \neq i\} \quad (4.1)$$

A união das partições de Voronoi de todas as "codewords" é chamada de diagrama de Voronoi. Os quantizadores que possuem a propriedade de dividir o espaço em partições de Voronoi foram

chamados por Gersho de quantizadores de Voronoi [23]. A Fig. 4.1 mostra um diagrama de Voronoi para um "codebook" de 12 "codewords" ( $N=12$ ) de dimensão  $K=2$ .

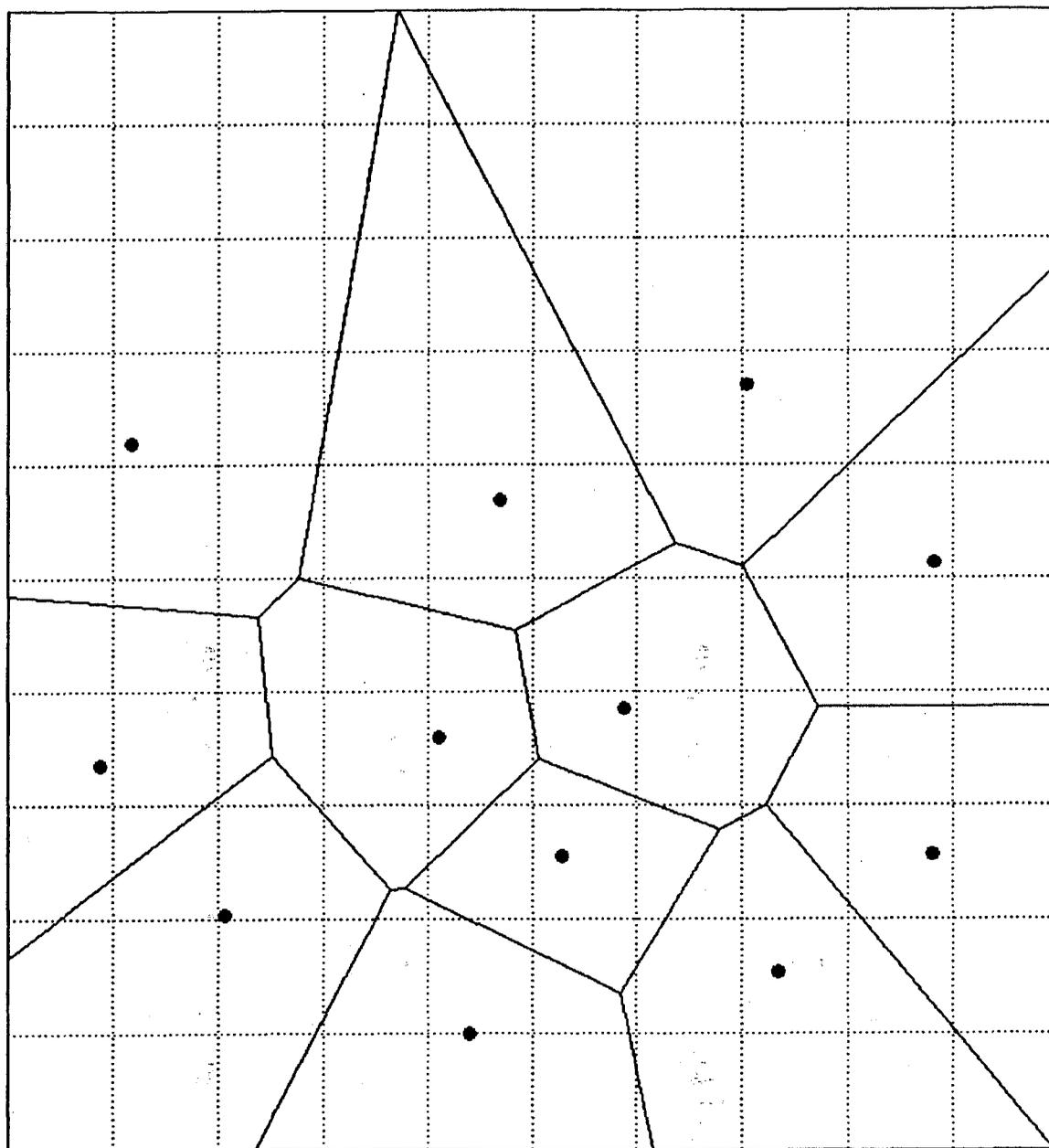


Fig. 4.1- Exemplo de diagrama de Voronoi.

A determinação das partições de Voronoi no plano ( $K=2$ ) é uma tarefa relativamente simples de ser executada "off-line" [17], [18]. Todavia, quando  $K$  assume valores grandes (acima de 8 aproximadamente), esta tarefa passa a ter complexidade demasiadamente grande. Os algoritmos para o cálculo das partições de Voronoi no espaço de dimensão  $K$  é objeto de grande interesse em geometria computacional [17], [18].

Historicamente o conceito de diagrama de Voronoi encontra-se associado à utilização do erro quadrático como medida de distorção [17], [18], [23]. Todavia este conceito pode ser estendido a outras medidas de distorção [54], [203]. Desta forma, podemos considerar que uma vez projetado o "codebook", o conjunto de N "codewords" divide todo o espaço K-dimensional em N regiões a partir da medida de distorção utilizada [171]. Estas regiões constituem-se nas partições de Voronoi generalizadas.

O motivo de estudarmos os diagramas de Voronoi é a idéia de utilizarmos este conhecimento na formulação de um algoritmo eficiente para a busca da melhor "codeword" [23]. Ao invés de reduzirmos a complexidade da etapa de codificação em QV impondo estruturas restritivas ao quantizador, podemos desenvolver algoritmos rápidos de busca que funcionem para um "codebook" arbitrário. Estes algoritmos geralmente baseiam-se na pré-computação de uma estrutura de dados que possibilite aumentar a eficiência da busca da melhor "codeword" [23], [52]. O algoritmo de busca pode também ser projetado de forma a não incorrer em aumento da distorção.

O objeto da próxima seção é a utilização de uma estrutura eficiente para a etapa de busca, baseada no conceito de árvore K-dimensional.

#### 4.3- MÉTODOS DE BUSCA RÁPIDA

A pesquisa de um método eficiente para a determinação do vizinho mais próximo, conduziu-nos a diversos métodos, obtidos a partir de extensa revisão bibliográfica [9], [10], [19], [40], [52], [66], [67], [81], [96], [98], [162], [163], [171], [186], [187], [204], [205]. Nesta seção, apresentaremos os conceitos nos quais baseiam-se os principais métodos de busca rápida, dando ênfase à busca rápida baseada em árvore K-dimensional.

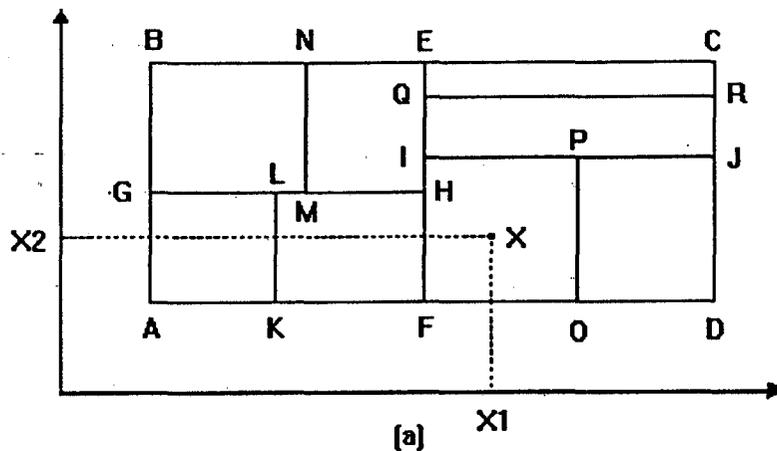
##### 4.3.1- ÁRVORE K-DIMENSIONAL

A árvore K-dimensional (árvore K-d) corresponde a uma estrutura de busca em árvore binária multidimensional. A árvore K-d é uma generalização da simples árvore binária unidimensional. A árvore K-d foi proposta por Bentley [7] como uma estrutura de dados para armazenamento de informações que devam ser recuperadas através de buscas associativas. Estas buscas são bastante utilizadas em processamento de dados, onde as informações devem ser encontradas de forma eficiente. Por exemplo, a partir de um banco de dados poderíamos querer determinar quem são as pessoas com idade superior a 30 anos e cujos nomes principiêm pela letra "A". A árvore K-d é uma estrutura de dados que facilita este tipo de pesquisa [7]. A árvore K-d vem sendo utilizada nas duas etapas mais importantes da QV: no projeto do "codebook" [97] e na codificação ou busca da melhor "codeword" [171].

Cada nó da árvore K-d divide o espaço  $\mathfrak{R}^K$  em duas regiões através de um hiperplano ortogonal a um dos K eixos cartesianos. Este hiperplano H é representado por  $H = \{x \in \mathfrak{R}^K / x_j = h\}$  e define duas regiões,  $\mathfrak{R}_L$  e  $\mathfrak{R}_R$  [171]. Para representar H necessita-se de apenas dois valores: j e h. O valor de j determina qual será o eixo ortogonal ao hiperplano H. O valor de h corresponde à interseção de H com o eixo j. Qualquer ponto (vetor X) no espaço  $\mathfrak{R}^K$  pode ser localizado em relação a H com uma simples comparação escalar,  $X(j) \leq h$ , onde  $X(j)$  é a j-ésima componente de X. A região inicial é todo o espaço  $\mathfrak{R}^K$  e corresponde à raiz da árvore na camada 1. As duas subregiões  $\mathfrak{R}_L$  e  $\mathfrak{R}_R$  obtidas pela divisão proporcionada por H, correspondem aos dois nós filhos na camada 2. Cada uma destas duas regiões é sucessivamente subdividida por hiperplanos ortogonais. Depois de d divisões sucessivas a partir da camada 1, temos uma árvore de profundidade d, com  $2^d$  regiões terminais chamadas folhas ("bucket" ou "leaf" na literatura de língua inglesa) na camada d+1.

Cada nó não-terminal é associado com uma região e um hiperplano divisor. A definição de cada nó requer apenas o armazenamento de dois escalares. Dado qualquer vetor X em  $\mathfrak{R}^K$ , uma seqüência de d comparações escalares entre os valores de h dos pares (j,h) que definem cada nó e as correspondentes j-ésimas componentes de X conduz a uma folha contendo o vetor. Resumindo: a árvore K-d de profundidade d secciona o espaço  $\mathfrak{R}^K$  em  $2^d$  regiões retangulares disjuntas (folhas) e permite identificar a folha contendo um dado vetor X em apenas d comparações escalares.

A estrutura básica da árvore K-d é mostrada na Fig. 4.2 para o caso bi-dimensional. A região raiz ABCD é dividida pelo hiperplano EF em duas metades. A Fig. 4.2(b) mostra a correspondente árvore de profundidade 3, gerada por divisões destas duas regiões por vários hiperplanos ortogonais a um dos eixos coordenados. Um vetor  $X = \{X_1, X_2\}$  é localizado dentro da região folha IFOP depois de 3 comparações escalares:  $X_1$  com EF,  $X_2$  com IJ e  $X_1$  com OP. Similarmente, o percurso correspondente a um vetor localizado na região folha BGMN irá consistir de uma seqüência de comparações com os hiperplanos EF, GH e MN.



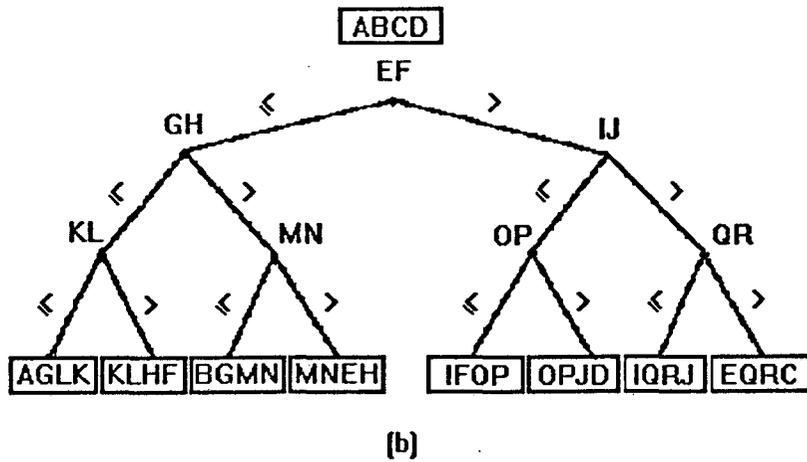


Fig. 4.2- Exemplo de árvore K-dimensional (adaptada de [171])

Uma vez definido o hiperplano divisor para uma dada região, basta localizarmos de que lado deste hiperplano localiza-se o vetor a ser quantizado. A Fig. 4.3 ilustra o conceito geométrico envolvido. Restringe-se a busca apenas às regiões correspondentes às "codewords" que situam-se deste mesmo lado do hiperplano. Este processo é repetido até que se atinja uma folha da árvore. Daí, realizamos uma busca exhaustiva entre as "codewords" respectivas a esta folha. Chamaremos esta estratégia de determinação da melhor "codeword", de busca simples em árvore K-d.

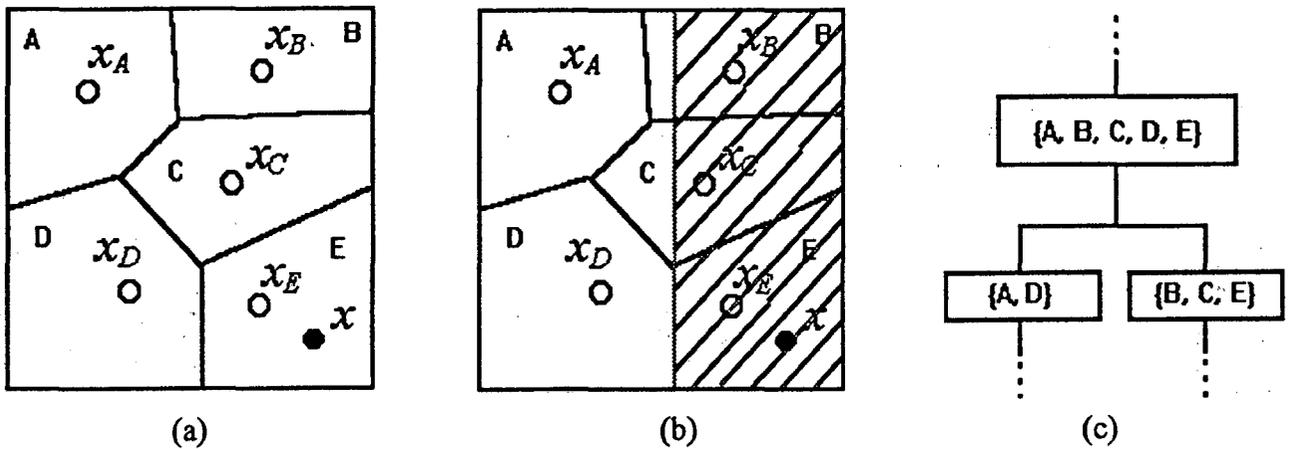


Fig.4.3 (a) Partições de Voronoi    (b) Separação em duas regiões    (c) Estrutura da árvore de busca simples

No processo de busca simples, estamos sujeitos a não identificar corretamente qual a melhor "codeword" e haver, conseqüentemente, um aumento da distorção média. O problema é que a melhor "codeword" pode pertencer a uma região que é atravessada por um hiperplano. De acordo com a estratégia de busca definida, a partir do momento em que a região atravessada por um hiperplano é associada a um dos lados deste hiperplano, todos os pontos pertencentes a esta região que se situem no lado oposto do hiperplano conduzirão à identificação errônea de seus respectivos vizinhos mais

próximos. Para evitarmos este tipo de erro, podemos adotar outra estratégia de busca que procure obter o mesmo resultado da busca exaustiva.

Nesta segunda estratégia, de forma similar à primeira, inicialmente verificamos de que lado do hiperplano separador encontra-se  $x$ . Em seguida, buscamos seu vizinho mais próximo desprezando as "codewords" respectivas às partições que tenham interseção nula com a região à qual pertence  $x$ . A Fig.4.4(b) ilustra este procedimento. Ao determinarmos que o vetor  $x$  está na região hachurada, podemos descartar da busca as partições A e D. As regiões correspondentes aos dois ramos da árvore inferiores a este nó serão então formadas pelos conjuntos de partições  $\{A,B,C,D\}$  e  $\{B,C,E\}$ , respectivamente, conforme mostrado na Fig. 4.4(c). Esta técnica evita soluções sub-ótimas porque descarta apenas as "codewords" que certamente não podem ser o vizinho mais próximo do vetor  $x$  a ser quantizado. Por este motivo a chamamos de busca conservativa. Em [171], este tipo de estratégia de busca é chamada "bucket-Voronoi intersections framework".

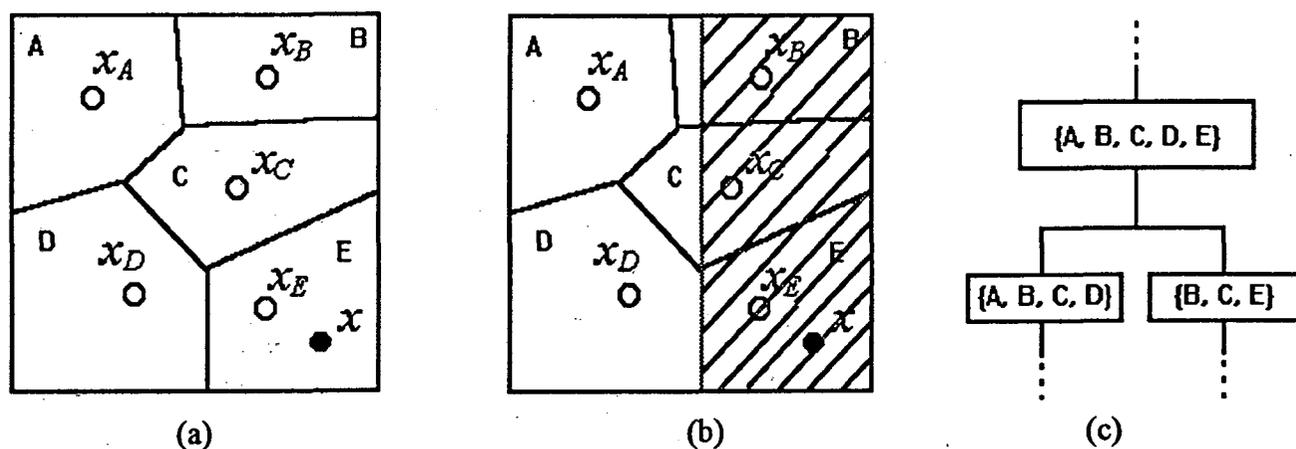


Fig.4.4 (a) Partições de Voronoi

(b) Separação em duas regiões

(c) Estrutura da árvore de busca conservativa

Em [52] é proposto um método de busca conservativa em árvore para a determinação rápida do vizinho mais próximo. Todavia, neste método os hiperplanos possuem orientações arbitrárias. Neste caso, a equação de cada hiperplano exige o armazenamento de  $K+1$  valores, sendo  $K$  a dimensão do espaço vetorial em questão. Para determinarmos de que lado do hiperplano divisor encontra-se um determinado vetor, precisamos calcular o produto interno entre este vetor e o vetor relativo à equação do hiperplano [52]. Isto exige maior complexidade computacional e mais memória para armazenamento do que no caso da utilização de hiperplanos ortogonais a um dos eixos. Além disso, a flexibilidade obtida com hiperplanos de orientações quaisquer não parece conduzir a um desempenho superior [67].

### 4.3.2- OUTROS MÉTODOS DE BUSCA RÁPIDA

Em [40] é proposto um método de busca rápida baseado na distância parcial. O cálculo da distância parcial consiste simplesmente em abortar a computação da distorção  $d(X, \hat{X}_i)$ , tão logo se saiba que  $\hat{X}_i$  não é o vizinho mais próximo a  $X$  [40]. Isto pode ser feito inserindo-se uma operação de comparação para cada iteração do cálculo da distância entre  $\hat{X}_i$  e  $X$ . Assim que a distância parcial entre o vetor  $X$  e uma "codeword" exceda a distância mínima  $d_{\min}$  já calculada, esta "codeword" é descartada economizando-se as operações de MAC que faltam para completar o cálculo da distância. Esta técnica é útil quando são utilizadas medidas de distorção acumulativas como, por exemplo, o erro quadrático  $d_2(X, \hat{X}_i)$ .

Em [66] foi utilizada uma estratégia semelhante ao cálculo da distância parcial. O teste efetuado baseia-se na comparação entre o erro absoluto associado a cada componente  $E_i = X - \hat{X}_i$  do vetor erro e a raiz quadrada da distância mínima já calculada. Este teste equivale a verificar se uma dada "codeword" está contida no hipercubo que circunscreve uma hiperesfera. Esta hiperesfera tem como raio a raiz quadrada da distância mínima  $d_{\min}$  já calculada, e é centrada no ponto correspondente ao vetor  $X$ . O princípio é que a distância de uma "codeword" ao vetor  $X$  só será menor do que  $d_{\min}$  caso esta "codeword" esteja contida na hiperesfera com raio  $\sqrt{d_{\min}}$  e centrada em  $X$ . Uma condição necessária (mas não suficiente) para que isto aconteça, é que a "codeword" que está sendo testada esteja contida no hipercubo correspondente. Os autores de [66] declaram que o método apresentado é superior ao do cálculo da distância parcial. Todavia, não levaram em consideração a exigência do cálculo da raiz quadrada nesta comparação por considerá-lo pouco freqüente e por não haver necessidade de precisão. Este fato merece uma avaliação mais cuidadosa já que a extração da raiz quadrada pode ser comparada, em termos de complexidade, a 25 operações MAC [210]. Em [98], os autores modificam este método e reduzem sua complexidade computacional. Por outro lado, o método apresentado em [98] não garante encontrar a melhor "codeword", podendo ocasionalmente incorrer em erro.

Em [96] é mostrado que para a eficiência da distância parcial é importante iniciarmos a busca no "codebook" por uma boa aproximação da melhor "codeword". Isto reduz o número médio de cálculos. O desempenho de [66] e [98], assim como de muitos outros métodos, também depende de uma boa estimativa inicial.

Muitos dos algoritmos de busca rápida baseiam-se na desigualdade triangular [163], [186]. Caso a medida de distorção utilizada obedeça à desigualdade triangular, teremos:

$$d(\hat{X}_i, \hat{X}_j) \leq d(\hat{X}_i, X) + d(\hat{X}_j, X) \quad (4.3)$$

Neste caso, é fácil mostrar que se:

$$d(\hat{X}_i, \hat{X}_j) \geq 2d(\hat{X}_i, X) \quad (4.4)$$

então, a seguinte desigualdade é verdadeira:

$$d(\hat{X}_j, X) \geq d(\hat{X}_i, \hat{X}_j) \quad (4.5)$$

A maneira mais óbvia de empregarmos a desigualdade triangular para diminuirmos o número de cálculos na etapa de codificação da QV, seria armazenarmos uma tabela com todas as distorções entre as "codewords"  $d(\hat{X}_i, \hat{X}_j)$  pré-calculadas [186]. Uma vez calculada a distorção  $d(\hat{X}_i, X)$  para uma "codeword"  $i$ , podemos descartar da busca todas aquelas que satisfazem (4.4). Caso (4.4) não seja satisfeita para determinada "codeword"  $k$ , calculamos  $d(\hat{X}_k, X)$  e se  $d(\hat{X}_k, X)$  for menor do que  $d(\hat{X}_i, X)$ , o novo valor de referência passa a ser  $d(\hat{X}_k, X)$ . O problema desta abordagem é a quantidade de memória necessária para o armazenamento da tabela contendo as  $N(N-1)/2$  distâncias, onde  $N$  é o número de "codewords". Em [163] é apresentada uma técnica baseada na desigualdade triangular que não requer o armazenamento desta tabela. Em [186] é mostrada uma estratégia para escolher a primeira "codeword" a ser testada (estimativa inicial). Assim, da mesma forma do que os outros já citados, os métodos baseados na desigualdade triangular também dependem de uma boa estimativa inicial.

Para diminuir a necessidade de memória para armazenamento, pode-se partir para outras considerações geométricas. Uma destas seria tomar como base apenas a norma dos vetores, ou equivalentemente, a sua energia. A partir de uma tabela com as normas de cada uma das  $N$  "codewords", pode-se restringir a busca ao espaço delimitado por hiper-anéis [186]. Esta técnica não poderia ser utilizada para as estruturas ganho-forma, uma vez que as "codewords" possuem energia normalizada.

Uma evolução dos métodos baseados na desigualdade triangular, são aqueles que utilizam o conceito de pontos de âncora [19], [187], [204]. Podemos enxergar estes métodos como uma solução intermediária entre o armazenamento apenas da energia das "codewords" e o armazenamento de todas as distâncias entre elas. No caso dos pontos de âncora, armazenam-se as distâncias entre cada "codeword" a cada um dos pontos de âncora. Em [19] são utilizados três pontos de âncora, mas o método é suscetível a erros [187]. Em [187] é apresentada uma evolução de [19], considerando o número de pontos de âncora um critério definido pelo projetista. Em [204] são definidas estratégias para determinar o número de pontos de âncora e seus respectivos valores, baseadas nas estatísticas de uma seqüência representativa de vetores.

Em [162] é apresentada uma estratégia de busca baseada na vizinhança entre as "codewords". O pré-processamento consiste na determinação, para cada "codeword"  $\hat{X}_i$ , das "codewords" cujas partições de Voronoi são vizinhas à partição relativa a  $\hat{X}_i$ , armazenando-se os índices destas "codewords" na forma de uma tabela. O método baseia-se no fato de que, calculada a distorção  $d(\hat{X}_i, X)$ , caso as distorções entre  $X$  e todas as "codewords" vizinhas à  $\hat{X}_i$  sejam maiores do que  $d(\hat{X}_i, X)$ ,  $\hat{X}_i$  é a melhor "codeword". No caso de haver uma "codeword" vizinha  $\hat{X}_i$  que apresente

distorção menor, repete-se o processo testando-se todas as vizinhas de  $\hat{X}_i$ . Este método requer, ainda mais do que os demais, uma boa estimativa inicial. Por isso, os autores utilizam um quantizador escalar para inicializar o processo. Os testes apresentados em [162] foram feitos com vetores de dimensão reduzida,  $K=4$ . Os autores não abordaram o problema da dependência não-linear entre o número médio de partições vizinhas a uma dada partição e a dimensão do espaço [17], o que pode comprometer a eficiência do método proposto em [162].

O desempenho de muitos dos algoritmos descritos acima depende da estimativa inicial, ou seja, de quão rápido obtém-se uma "codeword" que permita descartar muitas outras. Nestes casos é muito difícil fazer uma estimativa *a priori* do pior caso. Para isto seriam necessários testes estatisticamente significativos para obter-se uma estimativa aceitável. Além disso, muitas das avaliações disponíveis na literatura são feitas a partir do valor esperado do número de operações [1]. Todavia, em implementações em tempo real, o sistema deve ser projetado para atender ao pior caso. A árvore K-d permite ao projetista determinar o pior caso de forma exata, a partir do número de "codewords" da folha mais populosa.

Dos métodos descritos acima, os que se mostram mais promissores são os baseados no trabalho de Sethi [19] e utilizam pontos de âncora [187], [204].

A utilização de métodos baseados na desigualdade triangular (Eq. (4.3)) na aplicação específica da QV das LSFs apresenta restrições, devido nesta aplicação geralmente ser utilizada uma medida dinâmica de distorção [194]. Em nosso conhecimento, não existe uma forma direta de conciliar uma medida dinâmica de distorção com os métodos baseados na desigualdade triangular. Uma vantagem da árvore K-d é a possibilidade de incorporar uma medida dinâmica de distorção no projeto da árvore. Para isto, basta utilizar a ponderação do erro na etapa de estimativa dos limites das projeções das partições de Voronoi através do método de Monte Carlo.

#### 4.4- IMPLEMENTAÇÃO DA ÁRVORE K-DIMENSIONAL

A árvore K-d exige o armazenamento dos valores  $j$  (índice do eixo ortogonal ao hiperplano divisor) e  $h$  (interseção do hiperplano com o eixo  $j$ ) para cada nó da árvore. Além disso, devemos armazenar os índices das "codewords" contidas em cada folha. Para uma árvore balanceada (todos os nós terminais estão no mesmo nível de profundidade), cada um dos  $2^d - 1$  nós requer o armazenamento de  $j$  e  $h$ , onde  $d$  é o nível de profundidade da árvore, ou seja, a especificação de um hiperplano para cada nó requer  $2(2^d - 1)$  posições de memória. Para cada uma das  $2^d$  folhas, devemos armazenar além dos índices, um indicador de final da lista. Assim, considerando  $\bar{b}$  como sendo o número médio de "codewords" nas folhas, devemos reservar  $2^d(\bar{b} + 1)$  posições de memória para o armazenamento das folhas. O espaço total necessário para o armazenamento da árvore em número de posições de memória estática (ME) é então [171]:

$$ME = 2(2^d - 1) + 2^d(\bar{b} + 1) \quad (4.2)$$

A determinação da estrutura de busca em árvore K-d consiste no cálculo dos hiperplanos divisores. A técnica utilizada para o cálculo dos hiperplanos independe da estratégia de busca a ser empregada. A partir de um mesmo método de cálculo dos hiperplanos, podemos implementar uma estratégia de busca simples ou conservativa.

A técnica de menor complexidade para a determinação dos hiperplanos divisores é baseada nas variâncias dos componentes dos vetores nas diversas direções. Escolhemos como eixo  $j$  aquele correspondente à direção cujas componentes tenham a maior variância. O valor de  $h$  é dado pela média das componentes dos vetores no eixo  $j$  [171]. Esta técnica não exige o conhecimento do diagrama de Voronoi. Isto permite seu uso na etapa de projeto do "codebook" [57], [97], momento no qual as "codewords" são modificadas a cada iteração (caso do LBG) ou não são disponíveis (caso do PNN). No segundo caso, o diagrama de Voronoi respectivo não poderia nem mesmo ser determinado. Assim, o método da variância permite que projetemos a árvore K-d sem que tenhamos qualquer conhecimento do diagrama de Voronoi. Todavia, devemos implementar uma estratégia de busca simples, pois a estratégia de busca conservativa exige algum conhecimento do diagrama de Voronoi.

Existem outras técnicas de determinação dos hiperplanos divisores. Estas técnicas exigem algum conhecimento do diagrama de Voronoi respectivo ao "codebook" a ser organizado. Na verdade, não é necessário calcular o diagrama de Voronoi em si, mas sim os limites das projeções das partições de Voronoi nos eixos das coordenadas. No caso bidimensional, por exemplo, deveríamos determinar para cada partição (correspondendo a uma "codeword"), os limites superior e inferior de sua projeção nos eixos  $x$  e  $y$ .

Uma vez determinados os limites das projeções das partições de Voronoi, o passo seguinte consiste na determinação dos hiperplanos divisores a partir de um critério de custo. Os métodos existentes, diferenciam-se principalmente no critério de custo utilizado.

Em [52] e [67] o critério utilizado para a determinação dos hiperplanos foi o critério do máximo produto [171]. Para cada um dos  $K$  eixos é realizada uma varredura nos valores de  $h$ , escolhendo-se o valor de  $h$  que conduzir ao máximo produto  $N_L N_R$ , onde  $N_R$  é o número de partições contidas inteiramente no lado direito do hiperplano e o número  $N_L$  de partições contidas inteiramente no lado esquerdo. Os limites das projeções das partições são necessários para determinar  $N_R$  e  $N_L$ , e também para realizar as sucessivas divisões do espaço através dos hiperplanos.

O artigo [171] combate a técnica do máximo produto e apresenta duas outras técnicas. Estas técnicas realizam um procedimento de varredura similar ao explicado. Contudo, o critério de custo para escolha dos valores de  $j$  e  $h$  é distinta.

A primeira técnica é chamada de GOC ("Generalized Optimization Criterion") [171]. O GOC consiste na determinação, para cada eixo, do mínimo valor absoluto da diferença entre o número de

regiões totalmente à esquerda e à direita do hiperplano:  $|N_R - N_L|$ . Assim, no GOC, os valores  $j$  e  $h$  escolhidos são os que conduzirem ao menor valor de  $|N_R - N_L|$ .

A segunda destas técnicas é chamada de EOC ("Exact Optimization Criterion") e baseia-se na minimização do valor esperado da complexidade de busca. A idéia consiste em realizar o projeto da árvore levando em consideração a probabilidade de ocorrência de cada "codeword". Obtém-se as probabilidades necessárias para o cálculo do valor esperado da complexidade de busca [171] utilizando-se uma seqüência de treinamento (que pode ser a mesma utilizada no projeto do "codebook"). Na técnica EOC, procura-se agrupar as "codewords" de maior probabilidade de ocorrência em folhas pouco populosas e as "codewords" de menor probabilidade em folhas mais populosas. Dessa forma, obtém-se uma diminuição do número médio de "codewords" pesquisadas, uma vez que as folhas que serão pesquisadas com maior freqüência possuem um número menor de "codewords".

#### 4.4.1- DETERMINAÇÃO DOS LIMITES DAS PROJEÇÕES DAS PARTIÇÕES DE VORONOI

Podemos obter as projeções caso tenhamos calculado analiticamente os vértices das partições de Voronoi [17], [18] (considerando-se que a medida de distorção utilizada seja adequada, como o erro quadrático, por exemplo). Todavia, isto é uma tarefa árdua computacionalmente. Além disso, as partições externas - que possuem vértices no infinito - não têm seus limites determinados. Uma forma alternativa de abordar este problema é utilizar o método Monte Carlo.

Para o método de Monte Carlo, devemos compor uma seqüência de vetores estatisticamente válida, que pode ser a própria seqüência utilizada no projeto do "codebook", ou ainda vetores obtidos a partir de uma distribuição uniforme [52]. Codificamos esta seqüência, utilizando a busca exaustiva para achar qual "codeword" corresponde ao vizinho mais próximo de um dado vetor. Sabendo que este vetor pertence à região de Voronoi da respectiva "codeword", utilizamos as próprias amostras deste vetor para estimar os limites das projeções. Este método estatístico requer que o número de vetores utilizados na seqüência de treinamento seja bastante elevado. Caso as projeções não sejam determinadas com precisão suficiente, pode haver aumento significativo na distorção média. A utilização do método Monte Carlo geralmente fornece intervalos menores do que os limites reais das projeções. Isto poderia ser combatido através da utilização de um limite de segurança, de forma a expandir os limites estimados pelo método Monte Carlo.

Uma vez que se disponha de um algoritmo para o cálculo analítico das partições de Voronoi [17], [18], pode-se adotar uma solução mista que utilize tanto o cálculo analítico quanto o método Monte Carlo. O algoritmo analítico pode fornecer os vértices das partições internas e, conseqüentemente, suas projeções, além de indicar quais as "codewords" que correspondem a partições externas. O método Monte Carlo é então utilizado apenas para as "codewords" que

correspondam a partições externas. Desta forma, aumenta-se a precisão das projeções de partições internas e reduz-se a complexidade computacional do método de Monte Carlo pois o número de projeções a ser estimado é menor.

A complexidade da determinação das projeções das partições de Voronoi aumenta exponencialmente com a dimensão do espaço em questão. Para valores elevados de  $K$ , tanto a determinação analítica quanto o método Monte Carlo assumem complexidades elevadas. Suponhamos que desejemos criar uma distribuição uniforme de pontos para um espaço de dimensão  $K=4$ . Com 100 pontos para cada eixo, teríamos  $100^4$  vetores. Assim, deveríamos codificar 100 milhões de vetores para estimarmos os limites das projeções (Monte Carlo). No caso de um espaço de dimensão  $K=16$ , usando apenas 10 pontos para cada eixo, teríamos  $10^{16}$  vetores, o que representa uma complexidade extremamente elevada.

Neste trabalho estimamos os limites das partições de Voronoi através do método de Monte Carlo, baseado na codificação da mesma seqüência de treinamento utilizada para o projeto do "codebook".

#### 4.4.2- PROJETO DA ÁRVORE K-d ATRAVÉS DO MÉTODO GOC

Os métodos GOC, EOC e do máximo produto baseiam-se em um procedimento de varredura, onde todos os  $K$  eixos (onde  $K$  é a dimensão do espaço) são pesquisados de forma seqüencial. Para cada eixo é realizada uma varredura nos valores de  $h$ , calculando-se o critério de custo utilizado.

Tentaremos explicar mais detalhadamente este processo de varredura já inserindo-o no algoritmo GOC. Este mesmo processo é utilizado nos algoritmos EOC e do máximo produto, modificando-se apenas o critério de custo utilizado.

- Para cada nó da árvore, inicializa-se o processo de determinação do hiperplano assumindo-se o menor valor possível de  $X(j)$  como valor inicial de  $h$ . Neste caso, não existem partições à esquerda do hiperplano e  $N_L=0$  (uma vez que o valor de  $h$  é o menor valor assumido por  $X(j)$ ), tornando o valor  $|N_R - N_L| = N_R$ . Incrementa-se o valor de  $h$  até o máximo valor assumido por  $X(j)$ , calculando-se  $|N_R - N_L|$  para cada valor de  $h$ . Neste trabalho, o passo utilizado para este incremento foi determinado a partir dos valores mínimo e máximo de  $X(j)$ , de forma que para cada eixo fossem testados 1000 valores de  $h$  (para cada nó e para cada eixo determinava-se um valor de passo). Assim, para cada um dos  $K$  eixos, determina-se o valor de  $h$  que conduz ao menor valor  $|N_R - N_L|$  e dentre estes  $K$  hiperplanos (pares de valores de  $j$  e  $h$ ) escolhe-se o hiperplano correspondente ao menor valor de  $|N_R - N_L|$ .

Por exemplo, vamos supor que estamos projetando uma estrutura em árvore para realizar a busca em um "codebook" com  $N=1024$  "codewords" de dimensão  $K=6$ . Os  $2 N K$  limites (inferior e superior) das projeções das  $N$  partições já foram determinados (pelo método de Monte Carlo, por

exemplo). Supomos também, que adotamos o critério de busca conservativa e que o primeiro hiperplano ( $j(0)=4$  e  $h(0)=3,65$ ) já foi calculado. Este hiperplano divide o espaço em duas regiões.

O passo seguinte é determinarmos as "codewords" que constituirão cada uma das duas regiões. Para cada "codeword", comparamos seus limites (inferior e superior) das projeções correspondentes ao eixo  $j=4$  com o valor  $h=3,65$ . No caso do limite superior (para o eixo  $j=4$ ) da partição  $i$  ser menor do que  $h=3,65$  significa que toda partição está contida à esquerda do hiperplano e a "codeword"  $i$  participará apenas da região à esquerda. Caso o limite inferior (para  $j=4$ ) seja maior do que  $h=3,65$  significa que a partição está contida inteiramente à direita do hiperplano. No caso do valor  $h=3,65$  estar situado entre os limites inferior e superior, a partição  $i$  é atravessada pelo hiperplano e a "codeword"  $i$  participará das duas regiões. O algoritmo continua, buscando-se agora a determinação do hiperplano  $j(1)$  e  $h(1)$  que subdividirá a região à esquerda. Utiliza-se o processo de varredura já descrito, onde os cálculos dos valores de  $N_L$  e  $N_R$  baseiam-se no mesmo raciocínio usado para a divisão das "codewords". Desta forma, os hiperplanos vão sendo calculados até que se atinja o nível de profundidade da árvore desejado.

#### 4.5- CONCLUSÕES

Neste capítulo foi apresentada a árvore K-dimensional aplicada à busca rápida da melhor "codeword" em QV. Na seção 4.2 apresentamos o conceito de partições de Voronoi no qual se baseia o algoritmo de busca em árvore K-d. A árvore K-d é introduzida na seção 4.3 como um método para busca associativa, objetivo para o qual foi concebida por Bentley [7]. Citamos também a utilização da árvore K-d pelo algoritmo PNN [97] no projeto do "codebook". A utilização da árvore K-d é o motivo pelo qual o PNN é mais rápido do que o LBG. Enfocamos a utilização da árvore K-d em QV como um método de busca rápida da melhor "codeword". Buscamos nesta seção definir uma nomenclatura que diferencie o projeto da árvore no tocante à busca conservativa e à busca simples (não-conservativa). Ainda na seção 4.3, fizemos uma breve descrição de outros métodos de busca rápida utilizados em QV. Os métodos de busca rápida do vizinho mais próximo não são um assunto novo, mas ainda constituem-se em assunto de interesse atual, devido principalmente à expansão do emprego da QV. Na seção 4.4 procuramos fornecer detalhes de implementação e da etapa de projeto da árvore K-d. Abordamos também a estimativa dos limites das projeções das partições de Voronoi que é de importância crucial no projeto da árvore K-d.

# CAPÍTULO 5

## AValiação de Medidas de Distorção para a QV das LSFs

### 5.1- INTRODUÇÃO

A aplicação do conceito de transparência baseado na distorção espectral (SD) à QV das LSFs encontra-se amplamente difundido [146], [194], [210], [173]. Este conceito facilita os trabalhos que tratam especificamente da quantização das LSFs, uma vez que a realização de testes subjetivos com voz sintetizada é uma tarefa árdua.

Embora as LSFs sejam parâmetros relacionados ao domínio da frequência, não existe uma relação simples entre a SD e o erro quadrático calculado a partir dos vetores das LSFs quantizadas e não-quantizadas [211]. Por este motivo, a maioria dos sistemas de quantização vetorial de LSFs utilizam medidas dinâmicas de distorção baseadas na ponderação do erro quadrático. O objetivo destas ponderações é alcançar maior correlação com a distorção espectral e buscar maior consonância com a avaliação subjetiva. Esta segunda intenção normalmente não aparece de forma explícita, mantendo-se atrelada ao fato de que um bom desempenho em termos de distorção espectral é tido como sinônimo de qualidade subjetiva. Algo que começa a ser questionado por alguns especialistas [190].

Nosso objetivo neste capítulo é avaliar as medidas de ponderação propostas, principalmente nos aspectos de complexidade e distorção associadas. Na seção 5.2 especificamos as condições nas quais as simulações foram executadas. Apresentaremos também a medida de complexidade computacional que será utilizada nas avaliações realizadas neste trabalho. Na seção 5.3 apresentaremos a formulação das medidas de distorção mais utilizadas na QV das LSFs. A seção 5.4 apresenta os dados estatísticos destas medidas. Estas estatísticas são importantes para a compreensão da funcionalidade das medidas de distorção em questão. Além disso, apresentaremos na seção 5.4 estimativas da complexidade computacional das medidas de distorção. Na seção 5.5 avaliaremos o efeito da ponderação do erro quadrático. Utilizou-se uma estratégia alternativa para esta avaliação que não se limita a fornecer apenas o desempenho final do quantizador. Nossa estratégia baseia-se em informação proveniente da quantização a partir da própria SD e permite uma avaliação mais precisa do que a normalmente utilizada na literatura. As conclusões deste capítulo serão apresentadas na seção 5.6.

### 5.2- CONSIDERAÇÕES GERAIS

Já mencionamos no Capítulo 3 que a complexidade computacional associada à implementação de um quantizador vetorial depende da memória requerida e do número de operações necessárias à

codificação (Fig. 3.3). Na avaliação desta complexidade é útil contarmos com uma só medida que englobe estes dois aspectos. Esta medida seria dependente do compromisso entre a utilização de um processador mais poderoso e um maior dispêndio com memória para armazenamento, ou seja, dependente do custo de mercado. Adotaremos a mesma estratégia utilizada por Boisson de Marca em [210], que define uma medida de complexidade (C) como:

$$\text{Complexidade (C)} = \text{Memória Dinâmica (MD)}/5 + \text{Mem. Estática (ME)}/20 + \text{Operações Aritméticas (O)} \quad (5.1)$$

O valor total de operações O é uma soma ponderada das operações aritméticas envolvidas. O peso associado a cada tipo de operação é dado na Tabela 5.1.

Tabela 5.1 - Pesos associados às operações aritméticas [210]

Operação	Exemplo	Peso
Adições	$() = () + ()$	1
Multiplicações	$() = () * ()$	1
Multipl. e acumulação (MAC)	$() = () * () + ()$	1
Transferência de dados	float, int	1
Operações lógicas	shift, AND, OR	1
Divisões	$() = () / ()$	18
Raízes quadradas	$() = \text{sqrt}()$	25
Funções transcendentes	seno, log	25
Testes aritméticos	if, if then else	2
Adiciona-compara-seleciona	descodificação Viterbi	6

Neste trabalho utilizaremos a distorção espectral (SD) para a avaliação final da qualidade da quantização, em conformidade com a literatura (ex: [194]). Apresentaremos a SD média e também as porcentagens de quadros inseridas nas três principais faixas para a avaliação da qualidade da quantização [146].

A SD relativa à quantização de um único filtro LPC é dada pela equação (2.17) que repetiremos abaixo por conveniência [69].

$$SD_n = \sqrt{\frac{1}{\pi} \int_0^{\pi} \left( 10 \log \frac{S_n(\omega)}{\hat{S}_n(\omega)} \right)^2 d\omega} \quad (5.2)$$

onde  $S_n(\omega)$  é o espectro de potência do filtro LPC original,  $\hat{S}_n(\omega)$  é o espectro de potência do filtro LPC quantizado e  $SD_n$  é a distorção espectral em dB.

O trabalho de Soong e Juang definia a SD média como a raiz quadrada da média de  $SD_n^2$  ([69], Eq. 5.18). A mesma definição é adotada em muitos outros artigos (ex: [211] e [191]). Todavia, temos observado uma tendência cada vez mais forte em definir a SD média como a média de  $SD_n$ , que seria a definição mais natural (ex: [216] e [210]). Por este motivo, preferimos adotar no presente trabalho a definição para a SD média expressa abaixo:

$$\overline{SD} = \frac{\sum_{n=1}^N SD_n}{N} \quad (5.3)$$

Calculamos  $SD_n$  a partir de 200 pontos do espectro de potência e fazemos a integração numérica baseada em uma interpolação de segundo grau (primeira regra de Simpson) [28]. Este número de pontos foi determinado experimentalmente de forma a obtermos precisão de uma a duas casas decimais. A expressão utilizada para o cálculo do espectro de potência é [76]:

$$S(\omega) = |H(e^{j\omega})|^2 = \frac{2^{-P}}{\sin^2(\omega/2) \prod_{i=2,4,\dots,P} (\cos\omega - \cos\omega_i)^2 + \cos^2(\omega/2) \prod_{i=1,3,\dots,P-1} (\cos\omega - \cos\omega_i)^2} \quad (5.4)$$

onde  $P=10$  é a ordem do filtro LPC.

A partir da Eq. (5.1), estimamos a complexidade computacional do cálculo da Eq. (5.4) para uma dada frequência. Procuramos otimizar a rotina de acordo com os fatores definidos na Eq. (5.1) e na Tabela 5.1. Este resultado é dado na Tabela 5.2.

Tabela 5.2 - Complexidade do cálculo de um ponto do espectro de potência (Eq. (5.4))

Mem. estática (ME)	Mem. dinâmica (MD)	Operações (O)	Complexidade (C)
4	28	451	456,8

Para estudarmos a quantização vetorial do filtro LPC, seguimos o procedimento habitual, utilizando uma seqüência para o treinamento dos "codebooks" diferente da seqüência utilizada para os testes. Todos os "codebooks" foram projetados com o algoritmo LBG, usando a inicialização do tipo "splitting" [15]. A seqüência de treinamento foi constituída a partir de elocuções de durações variáveis de 23 pessoas (11 homens, 11 mulheres e 1 criança), amostradas a uma frequência de 8 kHz e quantizadas com 12 bits. A seqüência de teste foi constituída por 11 elocuções de pessoas

diferentes (7 homens e 4 mulheres) que não participam da seqüência de treinamento. Para a seqüência de treinamento deslocamos a janela em intervalos de 5 ms, resultando em um conjunto com 72.000 filtros LPC. Para a seqüência de teste deslocamos a janela em intervalos de 20 ms, correspondendo a 8.000 filtros. O método utilizado para a análise LPC foi o da autocorrelação [176].

O número de vetores utilizados na seqüência de teste é muito importante. Em concordância com outros trabalhos (ex: [212]), havíamos inicialmente utilizado em nossas pesquisas uma seqüência de teste com 2.000 filtros LPC [191]. Todavia, com o decorrer das investigações, pudemos constatar que a distorção espectral média é bastante dependente do locutor. Por este motivo, aumentamos o número de locutores (e conseqüentemente o número de vetores de teste) para 8.000 de forma a tornar a seqüência de teste mais representativa. Outros trabalhos nesta linha de investigação também utilizam para a avaliação um número de vetores nesta faixa [194].

### 5.3- MEDIDAS DE PONDERAÇÃO

Quando aplicado à comparação entre dois vetores de LSFs, o erro quadrático atribui a mesma importância (pondera uniformemente) a todas as regiões do espectro. Entretanto, seria desejável realizar uma quantização mais precisa das LSFs que correspondam às regiões dos formantes, em detrimento das LSFs respectivas aos vales do espectro [210]. Em [146], Atal e Paliwal propuseram uma medida dinâmica de distorção baseada no espectro de potência do filtro LPC. Nesta medida, o peso atribuído a uma dada LSF é proporcional ao valor do espectro de potência LPC na frequência desta LSF. Esta ponderação também dá maior peso às LSFs correspondentes aos formantes de maior amplitude do que às correspondentes aos formantes de amplitudes menores. Desde então, esta medida vem sendo muito utilizada [173], [190], [194], [213], [214], [215]. Sua expressão é dada por:

$$d(X, \hat{X}) = \sum_{i=1}^{10} \left[ z_i (S(w_i))^r (w_i - \hat{w}_i) \right]^2 \quad (5.5)$$

onde o vetor a ser quantizado  $X$  é formado pelas LSFs não-quantizadas  $w_i$  e  $\hat{X}$  é o vetor formado pelas LSFs quantizadas  $\hat{w}_i$ . O fator  $z_i$  é determinado empiricamente [146], sendo dado por:

$$z_i = \begin{cases} 1,0 & \text{para } 1 \leq i \leq 8 \\ 0,8 & \text{para } i = 9 \\ 0,4 & \text{para } i = 10 \end{cases} \quad (5.6)$$

O valor utilizado por Atal e Paliwal para a constante empírica  $r$  é de 0,15, enquanto Collura e Tremain [190] usam  $r=0,12$ . Neste trabalho utilizamos  $r=0,15$  por termos obtidos com este valor

melhores resultados para os 8.000 vetores da seqüência de teste. Ressaltamos que em outra simulação obtivemos, para uma seqüência de teste com um número menor de vetores, resultados ligeiramente melhores com  $r=0,12$ . Observa-se que as duas últimas LSFs são desenfaturadas devido ao sistema auditivo ser menos sensível às altas freqüências [194]. Chan et al mencionam em [215] que se pode obter uma redução da distorção espectral média (Eq. (5.3)) substituindo-se cada peso definido em [146] (Eq. (5.5)) por sua raiz quadrada. Os autores relatam uma diminuição de aproximadamente 0.15 dB na distorção média, correspondendo em alguns casos a 1 bit por filtro LPC. Neste trabalho, optamos por não incorporar esta modificação.

Uma vez que o objetivo é dar maior peso às LSFs correspondentes às regiões dos formantes, Farvardin et al [156] propuseram outra medida de ponderação. Os autores aproveitaram o fato de que existem no mínimo duas LSFs próximas uma da outra na freqüência em que o espectro possui um pico (região de formante) [53]. Juntamente com a Eq. (5.5), a medida proposta por Farvardin et al é uma das mais utilizadas [157], [210], [211], [212], sendo dada por [156]:

$$d(X, \hat{X}) = \sum_{i=1}^{10} v_i (w_i - \hat{w}_i)^2 \quad (5.6)$$

onde:

$$v_i = \frac{1}{w_i - w_{i-1}} + \frac{1}{w_{i+1} - w_i} \quad (5.7)$$

com os valores limites sendo dados por  $w_0 = 0$  e  $w_{11} = \pi$ .

Outra classe de medidas de ponderação são as baseadas na sensibilidade do espectro a uma pequena variação na LSF [69], [122], [174]. Em [37] é considerada a sensibilidade do atraso de grupo [37]. Das duas medidas baseada em sensibilidades, a mais utilizada é a primeira. Esta medida é dada pela Eq. (5.6) com os pesos  $v_i$  calculados a partir da expressão abaixo [69], [122]:

$$v_i = \sqrt{\int_0^\pi \left| \frac{\partial \log S(\omega)}{\partial w_i} \right| d\omega} \quad (5.8)$$

Em [173], Paksoy et al observa que as medidas propostas em [146] (Eq. (5.5)), [156] (Eq. (5.6)) e [37] (Eq. (5.8)) conduziram a resultados equivalentes. Farvardin et al indica que a medida de ponderação proposta em [156] (Eq. (5.6)) apresenta desempenho semelhante à ponderação baseada na sensibilidade do espectro [122] (Eq. (5.8)). Sabe-se que quanto menor a distância entre duas LSFs, maior a sensibilidade espectral nesta região [69]. Desta forma, a ponderação de [156] está de acordo com a sensibilidade do espectro em relação às LSFs.

#### 5.4- ESTADÍSTICAS E COMPLEXIDADE DAS MEDIDAS DE PONDERAÇÃO

Quando utilizamos o erro quadrático ponderado, a "codeword" escolhida não é aquela que conduzir a menor norma do vetor erro, mas sim, aquela que conduzir ao menor produto interno entre o vetor erro quadrático e o vetor de ponderação. Na etapa de determinação da melhor "codeword" é calculado o produto interno  $\|W\| \|E_i\| \cos \overline{WE_i}$ , onde  $W$  é o vetor de ponderação e  $E_i$  é o vetor erro quadrático correspondente à "codeword"  $\hat{X}_i$ . A norma do vetor  $W$  é irrelevante para esta busca e portanto podemos normalizar os vetores de ponderação. Sem a normalização seria impraticável compararmos vetores de uma mesma medida de ponderação devido às diferentes normas dos vetores de ponderação. Além disso, a normalização permite compararmos cada um dos tipos de ponderação dentro das mesmas ordens de grandeza.

Ressaltamos que foi necessário uniformizarmos a notação das medidas de distorção para que pudessemos tratar a ponderação do erro quadrático como um produto interno. Para a medida da Eq. (5.5), por exemplo, o vetor de ponderação  $W$  é composto pelo fator  $z_i(S(w_i))^r$  elevado ao quadrado.

As estatísticas de cada uma das medidas de ponderação normalizadas foram calculadas a partir da seqüência de treinamento de 72.000 vetores. Os resultados são mostrados nas Tabelas 5.3, 5.4 e 5.5 para as ponderações propostas em [146], [156] e [69], respectivamente. Estas medidas de distorção serão chamadas daqui em diante de POTÊNCIA [146], DIFERENÇA [156] e SENSIBILIDADE [69] por uma questão de clareza. Estas tabelas apresentam os valores mínimo, máximo, médio e o desvio padrão ( $\sigma$ ) [1] das componentes dos vetores de ponderação para cada medida.

Tabela 5.3 - Estatísticas do vetor de ponderação normalizado para a distorção da POTÊNCIA

	1	2	3	4	5	6	7	8	9	10
Mín.	0,085	0,044	0,037	0,021	0,011	0,012	0,009	0,007	0,006	0,001
Máx.	0,970	0,898	0,824	0,741	0,762	0,709	0,773	0,848	0,720	0,224
Média	0,658	0,557	0,288	0,150	0,133	0,108	0,107	0,104	0,069	0,014
$\sigma$	0,132	0,135	0,124	0,088	0,095	0,081	0,089	0,088	0,068	0,013

Tabela 5.4-Estatísticas do vetor de ponderação normalizado para a distorção da DIFERENÇA

	1	2	3	4	5	6	7	8	9	10
Mín.	0,136	0,080	0,046	0,034	0,035	0,028	0,038	0,039	0,049	0,039
Máx.	0,852	0,743	0,679	0,678	0,676	0,651	0,646	0,724	0,695	0,662
Média	0,495	0,493	0,268	0,197	0,213	0,207	0,217	0,232	0,267	0,217
$\sigma$	0,114	0,120	0,104	0,092	0,096	0,088	0,077	0,079	0,099	0,090

Tabela 5.5-Estatísticas do vetor de ponderação normalizado para a distorção da SENSIBILIDADE

	1	2	3	4	5	6	7	8	9	10
Mín.	0,148	0,137	0,080	0,089	0,088	0,070	0,873	0,101	0,088	0,086
Máx.	0,853	0,877	0,717	0,626	0,627	0,568	0,562	0,501	0,518	0,435
Média	0,460	0,412	0,294	0,231	0,271	0,244	0,262	0,262	0,269	0,255
$\sigma$	0,127	0,125	0,091	0,076	0,082	0,068	0,063	0,056	0,064	0,052

Os histogramas correspondentes à ponderação da POTÊNCIA, DIFERENÇA e SENSIBILIDADE aparecem nas Figs. 5.1, 5.2 e 5.3, respectivamente. Indicamos nestas figuras, as curvas relativas às ponderações das duas primeiras LSFs. Para a ponderação da POTÊNCIA (Fig. 5.1) indicamos também as curvas respectivas aos pesos da terceira e das duas últimas LSFs.

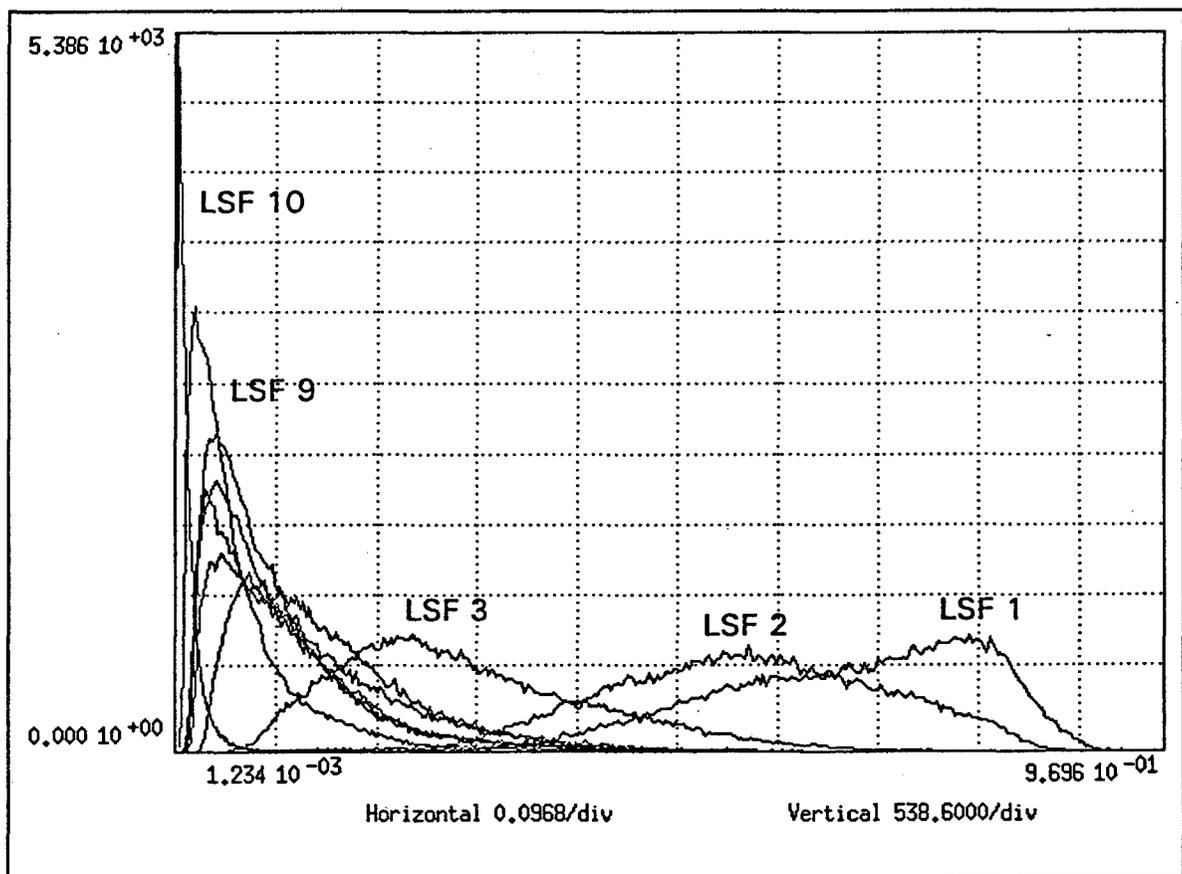


Fig. 5.1 - Histograma da medida da POTÊNCIA

Podemos observar da Fig. 5.1 que o fato do espectro de potência possuir normalmente o aspecto "passa-baixas", os pesos correspondentes às duas primeiras LSFs possuem em média um valor maior. Outra observação é que a desenfaturação das duas últimas LSFs faz com que seus respectivos pesos sejam significativamente menores, principalmente para a última LSF.

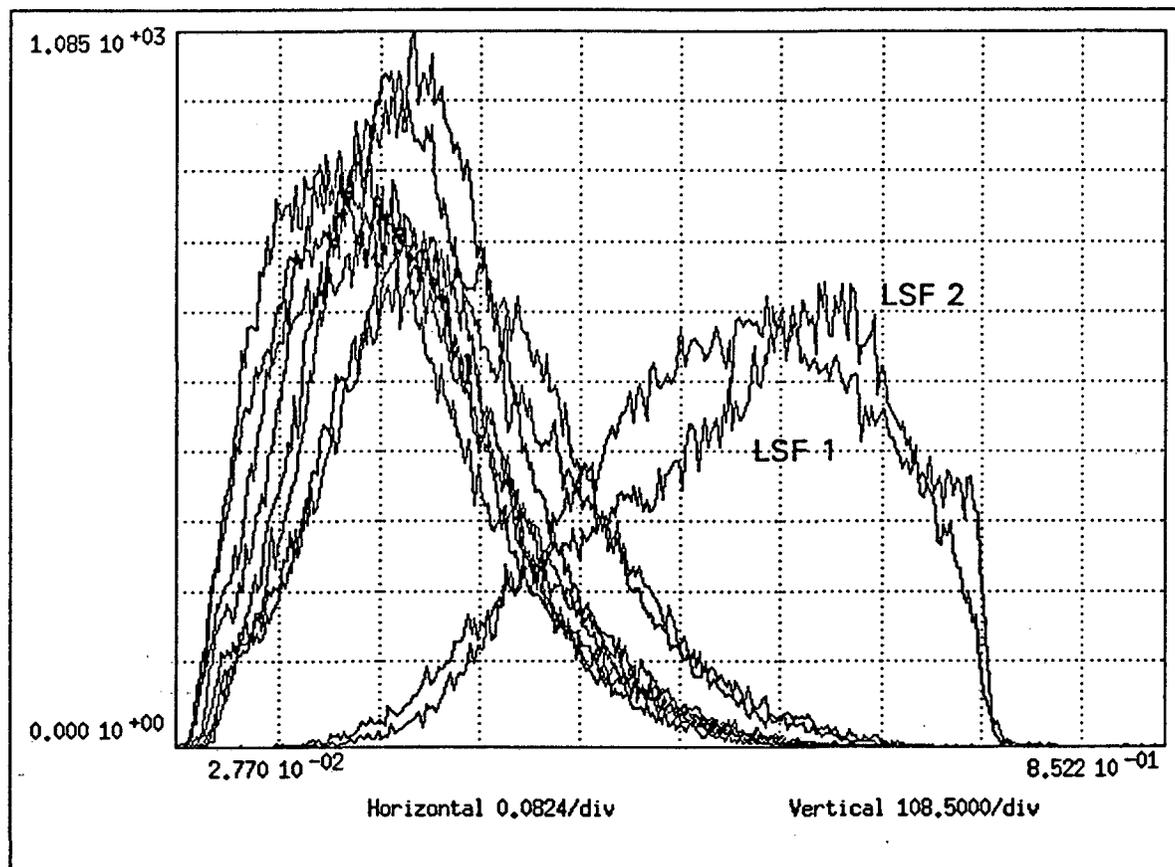


Fig. 5.2 - Histograma da medida da DIFERENÇA

Podemos observar da Fig. 5.2 que os pesos correspondentes às duas primeiras LSFs são em média maiores do que os pesos respectivos às outras. A ponderação através da DIFERENÇA confere às últimas 8 LSFs praticamente o mesmo grau de importância. Desta forma, esta medida não incorpora o aspecto subjetivo da menor importância das altas frequências de forma tão distribuída quanto a ponderação pela POTÊNCIA.

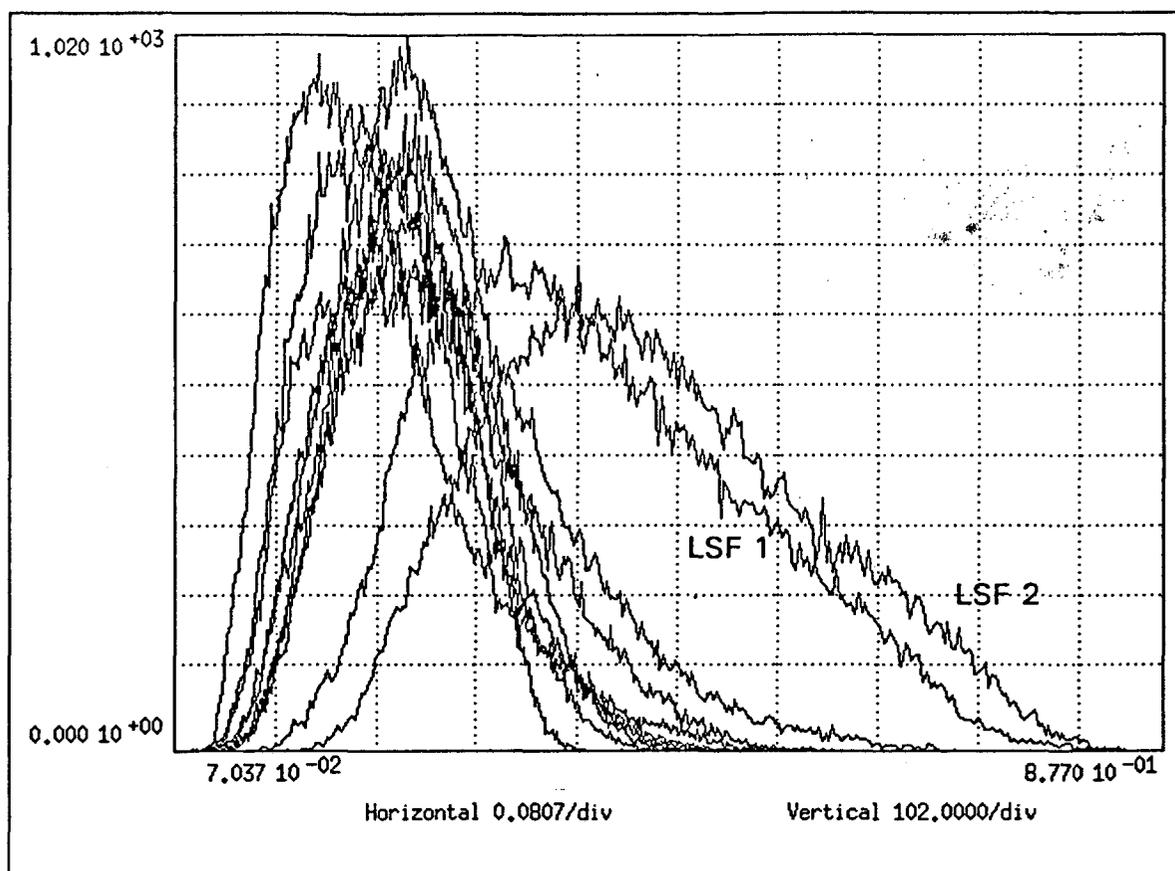


Fig. 5.3 - Histograma da medida da SENSIBILIDADE

A Fig. 5.3 mostra que o espectro de potência é mais sensível às duas primeiras LSFs. As oito LSFs restantes não apresentam distinção significativa em termos de importância. Esta característica das LSFs difere da apresentada pelos PARCORS que são ordenados em termos de importância.

A partir dos pesos de cada operação aritmética (Tabela 5.1) estimamos a complexidade do cálculo do vetor de ponderação para um dado vetor de LSFs a ser quantizado. A Tabela 5.6 mostra esta complexidade (C) para as três medidas de distorção. Dentro do possível, as rotinas para o cálculo dos vetores de ponderação da POTÊNCIA e da DIFERENÇA foram otimizadas em relação à complexidade (C).

Tabela 5.6 - Complexidade do cálculo do vetor de ponderação

	Mem. estática (ME)	Mem. dinâmica (MD)	Operações (O)	Complexidade (C)
POTÊNCIA	6	39	2364	2372,1
DIFERENÇA	4	36	288	295,4
SENSIBILIDADE	12	170	60000	60034,6

Podemos observar da Tabela 5.6 que a medida de distorção da SENSIBILIDADE apresenta uma complexidade computacional muito maior que as outras duas. Para esta medida realizamos apenas uma estimativa grosseira com o intuito de definir a ordem de grandeza de sua complexidade. Os valores exatos da complexidade associada à medida da SENSIBILIDADE são superiores aos apresentados.

## 5.5- AVALIAÇÃO DO EFEITO DA PONDERAÇÃO

Para estudarmos o efeito da ponderação do erro quadrático no desempenho dos quantizadores vetoriais, dividimos a tarefa em duas etapas. Esta metodologia será justificada no decorrer desta seção. Na primeira etapa, quantizamos o filtro LPC com apenas 10 bits utilizando um "codebook" sem restrições na estrutura (tais como particionamento ou multi-estágio). Na segunda etapa, estudamos a quantização do filtro LPC com 24 bits utilizando o particionamento proposto em [146] e [194].

Para a primeira etapa, projetamos três "codebooks" compostos de 1024 "codewords" de dimensão 10, correspondendo às LSFs respectivas ao filtro LPC de ordem 10. Para o primeiro, utilizamos o erro quadrático como medida de distorção, sem incorporar qualquer ponderação. Este "codebook" será chamado de CBK\_EQ. O segundo "codebook" foi projetado usando a ponderação da DIFERENÇA, sendo chamado CBK\_D. O terceiro utiliza a ponderação da POTÊNCIA, sendo chamado CBK\_P.

Já mencionamos que para medidas dinâmicas de ponderação, a centróide é dada pela Eq. (3.7). O problema do uso desta expressão é que a mesma não garante que o vetor de LSFs correspondente à centróide, mantenha a característica de ordenamento necessária para garantir a estabilidade do filtro LPC. Por este motivo, adotamos o mesmo procedimento de outros autores [173], [194], calculando a centróide a partir da média dos vetores que fizerem parte da respectiva região. Paksøy et al menciona em [173] que esta estratégia não implica em degradação do desempenho quando comparada à que utiliza a centróide consistente com a medida dinâmica de distorção (3.7).

Para cada um destes "codebooks", codificamos os 8000 vetores da seqüência de teste utilizando a SD como medida de distorção. Em outras palavras, determinamos para cada um dos vetores da seqüência de teste a "codeword" correspondente à menor distorção espectral. Com isto, obtivemos o limite superior para o desempenho destes "codebooks" (em termos de SD). Afirmamos isto porque qualquer outra medida de distorção utilizada para a busca conduzirá a um desempenho (em termos de SD) inferior ao obtido por este procedimento. Estes resultados são mostrados na Tabela 5.7. Mostramos também a entropia (ver por exemplo, [26], pp. 181) de cada um destes "codebooks". A entropia indica o valor médio da informação fornecida pelo "codebook". Neste caso a entropia ideal (máxima) seria igual a 10 bits, correspondendo à média de uma utilização de cada "codeword" a cada 1024 codificações.

Tabela 5.7 - Desempenho dos "codebooks" projetados com diferentes medidas de distorção

	CBK EQ (10 bits)	CBK P (10 bits)	CBK D (10 bits)
SD mínima (dB)	0.58	0.60	0.65
SD máxima (dB)	5.22	5.37	5.16
SD média (dB)	2,23	2,29	2,23
SD < 2 dB	36.27 %	33.76 %	36.44 %
SD entre 2 dB e 4 dB	63.41 %	65.49 %	63.09 %
SD > 4 dB	0.31 %	0.75 %	0.47 %
Entropia do "codebook"	9.42 bits	9.27 bits	9.43 bits

Podemos observar da Tabela 5.7 que as medidas de ponderação utilizadas no projeto do "codebook" não influenciaram significativamente no desempenho dos mesmos. O erro quadrático apresentou até mesmo ligeira superioridade em relação à distorção da DIFERENÇA.

Nas simulações correspondentes à Tabela 5.7 calculamos e armazenamos, para cada vetor da seqüência de teste (8000 vetores), a "codeword" que conduz à menor distorção espectral. Utilizamos esta informação para analisar a eficiência das medidas de ponderação na busca desta "codeword" para cada um dos "codebooks" da Tabela 5.7. Os resultados desta avaliação encontram-se nas Tabelas 5.8, 5.9 e 5.10 para os "codebooks" CBK\_EQ, CBK\_D e CBK\_P, respectivamente.

Nesta análise, consideramos que a "codeword" respectiva à menor SD é a correta e calculamos quantas vezes a medida de distorção utilizada acerta na busca, ou seja, encontra a "codeword" correta ('Acertos'). Colocando o quantizador em "loop", calculamos o número de tentativas erradas até que a "codeword" correta seja encontrada. Por exemplo, para cada vetor X a ser quantizado, inicializamos o contador de erros (contador igual a zero) e encontramos dentre as 1024 "codewords" aquela que conduz à menor distorção. Caso esta não seja a "codeword" correta, incrementamos o contador de erros e reiniciamos o processo de busca levando em conta as 1023 "codewords" restantes. Registramos a média do número de erros para todos os 8.000 vetores ('Média do núm. de erros') e o maior número de erros observado ('Pior caso do núm. de erros'). As três últimas linhas das tabelas indicam os erros máximo, mínimo e médio obtidos a partir da comparação com o resultado da codificação através da própria SD (Tabela 5.7).

Tabela 5.8 - Buscas para "codebook" projetado com erro quadrático (CBK EQ)

	Erro Quad.	POTÊNCIA	SENSIBIL.	DIFERENÇA
SD mínima (dB)	0,58	0,58	0,58	0,58
SD máxima (dB)	5,71	5,68	5,32	5,91
SD média (dB)	2,32	2,45	2,26	2,27
SD < 2 dB	33,12 %	27,95 %	35,24 %	35,09 %
SD entre 2 e 4 dB	65,84 %	69,57 %	64,16 %	64,20 %
SD > 4 dB	1,04 %	2,48 %	0,60 %	0,71 %
Acertos	64,84 %	49,01 %	78,60 %	75,96 %
Média do núm. de erros	1,91	3,27	1,36	1,52
Pior caso do núm. de erros	69	156	21	118
Erro máximo (dB)	1,66	2,92	0,85	1,36
Erro mínimo (dB)	0,09	0,22	0,03	0,04

Tabela 5.9 - Buscas para "codebook" projetado com medida da DIFERENÇA (CBK D)

	Erro Quad.	POTÊNCIA	SENSIBIL.	DIFERENÇA
SD mínima (dB)	0,65	0,65	0,65	0,65
SD máxima (dB)	5,65	5,97	5,52	5,39
SD média (dB)	2,33	2,43	2,26	2,27
SD < 2 dB	30,04 %	20,88 %	35,36 %	35,39 %
SD entre 2 e 4 dB	65,66 %	68,71 %	63,89 %	63,71 %
SD > 4 dB	1,3 %	2,41 %	0,75 %	0,90 %
Acertos	63,65 %	51,46 %	78,52 %	77,17 %
Média do núm. de erros	1,98	3,04	1,36	1,46
Pior caso do núm. de erros	61	131	17	70
Erro máximo (dB)	2,35	3,39	1,16	1,37
Erro mínimo (dB)	0,10	0,20	0,03	0,04

Tabela 5.10 - Buscas para "codebook" projetado com medida da POTÊNCIA (CBK P)

	Erro Quad.	POTÊNCIA	SENSIBIL.	DIFERENÇA
SD mínima (dB)	0,60	0,60	0,60	0,60
SD máxima (dB)	6,09	5,80	5,78	5,37
SD média (dB)	2,39	2,43	2,32	2,33
SD < 2 dB	30,49 %	29,05 %	32,70 %	32,66 %
SD entre 2 e 4 dB	67,53 %	68,41 %	66,23 %	66,12 %
SD > 4 dB	1,99 %	2,54 %	1,07 %	1,21 %
Acertos	62,00 %	55,75 %	77,10 %	75,94 %
Média do núm. de erros	1,98	2,57	1,37	1,47
Pior caso do núm. de erros	46	90	14	56
Erro máximo (dB)	2,46	2,03	0,92	1,64
Erro mínimo (dB)	0,10	0,14	0,03	0,04

Comparando os resultados obtidos com a busca através do erro quadrático e a busca através da medida da POTÊNCIA, podemos observar que o erro quadrático apresenta um melhor resultado para todos os "codebooks", mesmo para o CBK\_P, projetado com a medida da POTÊNCIA. Este resultado não está de acordo com [194]. Realizamos uma avaliação minuciosa desta questão através de gráficos dos espectros de potência quantizado e não-quantizado. Observamos que no caso da medida da POTÊNCIA, o erro entre os espectros concentrava-se nas altas frequências. Isto era esperado uma vez que as componentes do vetor de ponderação referentes à penúltima e à última LSF são desenfáticas por um fator de 0,64 e 0,16, respectivamente [194]. Podemos observar da Tabela 5.3 que o peso associado à última LSF limitou-se no máximo a 0,224 enquanto todos os outros pesos alcançaram valores maiores do que 0,70. Por outro lado, a Tabela 5.5 mostra que a sensibilidade do espectro de potência à última LSF é significativa, indicando que a concentração do erro nas últimas LSFs acarretará um erro não desprezível entre os espectros de potências nas altas frequências. Como a expressão da distorção espectral não faz distinção entre diferentes frequências do espectro, o erro nas altas frequências tem o mesmo peso do erro nas baixas frequências no cômputo da SD.

Resolvemos refazer as simulações respectivas à medida da POTÊNCIA, desconsiderando a desenfáticação das duas últimas LSFs. Por conveniência, chamaremos esta medida de distorção de POTÊNCIA-2. Os autores não especificam claramente, mas esta mesma modificação parece ter sido adotada em [213]. Os resultados obtidos são mostrados na Tabela 5.11.

Tabela 5.11 - Busca utilizando a medida de distorção da POTÊNCIA-2

	CBK EQ (10 bits)	CBK P (10 bits)	CBK D (10 bits)
SD mínima (dB)	0,58	0,60	0,65
SD máxima (dB)	5,68	5,80	5,97
SD média (dB)	2,35	2,38	2,34
SD < 2 dB	31,06 %	30,41 %	31,78 %
SD entre 2 dB e 4 dB	67,83 %	67,76 %	66,96 %
SD > 4 dB	1,11 %	1,82 %	1,26 %
Acertos	58,95 %	62,17 %	60,86 %
Média do núm. de erros	2,56	2,21	2,43
Pior caso do núm. de erros	150	90	118
Erro máximo (dB)	1,57	1,41	1,83
Erro médio (dB)	0,12	0,09	0,11

Podemos observar que a medida da POTÊNCIA-2 conduz a uma distorção menor do que a medida proposta originalmente. Acreditamos que este resultado é coerente. O motivo principal de termos aprofundado esta questão é a possibilidade de discutir a relação entre a SD e a qualidade subjetiva da quantização.

Logicamente, Atal e Paliwal [194] propuseram a desenfaturização das duas últimas LSFs devido às características do sistema auditivo humano que é mais sensível às baixas frequências. Este procedimento possui um caráter subjetivo que não é refletido pela SD. Podemos inferir deste caso que a SD não deve ser vista como uma medida inquestionável da qualidade subjetiva da quantização. O grande mérito da SD é fornecer um indicador relativamente confiável desta qualidade subjetiva. O problema específico da atribuição de menor importância às altas frequências já está sendo questionado por alguns pesquisadores [190]. Uma alternativa é computar a SD apenas em uma faixa do espectro como, por exemplo, de 0 a 3000 Hz [190] ou de 125 a 3450 Hz [210].

Na segunda etapa da avaliação do efeito da ponderação, implementamos o particionamento seguindo as sugestões fornecidas por [146] e [194]. Particionamos o vetor das LSFs em dois vetores, um com as primeiras quatro LSFs e o outro com as seis restantes. Para cada um destes dois vetores, projetamos dois "codebooks" independentes de 4096 "codewords" através do algoritmo LBG. Adotando esta estrutura levamos em conta que as LSFs mais significativas devem ser quantizadas com maior precisão [194]. Para projetarmos o par de "codebooks" optamos por utilizar o erro quadrático como medida de distorção. Chamaremos este "codebook" de CBK\_EQ\_SPLIT.

Como nosso interesse é a quantização vetorial do filtro LPC no codificador CELP, gostaríamos de poder realizar as mesmas simulações da primeira etapa utilizando 24 bits para quantizar os parâmetros do filtro. Neste caso, o problema é a determinação da melhor "codeword" em termos de

SD. Precisariamos comparar o espectro de cada filtro a ser quantizado com aproximadamente 16 milhões de espectros, correspondentes a cada uma das combinações entre "codewords" dos dois "codebooks" (4096 "codewords" para cada "codebook"). Nossas estimativas são de que precisaríamos de semanas de uso exclusivo de um computador do porte do Convex 210 existente na UFSC. Esta complexidade excessiva foi o motivo de atacarmos o problema nas duas etapas já mencionadas. Os resultados de uma só dessas etapas apresentaria fragilidade, mas as duas etapas se complementam caso haja coincidência de resultados.

Nesta segunda etapa adotamos uma solução aproximada para estimar qual seria o melhor par de "codewords" (uma vez que utilizamos nesta segunda etapa o particionamento em dois "codebooks" proposto em [194]). Fizemos uma pré-seleção de aproximadamente 400 pares de "codewords" e dentre estes pares determinamos qual o par que conduz à menor SD. Esta pré-seleção é feita de forma independente para cada "codebook", selecionando-se 20 "codewords" de cada um. Para esta pré-seleção testamos o erro quadrático e a medida da DIFERENÇA. Estes resultados são mostrados na Tabela 5.12.

Nesta e nas outras simulações envolvendo a busca de "codewords" particionadas, o ordenamento das LSFs é testado ainda na etapa da busca, de forma a garantir a estabilidade do filtro quantizado. O teste do ordenamento não é realizado na pré-seleção dos dois grupos de 20 "codewords" pois esta pré-seleção é feita de forma independente para cada "codebook". Apenas ao realizarmos as combinações possíveis entre os dois grupos de 20 "codewords", descartamos os pares que não satisfaçam o teste de ordenamento.

Tabela 5.12 - Desempenho da QV por particionamento para CBK\_EQ\_SPLIT (24 bits) para a codificação através da SD

Método de pré-seleção	Erro quadrático	DIFERENÇA
SD mínima (dB)	0,29	0,29
SD máxima (dB)	3,58	3,07
SD média (dB)	1,17	1,17
SD < 2 dB	97,47 %	97,68 %
SD entre 2 dB e 4 dB	2,52 %	2,32 %
SD > 4 dB	0,00 %	0,00 %
Entropia do "codebook" 1	11,12 bits	11,02
Entropia do "codebook" 2	11,13 bits	11,03

Utilizando o mesmo "codebook" CBK\_EQ\_SPLIT, quantizamos a seqüência de teste empregando para a busca as distorções do erro quadrático, da POTÊNCIA, da DIFERENÇA, da SENSIBILIDADE e da POTÊNCIA-2. Os resultados são mostrados na Tabela 5.13.

Tabela 5.13 - Busca usando diferentes medidas de distorção para o CBK EQ SPLIT (24 bits)

	Erro quadr.	POTÊN.	POTÊN. 2	SENSI.	DIFER.
SD mínima (dB)	0,29	0,29	0,29	0,29	0,29
SD máxima (dB)	4,22	4,65	3,58	3,59	4,22
SD média (dB)	1,24	1,34	1,21	1,19	1,20
SD < 2 dB	94,56 %	89,46 %	96,06 %	96,46 %	96,16 %
SD entre 2 e 4 dB	5,42 %	10,49 %	3,94 %	3,54 %	3,82 %
SD > 4 dB	0,01 %	0,05 %	0,00 %	0,00 %	0,01 %
Acertos ("codebook 1")	63,29 %	81,36 %	81,36 %	77,84 %	78,85 %
Média do núm. de erros (1)	2,03	1,31	1,31	1,36	1,37
Pior caso do núm. de erros (1)	20	19	19	11	22
Acertos ("codebook" 2)	70,89 %	52,51 %	76,31 %	81,66 %	79,91 %
Média do núm. de erros (2)	1,67	2,37	1,45	1,29	1,35
Pior caso do núm. de erros (2)	20	59	27	12	17

Podemos observar que a Tabela 5.13 apresenta resultados coerentes com os obtidos na primeira etapa. A desenfatização das duas últimas LSFs continua provocando uma piora no desempenho da busca.

Nesta seção apresentaremos também os resultados obtidos com o particionamento proposto em [146] e [194] para diferentes taxas de bits. Realizamos o particionamento da mesma forma que Atal e Paliwal [146], procurando repartir o número total de bits para a codificação do filtro LPC de forma igualitária entre os dois "codebooks". Por exemplo, para a QV por particionamento do filtro LPC usando 22 bits, projetamos um "codebook" de 2048 "codewords" (11 bits) para as primeiras quatro LSFs, e outro "codebook" de 2048 "codewords" para as seis LSFs menos significativas. Quando o número total de bits for ímpar, o bit excedente é associado ao primeiro "codebook". Todos os "codebooks" foram projetados utilizando o erro quadrático como medida de distorção. O par de "codebooks" respectivos à quantização do filtro LPC usando 24 bits é o mesmo par denominado CBK\_EQ\_SPLIT.

Na Tabela 5.14 apresentamos os resultados obtidos para a codificação da seqüência de teste utilizando o erro quadrático na busca da melhor "codeword". Nas tabelas 5.15 e 5.16 apresentamos os resultados da utilização das medidas da POTÊNCIA e da DIFERENÇA, respectivamente.

Tabela 5.14 - SD para busca utilizando o erro quadrático.

No. de bits	22	23	24	25	26
Média (dB)	1,38	1,36	1,24	1,22	1,12
<2 dB (%)	90,36 %	90,74 %	94,56 %	94,66 %	96,64 %
entre 2 e 4	9,72 %	9,25 %	5,42 %	5,32 %	3,35 %
<4 dB (%)	0,01 %	0,01 %	0,01 %	0,01 %	0,01 %

Tabela 5.15 - SD para busca utilizando a medida da POTÊNCIA.

No. de bits	22	23	24	25	26
Média (dB)	1,48	1,46	1,34	1,32	1,22
<2 dB (%)	83,97 %	84,67 %	89,46 %	89,82 %	92,70 %
entre 2 e 4	15,96 %	15,27 %	10,49 %	10,12 %	7,29 %
<4 dB (%)	0,06 %	0,05 %	0,05 %	0,05 %	0,01 %

Tabela 5.16 - SD para busca utilizando a medida da DIFERENÇA.

No. de bits	22	23	24	25	26
Média (dB)	1,34	1,32	1,20	1,18	1,08
<2 dB (%)	92,17 %	92,62 %	96,16 %	96,31 %	97,50 %
entre 2 e 4	7,82 %	7,38 %	3,82 %	3,67 %	2,50 %
<4 dB (%)	0,01 %	0,01 %	0,01 %	0,00 %	0,00 %

Comparando as tabelas 5.14 e 5.16, constatamos que a medida da DIFERENÇA permite alcançar um desempenho significativamente melhor do que o erro quadrático em termos de SD. Esta melhoria corresponde a uma diminuição de 1 a 2 bits na taxa, para um mesmo desempenho em termos de SD. Além da melhoria em termos de distorção, já mencionamos que a medida da DIFERENÇA é bastante atrativa por sua simplicidade computacional. Estes dois fatores foram os motivos pelos quais adotamos a medida da DIFERENÇA na etapa de projeto e busca da árvore K-dimensional.

## 5.6- CONCLUSÕES

Levando-se em conta seu bom desempenho em termos de distorção e sua reduzida complexidade computacional, a medida da DIFERENÇA mostrou-se superior em relação às demais. Nas frequências em que a diferença entre LSFs é menor, a sensibilidade do espectro de potência é maior [69]. Além disso, nas regiões de formantes existem no mínimo duas LSFs próximas [53].

Assim, a medida da DIFERENÇA compartilha características tanto com a medida da SENSIBILIDADE como com a medida da POTÊNCIA.

A medida da SENSIBILIDADE apresenta uma formulação puramente matemática. O bom desempenho apresentado por esta medida é coerente com sua motivação matemática. Por outro lado, podemos observar que a medida da POTÊNCIA obteve o pior desempenho. Todavia, devemos considerar que a medida da POTÊNCIA possui uma formulação de caráter subjetivo. Isso indica que os resultados obtidos utilizando-se a medida da POTÊNCIA não são necessariamente inferiores subjetivamente. Além disso, é mencionado em [194] que os fatores que multiplicam as duas últimas LSFs foram determinados empiricamente. Podemos considerar que, eventualmente, estes fatores não são os mais adequados para as condições em que foram feitas nossas simulações. Talvez um ajuste dos fatores pudesse levar à diminuição da distorção média comparada com a distorção correspondente ao erro quadrático, da mesma forma observada por Atal e Paliwal em [194].

Atualmente, há quase unanimidade na utilização do conceito de quantização transparente baseado na SD para a avaliação em sistemas de quantização de LSFs [173], [194], [210], [214]. É incontestável a importância do conceito de quantização transparente. Entretanto, devemos observar que, como toda medida objetiva, este conceito não pode ser adotado de forma inquestionável. De fato, a SD é apenas uma das medidas objetivas já publicadas na literatura [73], [83]. Existem diversas outras medidas que podem retratar tão bem, ou até melhor que a SD, o resultado de uma avaliação subjetiva [83].

Neste trabalho, não tivemos a pretensão de formalizar um estudo sobre a utilização da SD como um indicativo da avaliação subjetiva da QV das LSFs. Todavia, os resultados indicam claramente que a SD não incorpora alguns fatores subjetivos importantes como a maior importância das baixas frequências. Parece-nos natural que os pesquisadores busquem medidas objetivas alternativas ou o aperfeiçoamento do conceito de transparência para a avaliação da quantização do filtro LPC [190].

Apesar do questionamento acerca da SD, mantivemos neste trabalho a utilização do conceito de quantização transparente. Devido ao bom desempenho em termos de SD e à baixa complexidade computacional, a medida de distorção que será utilizada nos quantizadores que serão descritos no próximo capítulo é a medida da DIFERENÇA.

## CAPÍTULO 6

### CODIFICAÇÃO CELP USANDO ÁRVORE K-d PARA A QV DAS LSFs

#### 6.1- INTRODUÇÃO

O problema da aplicação de QV à codificação dos parâmetros do filtro LPC no CELP é a dimensão necessária para o "codebook". Seriam necessários "codebooks" com aproximadamente 500.000 configurações de filtro para satisfazer à exigência do CELP no tocante à resolução da quantização [190]. O armazenamento de um "codebook" de 19 bits ( $2^{19} = 524.288$  "codewords"), com "codewords" de dimensão 10 (ordem do filtro LPC), exigiria cerca de 10 Mbytes de memória (para 16 bits/amostra). Além disso, o projeto deste "codebook" consumiria um tempo considerável. Seriam necessárias dezenas de horas de voz digitalizada apenas para compor uma seqüência de treinamento estatisticamente válida. Por este motivo, a utilização de uma estrutura restritiva (multi-estágio, com particionamento, etc...) para o quantizador torna-se obrigatória.

Para uma dada distorção, a QV inerentemente conduz à redução na taxa de bits e ao aumento da complexidade do quantizador, quando comparada com a quantização escalar. Muitas estruturas já foram desenvolvidas para os quantizadores vetoriais de LSFs visando à redução da complexidade [155], [156], [157], [173], [174], [183], [190], [194], [210], [211], [212], [213], [214], [215], [216]. Todavia, a redução da complexidade através da aplicação de técnicas de busca rápida não se encontra devidamente explorada.

No Capítulo 4 fizemos uma breve revisão dos métodos de busca rápida. A dificuldade da utilização de alguns dos métodos de busca rápida no problema específico da QV das LSFs é a incorporação da medida dinâmica de distorção. Alguns dos mais eficientes métodos de busca rápida baseiam-se na desigualdade triangular e seus princípios perdem a validade quando a medida de distorção é dinâmica. A vantagem da árvore K-dimensional neste caso é a possibilidade de incorporação da medida dinâmica de distorção em seu projeto.

A busca em árvore K-dimensional (K-d) não exige qualquer estrutura especial do "codebook". Utilizaremos a estrutura com particionamento proposta por Atal e Paliwal [194]. A QV das LSFs utilizando a estrutura com particionamento ("split") [146] apresenta um bom compromisso entre complexidade e desempenho [173]. A busca exaustiva da melhor "codeword" nesta estrutura nos fornecerá uma referência em termos de desempenho. Nosso objetivo é a manutenção deste desempenho realizando-se a busca em um pequeno número de "codewords".

A motivação principal deste trabalho é avaliar o desempenho da busca rápida no caso específico da QV das LSFs, onde são utilizadas medidas dinâmicas de distorção. O quantizador vetorial que estaremos utilizando baseia-se na estrutura de particionamento proposta em [146]. Ressaltamos que existem outras estruturas que permitem obter o mesmo desempenho proporcionado pelo

particionamento proposto em [146] com menor complexidade computacional (ex: [173] e [183]). Contudo, muitas destas estruturas podem incorporar a busca em árvore K-d. Assim, os resultados obtidos utilizando-se a estrutura com particionamento [146] podem ser transportados para outras estruturas, guardando-se as devidas proporções.

Organizamos este capítulo da forma que se segue. Na seção 6.2 realizaremos uma breve avaliação dos métodos para análise LPC. Na seção 6.3 apresentaremos os resultados obtidos com o particionamento. Estes resultados servirão de referência para a comparação com os resultados obtidos utilizando-se busca em árvore K-dimensional. Estes últimos são mostrados na seção 6.4. Na seção 6.5 avaliaremos a aplicação da técnica de busca rápida integrando-a ao codificador CELP. As conclusões deste capítulo serão apresentadas na seção 6.6.

## 6.2- AVALIAÇÃO DE MÉTODOS PARA ANÁLISE LPC

Realizamos testes com quantização escalar no intuito de determinarmos o método a ser utilizado para a análise LPC. Dentre os métodos existentes, testamos três possibilidades: o da autocorrelação sem pré-ênfase, o da autocorrelação com pré-ênfase e o da covariância estabilizado [176].

O coeficiente de pré-ênfase utilizado foi de 0,9. Maiores detalhes sobre a pré-ênfase, também chamada de compensação das altas frequências [194], podem ser encontrados em [21], [148] e [190].

Para o método da autocorrelação, utilizamos uma janela de Hamming de 30 ms. Não realizamos "janelamento" do sinal de voz para o método da covariância.

Com exceção de uma única modificação, utilizamos o método da covariância desenvolvido em [176]. A modificação que fizemos refere-se ao tratamento da ocasional instabilidade do filtro LPC calculado pelo método da covariância. Em [176], a estabilização do filtro é alcançada com a limitação dos valores dos PARCORs à faixa de -1 a 1. Após termos realizado alguns testes, adotamos a reflexão dos pólos externos ao círculo unitário para dentro do mesmo (pólo fora da circunferência de raio unitário substituído por seu recíproco), porque este processo apresentou melhores resultados. Para o método da covariância não empregamos a ponderação do erro [32] utilizado por Atal e Paliwal [194], nem alteramos a diagonal principal da matriz de covariância [217].

Uma vez calculado o filtro LPC, realizamos a expansão de banda antes do processo de determinação das LSFs, da mesma forma que no CELP do DoD [181].

Testamos os três métodos de análise LPC usando a quantização escalar das LSFs adotada no CELP do DoD [181] que emprega 34 bits para cada filtro LPC. O CELP do DoD realiza a quantização uniforme de cada LSF. A configuração da alocação dos bits para cada LSF é : {3,4,4,4,4,3,3,3,3,3}. Os resultados obtidos são mostrados na Tabela 6.1.

Tabela 6.1 - Comparação entre métodos de análise LPC para quantização escalar

	Autocorrelação sem pré-ênfase	Autocorrelação com pré-ênfase	Covariância (com pré-ênfase)
SD mínima (dB)	0,33	0,36	0,47
SD máxima (dB)	3,14	5,02	4,86
SD média (dB)	1,35	1,45	1,46
SD < 2 dB (%)	94,71	92,60	91,64
SD entre 2 dB e 4 dB (%)	5,29	7,29	8,26
SD > 4 dB (%)	0,0	0,11	0,10

A Tabela 6.1 mostra uma superioridade do método da autocorrelação sem pré-ênfase. Realizamos alguns testes utilizando QV com o objetivo de convalidar a Tabela 6.1. Projetamos "codebooks" diferentes para cada um dos três métodos. Estes testes, apesar de não terem sido exaustivos, também foram favoráveis à autocorrelação sem pré-ênfase.

Além da vantagem em termos de SD, o método da autocorrelação garante a estabilidade do filtro e apresenta menor complexidade computacional [176]. Por este motivo adotamos o método da autocorrelação. Devido aos resultados obtidos, não realizamos a pré-ênfase do sinal de voz.

### 6.3- RESULTADOS DE REFERÊNCIA

Buscando obter resultados de referência, avaliamos a quantização escalar da razão log-área dos coeficientes PARCORs utilizada no CELP desenvolvido por Fernandes [176], [188]. Os resultados desta avaliação, em conjunto com os apresentados na Tabela 6.1, servirão como medida de comparação em termos de distorção e taxa de bits com os resultados obtidos com a QV. Avaliamos as configurações ilustradas em [176] (pp.75) e mostramos os resultados obtidos na Tabela 6.2. Em [176], foi mencionado que os testes subjetivos de comparação entre os quatro quantizadores descritos na Tabela 6.2 não indicaram diferenças significativas, motivando a adoção da configuração com o menor número de bits (32 bits).

Tabela 6.2 - Quantização escalar da razão log-área dos coeficientes PARCOR

	32 bits	34 bits	36 bits	40 bits
SD mínima (dB)	0,27	0,36	0,30	0,23
SD máxima (dB)	5,36	5,28	5,24	5,66
SD média (dB)	1,54	1,34	1,04	0,77
SD < 2 dB	89,15 %	97,06 %	98,94 %	99,65 %
SD entre 2 dB e 4 dB	10,79 %	2,9 %	1,02 %	0,31 %
SD > 4 dB	0,06 %	0,04 %	0,04 %	0,04 %

#### 6.4- QV DAS LSFs USANDO BUSCA RÁPIDA EM ÁRVORE K-d

Como dissemos na introdução deste capítulo, o quantizador vetorial que nos servirá de referência é o baseado no particionamento em dois "codebooks" de 4096 "codewords" (total de 24 bits). Utilizaremos o par de "codebooks" denominado CBK\_EQ\_SPLIT. Os resultados obtidos para este quantizador utilizando-se a busca exaustiva são mostrados na Tabela 5.15.

Para estimar os limites das projeções das partições de Voronoi usamos o método de Monte Carlo. Codificamos os 72.000 vetores da seqüência de treinamento baseando-nos na medida da DIFERENÇA.

Na Tabela 6.3 apresentamos os resultados obtidos com a utilização da busca conservativa em árvore. Para o projeto desta árvore foi utilizado o algoritmo EOC que se baseia na minimização do valor esperado da busca. Em cada caso, indicamos o número médio ('Média') de "codewords" correspondentes a cada nível de profundidade da árvore. De acordo com o explicado no Capítulo 4, este número nos permite calcular a quantidade de memória necessária para o armazenamento da árvore. Fornecemos também o número máximo de "codewords" ('Máximo'), ou seja, o número de "codewords" da folha mais populosa (caso a árvore terminasse no referido nível). São mostrados também os resultados obtidos em termos de SD para a codificação dos 8000 vetores da seqüência de teste. Finalmente, indicamos estimativas para a complexidade computacional seguindo a Eq. (5.1) e a Tabela 5.1.

Em aplicações em tempo real, o processador utilizado dever ser capaz de atender ao pior caso. O projetista pode basear-se em estatísticas para dimensionar a necessidade de processamento do algoritmo. Baseando-nos em estatísticas, podemos contabilizar o ganho advindo de técnicas como o cálculo da distância parcial [40] e o ordenamento do "codebook" [96]. Todavia, para definirmos um limite máximo do número de operações necessárias, não consideraremos neste trabalho o ganho computacional correspondente aos usos de distâncias parciais e do ordenamento do "codebook". Portanto, o número de operações (O) que aparece na Tabela 6.3 é calculado a partir do número máximo de "codewords" em um determinado nível da árvore. A incorporação das duas técnicas citadas [40], [96] permite-nos reduzir em mais da metade esta complexidade computacional [191].

Tabela 6.3 - Desempenho da árvore projetada pelo algoritmo EOC  
(número de "codewords" por nível da árvore)

Nível	0	1	2	3	4	5	6	7	8
Média ("cod." 1)	4096	2189,0	1211,5	677,0	382,81	222,91	132,95	80,06	49,27
Máximo ("cod." 1)	4096	2430	1472	895	506	281	182	120	73
Média ("cod." 2)	4096	2345,0	1403,0	871,75	551,81	356,34	233,81	155,41	104,31
Máximo ("cod." 2)	4096	2402	1472	933	623	426	284	209	153
Mem. Estát. (ME)	40960	40962	51463	53392	56004	59682	64810	71862	81806
Mem. Dinâ. (MD)	25	25	25	25	25	25	25	25	25
Operações (O)	245760	144803	88342	55101	34616	22135	14658	10481	7348
Complexidade (C)	247813	146856	90918	57776	37421	25124	17903	14079	11443

Em [171], os autores propõe os algoritmos EOC e GOC para o projeto de árvores K-d. Neste trabalho justifica-se teoricamente o fato destes dois algoritmos superarem o método baseado no critério do máximo produto, anteriormente utilizado para o projeto de árvores K-d [52], [67]. A superioridade dos algoritmos EOC e GOC em relação ao critério do máximo produto foi comprovada em nossas simulações. Entretanto, os autores mencionam que o algoritmo EOC atingiu melhor desempenho no que se refere ao pior caso, ou seja, ao número máximo de "codewords" em uma folha para um dado nível da árvore. Isto não foi confirmado em nossas simulações, como será mostrado adiante. Ressaltamos que as simulações realizadas em [171] foram baseadas na QV do próprio sinal de voz, utilizando-se "codewords" de dimensão  $K = 8$ .

Testamos o algoritmo GOC a partir das mesmas estimativas para os limites das projeções das partições utilizadas para o EOC. Os resultados encontram-se na Tabela 6.4.

Tabela 6.4 - Desempenho da árvore projetada pelo algoritmo GOC  
(número de "codewords" por nível de profundidade da árvore)

Nível	0	1	2	3	4	5	6	7	8
Média ("cod." 1)	4096	2189,5	1210,2	679,12	389,31	229,28	137,22	84,16	53,05
Máximo ("cod." 1)	4096	2193	1224	699	404	247	152	96	65
Média ("cod." 2)	4096	2348,5	1402,0	878,37	560,94	366,81	242,92	163,70	112,22
Máximo ("cod." 2)	4096	2349	1409	926	606	412	284	206	150
Mem. Estát. (ME)	40960	40962	51192	53082	55710	59491	64588	71982	82681
Mem. Dinâ. (MD)	25	25	25	25	25	25	25	25	25
Operações (O)	245760	137207	80122	50145	31556	20815	13938	9797	7048
Complexidade (C)	247813	139260	82687	52804	34346	23795	17172	13401	11187

Comparando os resultados obtidos com os algoritmos EOC e GOC, podemos observar que o algoritmo GOC conduziu a um resultado ligeiramente melhor em termos do pior caso, ou seja, do número máximo de "codewords" de uma folha para determinado nível da árvore. Isto significa um menor número de operações aritméticas. Por outro lado, o algoritmo EOC atinge um menor número médio de "codewords" para cada nível da árvore. Este fato reflete-se na memória necessária para o armazenamento da árvore, que é menor para o EOC. Todavia, no cômputo geral da medida de complexidade definida, o algoritmo GOC apresenta superioridade em relação ao EOC. Por este motivo, optamos pela utilização das árvore projetadas pelo algoritmo GOC. A Fig. 6.1 ilustra a substancial redução na complexidade computacional alcançada pela busca em árvore K-d. A Fig. 6.1 ilustra também a ligeira vantagem do algoritmo GOC.

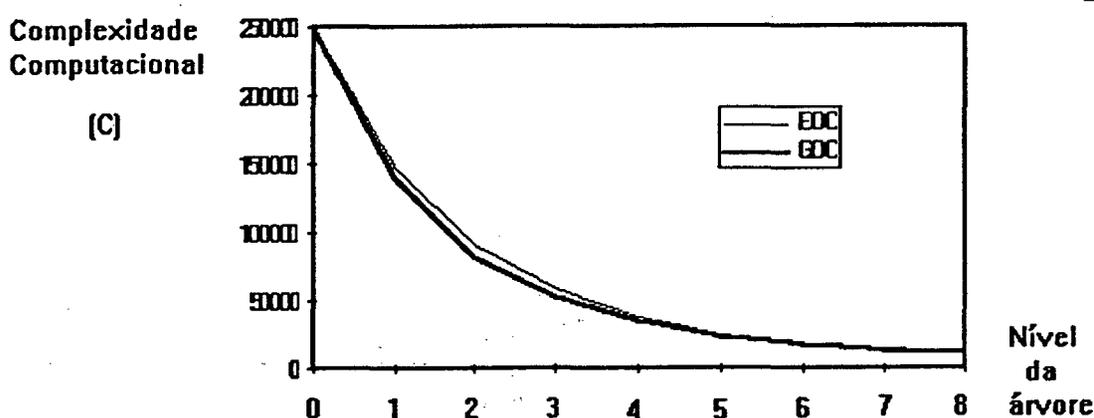


Fig. 6.1 - Comparação entre os algoritmos EOC e GOC

Resta-nos identificar se a busca em árvore K-d acarreta degradação significativa do desempenho do quantizador. Apresentamos na Tabela 6.5 os resultados da codificação dos 8.000 vetores da seqüência de teste para cada nível da árvore referente à Tabela 6.4. Da mesma forma que as duas tabelas anteriores, o nível 0 corresponde à própria busca exaustiva.

Tabela 6.5 - Avaliação da busca em árvore em termos de SD

Nível	0	1	2	3	4	5	6	7	8
SD média (dB)	1,20	1,21	1,21	1,21	1,21	1,21	1,21	1,22	1,22
SD < 2 dB (%)	96,16	96,06	95,96	95,94	95,82	95,77	95,70	95,55	95,49
$2 \leq SD \leq 4$ dB (%)	3,82	3,92	4,02	4,05	4,12	4,17	4,26	4,41	4,47
SD > 4 dB (%)	0,01	0,01	0,01	0,01	0,05	0,05	0,04	0,04	0,04

Podemos observar que de uma forma geral, a degradação introduzida pela busca em árvore não é significativa. Uma observação importante refere-se ao nível 1 da árvore. Este nível não exige o

armazenamento dos índices das "codewords" que compõe cada folha. Uma vez que se tenha determinado quais as "codewords" que pertencem a ambas as folhas, podemos organizar o "codebook" da maneira descrita na Fig. 6.2. As "codewords" relativas à primeira folha se estenderiam até um determinado endereço de memória. Outro endereço de memória indicaria a primeira "codeword" da segunda folha. As "codewords" que estivessem entre estes dois endereços pertenceriam às duas folhas e seriam pesquisadas em todas as buscas. As tabelas 6.3 e 6.4 indicam que esta estratégia permite alcançarmos redução substancial da complexidade sem que haja aumento na memória necessária (apenas os dois endereços precisam ser armazenados).

A estratégia descrita acima não se restringe à QV das LSFs, podendo ser utilizada em várias outras aplicações que envolvam QV. Além disso, podemos subdividir as "codewords" em um número maior de folhas. A única restrição neste caso é que uma "codeword" não pode pertencer a mais de duas "folhas". Uma alternativa relativamente simples seria fixar o valor de  $j$ , ou seja, escolher um único eixo que seria ortogonal a todos os hiperplanos divisores.

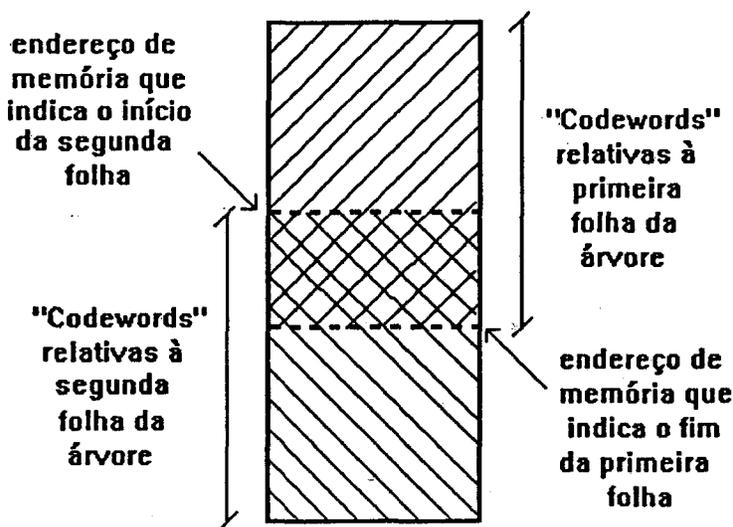


Fig. 6.2 - Organização do "codebook" para a utilização de um único nível da árvore.

### 6.5- AVALIAÇÃO DA VOZ SINTETIZADA

Nosso objetivo nesta etapa é avaliar se a QV utilizando busca rápida em árvore K-d introduz degradação perceptível no sinal de voz sintetizado pelo codificador CELP. Para este fim, utilizamos o CELP de referência desenvolvido em [176].

Para avaliarmos apenas os efeitos da quantização dos parâmetros do filtro LPC (filtro PCA), mantivemos a excitação quantizada a uma taxa de 4.600 bits/s, distribuídos da forma mostrada na Tabela 6.6. Os demais parâmetros do codificador são os mesmos apresentados na Tabela 2.1, com exceção do número de "codewords" do "codebook" fixo. Reduzimos o número de "codewords" do

"codebook" fixo para 128, no intuito de evitar que a excitação seja capaz de mascarar a quantização do filtro LPC.

Tabela 6.6 - Distribuição de bits para a excitação (amostragem a 8 kHz)

Ganho do PLA (b)	800 bits/s
Atraso do PLA (M)	1400 bits/s
Ganho do "codebook" gaussiano	1000 bits/s
Índice do "codebook" gaussiano	1400 bits/s

Existe uma dificuldade muito grande em exprimir a qualidade subjetiva da voz sintetizada por um codificador através de uma medida objetiva [14], [83]. Uma das maneiras de avaliarmos a voz sintetizada por um codificador é comparando as formas de onda do sinal sintetizado e do sinal original. A razão sinal / ruído segmentada média (RSRSEG) [176] baseia-se nesta comparação, sendo uma das medidas objetivas mais utilizadas em codificação de voz.

Os codificadores da forma de onda buscam justamente a semelhança entre as formas de onda do sinal sintetizado e do sinal original. Por isso, as medidas objetivas baseadas na semelhança entre as formas de onda, como a RSRSEG, são mais coerentes para os codificadores que baseiam-se neste princípio. Para os codificadores paramétricos e híbridos, estas medidas são bastante deficientes. Nestes casos, a RSRSEG deve ser avaliada com reservas pois não expressa necessariamente a qualidade subjetiva da voz sintetizada [176].

Buscando melhor aproximação com critérios subjetivos, utilizamos também neste trabalho, a razão sinal / ruído ponderada média  $\overline{RSR}_w$ . A  $RSR_w$  é a medida que os codificadores CELP buscam otimizar [208]. O valor da  $RSR_w$  para um quadro de análise é dado por [208]:

$$RSR_w = 10 \log_{10} \left( \frac{\sum_{n=1}^{160} s^2(n)}{\sum_{n=1}^{160} [(s(n) - \hat{s}(n))_w]^2} \right) \quad (6.1)$$

onde  $s(n)$  é o sinal de voz original,  $\hat{s}(n)$  é o sinal de voz sintetizado e  $(s(n) - \hat{s}(n))_w$  corresponde ao erro ponderado, ou seja, ao sinal na saída do filtro de ponderação  $W(z)$  (Eq. (2.2)).

Para efeito de comparação mostramos na Tabela 6.7 os resultados obtidos utilizando-se a quantização escalar do filtro LPC (filtro PCA). Nestas simulações foram utilizados dois locutores que não pertenciam à seqüência de treinamento dos "codebooks". Cada um destes locutores contribuiu com aproximadamente 11 segundos de voz. A elocução proferida é uma frase foneticamente equilibrada.

Tabela 6.7 - Resultados objetivos (RSRSEG e  $\overline{RSR_w}$  em dB) utilizando quantização escalar.

	LSFs - 34 bits (DoD)	PARCORs - 32 bits ([176])	PARCORs - 40 bits ([176])
Locutor masculino	14,30 e 17,78	14,12 e 17,76	14,62 e 18,17
Locutor feminino	13,10 e 16,08	13,39 e 16,20	13,54 e 16,43

Na Tabela 6.8 apresentamos os resultados obtidos para a o CELP de referência, incorporando a busca rápida em árvore K-d utilizando a medida da DIFERENÇA. A árvore utilizada foi a correspondente às Tabelas 6.4 e 6.5. O nível 0 corresponde à busca exaustiva.

Tabela 6.8 - Resultados objetivos (RSRSEG e  $\overline{RSR_w}$  em dB) utilizando QV c/ busca em árvore K-d

Nível	0	1	2	3	4	5	6	7	8
Loc. masculino (RSRSEG)	14,51	14,57	14,60	14,63	14,56	14,61	14,67	14,64	14,62
Loc. masculino ( $\overline{RSR_w}$ )	17,90	17,96	17,94	17,97	17,90	17,92	17,98	17,92	17,90
Loc. feminino (RSRSEG)	13,50	13,48	13,51	13,43	13,59	13,60	13,50	13,50	13,59
Loc. feminino ( $\overline{RSR_w}$ )	16,30	16,27	16,29	16,21	16,32	16,28	16,21	16,26	16,30

Realizamos testes informais para avaliar subjetivamente a qualidade da voz sintetizada. Os dois quantizadores escolhidos para a comparação foram o quantizador escalar usado no CELP do DoD (Tabela 6.7) com 34 bits para representar as LSFs e o quantizador vetorial de 24 bits usando busca rápida em árvore. A árvore utilizada foi a relativa ao nível 8 da Tabela 6.8. Comparando as vozes sintetizadas pelo CELP de referência incorporando estes quantizadores, não conseguimos observar diferença perceptível.

## 6.6- CONCLUSÕES

Neste capítulo apresentamos os resultados obtidos com a utilização de um método para redução da complexidade computacional da QV do filtro LPC, baseado em árvore K-dimensional. Os resultados indicaram que mesmo utilizando-se uma medida dinâmica de distorção, a árvore K-d não introduz degradação significativa.

A utilização da ponderação do erro através da medida da DIFERENÇA mostra-se importante. Assim, o projeto da árvore K-d deve levar em conta a ponderação do erro. A ponderação do erro torna inviável a utilização de métodos analíticos para a determinação dos vértices das partições de Voronoi, pois as partições de Voronoi passam a ser regiões não-convexas. Entretanto, o método de Monte Carlo pode incorporar a ponderação dinâmica sem dificuldades.

Uma observação importante refere-se ao algoritmo usado para o projeto da árvore. Quando a preocupação maior é o número máximo de "codewords" em uma folha para um determinado nível da árvore, o algoritmo GOC supera o algoritmo EOC. Além desta vantagem, o GOC exige um tempo significativamente menor para o projeto do que o EOC.

Apresentamos também uma estratégia que reduz a complexidade em quase 50 % a partir do uso de um único hiperplano divisor. Esta estratégia corresponde à árvore de um único nível e não requer o armazenamento dos índices das "codewords" relativos a cada folha da árvore.

## CAPÍTULO 7

### CONCLUSÕES

Inicialmente, realizamos neste trabalho uma revisão bibliográfica da codificação CELP. Acreditamos que a evolução cronológica seria a maneira mais conveniente de realizar esta exposição. Através desta revisão bibliográfica, apresentamos a motivação para as pesquisas com respeito à utilização da quantização vetorial na codificação do filtro LPC no CELP (filtro PCA). Utilizamos como exemplo o caso específico da telefonia móvel celular que vem exigindo a diminuição na taxa de bits dos codificadores de voz.

Devido à necessidade da diminuição da taxa de bits, nossa preocupação neste trabalho foi especificamente com a quantização vetorial do filtro LPC. Nossa proposta é a utilização de uma técnica de busca rápida baseada em árvore K-dimensional para a redução da complexidade computacional associada à quantização vetorial do filtro LPC.

Atualmente, existem muitos quantizadores vetoriais cujas estruturas permitem considerável redução na complexidade computacional, tanto em termos do número de operações quanto de memória necessária para armazenamento. Contudo, a utilização de métodos de busca rápida para a redução da complexidade da quantização das LSFs não se encontra suficientemente explorada.

Neste trabalho realizamos uma avaliação profunda das medidas dinâmicas comumente utilizadas em sistemas de quantização vetorial de LSFs. Procuramos estimar qual o ganho advindo da ponderação dinâmica do erro quadrático. Para realizar esta avaliação adotamos uma estratégia diferente da utilizada na literatura. Utilizamos a própria distorção espectral na etapa de codificação de forma a determinar a "codeword" que conduz à menor distorção espectral. A partir desta informação, calculamos estatísticas que permitem uma comparação mais precisa do que a obtida através da abordagem normalmente utilizada [194].

Devemos ressaltar que nossos resultados não apresentaram concordância com algumas considerações tecidas em [194]. Em nossa avaliação, a distorção espectral não expressa infalivelmente os fatores subjetivos nos quais se baseia a medida de distorção proposta em [194]. Assim, a desenfaturação das duas últimas LSFs sugerida em [194] ocasionou um aumento na distorção espectral. Ressaltamos que os fatores empíricos utilizados na medida da POTÊNCIA [194] foram obtidos a partir de arquivos de voz diferentes dos utilizados neste trabalho. Talvez, estes fatores pudessem ser ajustados de forma a conduzir à diminuição na distorção espectral mencionada por Atal e Paliwal [194].

Notadamente, a distorção espectral não incorpora características importantes do sistema auditivo humano como por exemplo, a maior sensibilidade às baixas frequências. Apesar do

questionamento em relação à medida objetiva utilizada, acreditamos que a ponderação dinâmica do erro é muito importante. Pelos resultados obtidos, podemos observar que o erro quadrático sem ponderação apresenta-se como uma medida bastante deficiente. Isto pode ser visto tanto nos resultados em termos de distorção espectral, quanto na comparação com a medida da POTÊNCIA [194], que é baseada em fatores subjetivos. Os resultados obtidos utilizando-se ponderação são significativamente diferentes dos obtidos sem a ponderação. A distorção espectral pode vir a ser substituída por outras medidas de distorção na avaliação final da quantização do filtro LPC e conseqüentemente propiciar o aparecimento de outras medidas a serem utilizadas na etapa de codificação. Todavia, nos sistemas de quantização das LSFs deve ser mantida a importância do caráter dinâmico das medidas usadas na etapa de codificação.

Sendo assim, os métodos de busca rápida devem incorporar as medidas dinâmicas de distorção para que possam ser aplicados convenientemente ao caso específico da quantização vetorial das LSFs. Buscamos avaliar a potencialidade do uso da árvore K-dimensional neste contexto. Outros métodos de busca rápida (os baseados na desigualdade triangular, por exemplo) apresentam restrições quando a medida de distorção empregada é dinâmica.

Inicialmente, definimos os conceitos de busca conservativa e busca simples usando árvore K-dimensional. Mostramos que o projeto da árvore, tendo em vista a busca conservativa, pode incorporar uma medida dinâmica de distorção através da utilização do método de Monte Carlo na estimativa dos limites das projeções das partições de Voronoi.

Os resultados obtidos indicam que, mesmo incorporando a ponderação dinâmica no processo de estimativa das projeções, a árvore K-dimensional permite alcançarmos uma redução considerável na complexidade computacional. A busca rápida em árvore pode ser utilizada em conjunto com diversos dos métodos de estruturação do "codebook" propostos, diminuindo ainda mais a complexidade computacional associada a estes métodos.

O objetivo principal deste trabalho reside na diminuição da complexidade computacional através da busca rápida. Assim, não nos preocupamos com os valores absolutos de desempenho - em termos de distorção espectral - dos quantizadores implementados. Nossa abordagem consistiu na comparação do desempenho obtido com busca rápida e o obtido com a busca exaustiva. Contudo, o conceito de quantização transparente baseia-se em valores absolutos, além de ser interessante a comparação do desempenho dos quantizadores implementados com os resultados divulgados na literatura. Assim, tentamos dentro do possível, reproduzir resultados apresentados em outros trabalhos. Todavia, esta tarefa apresenta sérios contratempos. Um exemplo, é a dependência do desempenho do quantizador com o método utilizado para a análise LPC.

Recentemente, foram publicados alguns resultados que permitem melhor entendimento da influência da análise LPC no processo de quantização do filtro LPC [217]. Sabe-se que no método da covariância, quando os ganhos de predição são elevados, pode haver o mau condicionamento da matriz de covariância [12]. Uma forma de contornar este problema é multiplicar os componentes da

diagonal principal da matriz de covariância por  $(1+\alpha)$ , onde  $\alpha$  é um número positivo pequeno [217]. Uma consequência deste procedimento é a diminuição do ganho de predição do filtro LPC. A avaliação da diminuição do ganho de predição no contexto da quantização transparente do filtro LPC, ainda não recebeu a atenção merecida nas publicações abordando a quantização do filtro LPC.

Podemos observar em [217], que diminuindo o ganho de predição através do aumento de  $\alpha$ , a variância das LSFs diminui. Conseqüentemente, diminui a distorção espectral associada à quantização do filtro LPC para uma dada taxa de bits. Isso porque as respostas em frequência dos filtros LPC passam a ser mais parecidas, facilitando a quantização. Contudo, a diminuição do ganho de predição do filtro LPC exige maior precisão na quantização da excitação dos codificadores de voz. Acreditamos que esta relação de compromisso não está sendo considerada de maneira adequada.

Durante o desenvolvimento deste trabalho, os quantizadores implementados utilizando a busca exaustiva e o método da autocorrelação na análise LPC, apresentaram resultados inferiores aos obtidos em [194] e [173]. Através de contato pelo correio eletrônico, Paksoy expôs que obteve resultados semelhantes quando usou o método da autocorrelação, passando a usar o método da covariância por sugestão de Paliwal. Paliwal fazia a suposição de que a pré-ênfase do sinal de voz seria um fator muito importante para o desempenho do método da covariância. Por esse motivo, realizamos as simulações apresentadas no Capítulo 6, buscando identificar a importância da pré-ênfase na análise LPC. Ressaltamos que o filtro de pré-ênfase utilizado é um simples filtro FIR de primeira ordem [148], existindo outras técnicas de filtragem que podem conduzir a melhores resultados [190]. Pudemos observar que a pré-ênfase realmente diminui a faixa dinâmica dos coeficientes do filtro LPC [21]. Todavia, isso não se reflete na diminuição da variância das LSFs. O ganho de predição não apresenta mudança significativa devida à pré-ênfase [21]. Assim, o fator principal da vantagem apresentada pelo método da covariância aparentemente decorre da diminuição do ganho de predição do filtro LPC através da alteração da diagonal principal da matriz de covariância. Essa técnica não foi incorporada ao método da covariância nas simulações realizadas.

Parece-nos pouco conveniente comparar métodos de quantização do filtro LPC a partir de arquivos de LSFs diferentes, sem apresentar o ganho de predição associado. Sugerimos que os trabalhos referentes à quantização do filtro LPC apresentem além do desempenho em termos de distorção espectral, os valores do ganho de predição.

Como sugestão para trabalhos futuros, apresentamos a comparação da busca em árvore K-dimensional com outros métodos de busca rápida. Como a maioria destes métodos não permite a incorporação de forma direta das medidas dinâmicas de distorção, os mesmos poderiam ser utilizados para identificar, através do erro quadrático, não apenas uma "codeword", mas algumas delas, deixando para que a decisão final fosse feita a partir de outra medida de distorção.

Outro estudo que pode ser aprofundado refere-se aos dois aspectos do projeto da árvore: a estimação dos limites das projeções das partições de Voronoi e a determinação dos hiperplanos divisores. A estimativa proporcionada pelo método de Monte Carlo é otimista, no sentido de que os

limites reais das projeções, são sempre maiores do que os valores obtidos por Monte Carlo (considerando que as LSFs não assumem valores negativos). Os limites obtidos por Monte Carlo podem ser expandidos de forma que se aproximem do valor correto. Este processo tende a diminuir a eficiência da árvore em termos de complexidade computacional e melhorar o desempenho no tocante à distorção. O projeto dos hiperplanos divisores pode ser aprimorado utilizando-se algum método de decisão atrasada para a escolha do eixo  $j$ . Da forma como é comumente feita, a determinação do hiperplano divisor baseia-se apenas no nível da árvore em questão, não se considerando as decisões posteriores.

No contexto mais amplo de codificação de voz, temos duas sugestões a fazer. A primeira é o aprimoramento da voz sintetizada e, possivelmente, a redução da complexidade computacional, através da classificação do sinal de voz. Os resultados que estão sendo obtidos neste sentido são encorajadores [86], [117], [189], [195]. Acreditamos que, na busca de uma codificação mais eficiente, haverá uma crescente necessidade em extrair e identificar o maior número possível de características acústicas do sinal de voz. As pesquisas nesse sentido podem utilizar parte do conhecimento desenvolvido na área de reconhecimento de locução e locutor.

A outra proposta é o estudo do vocoder multi-bandas (IMBE) [164]. Este codificador possui um grande potencial para a obtenção de voz sintetizada com boa qualidade com baixas taxas de bits. Acreditamos que este codificador apresenta grande potencialidade de concorrer com o CELP na busca por boa qualidade de voz a taxas entre 3 e 4 kbps.

Como conclusão geral, gostaríamos de ressaltar a importância crescente da quantização vetorial nas mais diversas aplicações. Esta expansão faz crescer o interesse por métodos de busca rápida. O método de busca rápida baseado em árvore  $K$ -dimensional aqui abordado, pode ser utilizado em outros sistemas de quantização vetorial como, por exemplo, codificação de imagens e reconhecimento de voz. Mesmo no contexto do CELP, a aplicação de métodos de busca rápida pode transcender a quantização vetorial do filtro LPC. Em [218], obtém-se redução na complexidade computacional da busca da melhor "codeword" do "codebook" fixo através de um método de busca rápida baseado na pré-seleção de algumas "codewords".

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] PAPOULIS, Athanasios. Probability Random Variables and Stochastic Processes. New York, McGraw-Hill, 1965. 583p.
- [2] ATAL, Bishnu S. & SCHROEDER, Manfred R.. Adaptive Predictive Coding of Speech Signals. The Bell System Technical Journal, E.U.A., :1973-86, Out. 1970.
- [3] OPPENHEIM, A.. Speech Spectrograms Using the Fast Fourier Transform. IEEE Spectrum, E.U.A., 7:57-62, Ago. 1970.
- [4] ATAL, Bishnu S. & HANAUER, Suzanne. Speech Analysis and Synthesis by Linear Prediction of the Speech Wave. The Journal of the Acoustical Society of America, New Jersey, 50(2/2):637-55, Abr. 1971.
- [5] BERGER, Toby. Rate Distortion Theory. E.U.A., Prentice-Hall, 1971. 311p.
- [6] MAKHOUL, John. Linear Prediction: A Tutorial Review. Proceedings of the IEEE, E.U.A., 63(4):561-80, Abr. 1975.
- [7] BENTLEY, J.. Multidimensional Binary Search Trees Used for Associative Searching. Communications of the ACM. 18(9):509-17, Set. 1975.
- [8] ITAKURA, F. "Line Spectrum Representation of Linear Predictive Coefficients of Speech Signals", Journal of Acoustical Society of America, E.U.A., 57: 535(a), s35(A), 1975.
- [9] FRIEDMAN, J. & BASKETT, F. & SHUSTEK, L.. An Algorithm for Finding Nearest Neighbors. IEEE Transactions on Computers. 24:1000-6, Out. 1975.
- [10] YUNCK, THOMAS. A Technique to Identify Nearest Neighbors. IEEE Transactions on Systems, Man, and Cybernetics, 6(10):678-83, Out. 1976.
- [11] RABINER, Lawrence R. & SCHAFER, Ronald W.. Digital Processing of Speech Signals. New Jersey, Prentice-Hall, 1978. 512p.
- [12] ATAL, Bishnu S. & SCHROEDER, Manfred R.. Predictive Coding of Speech Signals and Subjective Error Criteria. IEEE Transactions on Acoustics, Speech, and Signal Processing, E.U.A., 27(3):247-54, Jun. 1979.
- [13] ROCHA, Luis F.. Predicción Lineal Aplicada a Senales de Voz. Revista Telegrafica Electronica, Argentina, 799:962-5/969, Set. 1979 (Parte 1), 800:1139-41/1148, Out 1979 (Parte 2).
- [14] GRAY, Robert M. & BUZO, Andrés & GRAY JR., Augustine H. & MATSUYAMA, Yasuo. Distortion Measures for Speech Processing. IEEE Transactions on Acoustics, Speech, and Signal Processing, E.U.A., 28(4):367-76, Ago. 1980.
- [15] LINDE, Yoseph & BUZO, Andrés & GRAY, Robert M.. An Algorithm for Vector Quantizer Design. IEEE Transactions on Communications, E.U.A., 28(1):84-95, Jan. 1980.
- [16] BUZO, A.; GRAY, A. & MARKEL, J.. Speech Coding Based Upon Vector Quantization. IEEE Transactions on Acoustics, Speech and Signal Processing, E.U.A., 28:562-74, 1980.
- [17] BOWYER, A.. Computing Dirichlet Tessellations. The Computer Journal. 24(2):162-6. 1981.

- [18] WATSON, D. F.. Computing the n-dimensional Delaunay tessellation with Application to Voronoi Polytopes. The Computer Journal. 24(2):167-71. 1981.
- [19] SETHI, I.. A Fast Algorithm for Recognizing Nearest Neighbors. IEEE Transactions on Systems, Man, and Cybernetics. 11(3):245-8, Mar. 1981.
- [20] SCHROEDER, Manfred R. & ATAL, Bishnu S.. Speech Coding Using Efficient Block Codes. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Paris, 1982. Anais... Paris, IEEE, 1982. p. 1668-71.
- [21] ATAL, Bishnu S.. Predictive Coding of Speech at Low Bit Rates, IEEE Transactions on Communications, E.U.A., 30:600-14, Abr. 1982.
- [22] ABUT, H.; GRAY, R. & REBOLLEDO, G.. Vector Quantization of Speech and Speech-Like Waveforms. IEEE Transactions on Acoustics, Speech and Signal Processing, E.U.A., 30(3):423-35, Mar. 1982.
- [23] GERSHO, A.. On The Structure of Vectors Quantizers. IEEE Transactions on Information Theory, E.U.A., 28(2):157-66. Mar. 1982.
- [24] JUANG, B. & GRAY, D. & GRAY, A.. Distortion Performance of VQ for LPC voice Coding. IEEE Transactions on Acoustics, Speech and Signal Processing, E.U.A., 30(2):294-303, Abr. 1982.
- [25] LLOYD, S.. Least Squares Quantization in PCM. IEEE Transactions on Information Theory, E.U.A., 28(2):129-37, Mar. 1982.
- [26] ROSENFELD, A. & KAK, A.. Digital Picture Processing. E.U.A., Academic Press, 1982. 435p.
- [27] GERSHO, A. & CUPERMAN, V.. Vector Quantization : A pattern-Matching Technique for Speech Coding. IEEE Communications Magazine, :15-21, Dez. 1983
- [28] BARROSO, L. et al. Cálculo Numérico. São Paulo. Harper & Row do Brasil. 1983. 283p.
- [29] GRAY, Robert M.. Vector Quantization. IEEE ASSP Magazine, E.U.A., :4-29, Abr. 1984.
- [30] SOONG, Frank K. & JUANG, Biing-Hwang. Line Spectrum Pair (LSP) and Speech Data Compression. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, San Diego, 1984. Anais... San Diego, IEEE, 1984. p. 1.10.1-1.10.4.
- [31] SABIN, M. & GRAY, R.. Product Code Vector Quantizers for Waveform and Voice Coding. IEEE Transactions on Acoustics, Speech and Signal Processing, E.U.A., 32,:474-88, 1984.
- [32] SINGHAL, S. & ATAL, B.. Improving Performance of Multi-Pulse LPC Coders at Low Bit Rates. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, San Diego, 1984. Anais... San Diego, IEEE, 1984. p. 1.3.1-1.3.4.
- [33] SCHROEDER, Manfred & ATAL, Bishnu S.. Code-Excited Linear Prediction (CELP): High-Quality Speech at Very Low Bit Rates. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tampa, 1985. Anais... Tampa, IEEE, 1985. p. 937-40.

- [34] OHYAMA, Guen. A Stochastic Model of Excitation Source for Linear Prediction Speech Analysis-Synthesis. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tampa, 1985. Anais... Tampa, IEEE, 1985. p. 941-4.
- [35] SCHROEDER, Manfred & ATAL, Bishnu S.. Stochastic Coding of speech Signals at Very Low Bit Rates: The Importance os Speech Perception. Speech Communication, E.U.A., 4(1-3):155-62, Ago. 1985.
- [36] MAKHOUL, John & ROUCOS, Salim & GISH, Herbert. Vector Quantization in Speech Coding. Proceedings of the IEEE, 73(11):1551-88, Nov. 1985.
- [37] KANG, George S. & FRANSEN, Lawrence J.. Application of Line-Spectrum Pairs to Low-Bit-Rate Speech Encoders. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tampa, 1985. Anais... Tampa, IEEE, 1985. p. 244-7.
- [38] OPPENHEIM, Alan V. & SCHAFER, Ronald W.. Digital Signal Processing. New Jersey, Prentice-Hall, 1985. 585p.
- [39] KOPEC, G.. The Signal Representation Language SRL. IEEE Transactions on Acoustics, Speech and Signal Processing, E.U.A., 33(4):921-32. Ago. 1985.
- [40] BEI, C. & GRAY, R.. An Improvement of the Minimum Distortion Encoding Algorithm for Vector Quantization. IEEE Transactions on Communications, E.U.A., 33(10):1132-3, Out. 1985.
- [41] OHYAMA, Ghen. A Novel Approach to Estimating Excitation Code in Code-Excited Linear Prediction Coding. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1986. Anais... Tokyo, IEEE, 1986. p. 3067-70.
- [42] ATAL, Bishnu S.. High-Quality Speech at Low Bit Rates: Multi-Pulse and Stochastically Excited Linear Predictive Coders. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1986. Anais... Tokyo, IEEE, 1986. p. 1681-684.
- [43] HERNÁNDEZ-GÓMEZ, L. A. & CASAJÚS-QUIRÓS, F. J. & FIGUEIRAS-VIDAL, A. R. & GARCHA-GÓMEZ, R.. Reducing Complexity on a Code-Excited Linear Predictor. In: Signal Processing III: Theories and Applications. Holanda, Elsevier Science Publishers B. V., 1986. p. 481-3.
- [44] TRANCOSO, Isabel M. & ALMEIDA, Luis B. & TRIBOLET, Jose M.. A Study on the Relationships Between Stochastic and Harmonic Coding. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1986. Anais... Tokyo, IEEE, 1986. p. 1709-12.
- [45] HERNÁNDEZ-GÓMEZ, L. A. & CASAJÚS-QUIRÓS, F. J. & FIGUEIRAS-VIDAL, A. R. & GARCHA-GÓMEZ, R.. On the Behavior of Reduced Complexity Code Excited Linear Prediction (CELP). In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1986. Anais... Tokyo, IEEE, 1986. p. 469-72.

- [46] DAVIDSON, Grant & GERSHO, Allen. Complexity Reduction Methods for Vector Excitation Coding. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1986. Anais... Tokyo, IEEE, 1986. p. 3055-8.
- [47] ROSE, Richard C. & BARNWELL III, Thomas P.. The Self-Excited Vocoder - An Alternate Approach to Toll Quality at 4800 bps. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1986. Anais... Tokyo, IEEE, 1986. p. 453-6.
- [48] TRANCOSO, Isabel M. & ATAL, Bishnu S.. Efficient Procedures for Finding the Optimum Innovation in Stochastic Coders. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1986. Anais... Tokyo, IEEE, 1986. p. 2375-8.
- [49] KROON, Peter & DEPRETTERE, Ed F. & SLUYTER, Rob J.. Regular-Pulse Excitation - A Novel Approach to Effective and Efficient Multipulse Coding of Speech. IEEE Transactions on Acoustics, Speech, and Signal Processing, E.U.A., 34(5):1054-63, Oct. 1986.
- [50] KROON, Peter & SLUYTER, Rob B. & DEPRETTERE, Ed. F.. A Low Complexity Regular Pulse Coding Scheme with a Reduced Transmission Delay. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1986. Anais... Tokyo, IEEE, 1986. p. 3083-6.
- [51] KABAL, P. & RAMACHANDRAN, P.. The Computation of Line Spectral Frequencies Using Chebyshev Polynomials. IEEE Transactions on Acoustics, Speech and Signal Processing, E.U.A., 34(6):1419-26. Dez. 1986.
- [52] CHENG, D. & GERSHO, A.. A Fast codebook search algorithm for nearest neighbor pattern matching. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1986. Anais... Tokyo, IEEE, 1986. p. 265-8.
- [53] SUGAMURA, N. & ITAKURA, F.. Speech Analysis and Synthesis methods developed at ECL in NTT - from LPC to LSP. Speech Communications, 5:199-215. Jun. 1986.
- [54] EDELSBRUNNER, Herbert & SEIDEL, Raimund. Voronoi Diagrams and Arrangements. In: Discrete and Computational Geometry. Springer-Verlag, New York, 1986. p. 25-44.
- [55] RAMAMURTHI, Bhaskar & GERSHO, Allen. Classified Vector Quantization of Images. IEEE Transactions on Communications, 34(11):1105-15. Nov. 1986.
- [56] ADOUL, J-P. & LAMBLIN, C.. A Comparison of Some Algebraic Structures for CELP Coding of Speech. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1987. Anais... Tokyo, IEEE, 1987. p. 1953-6.
- [57] EQUITZ, Wiliam. Fast Algorithms for Vector Quantization Picture Coding. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1987. Anais... Tokyo, IEEE, 1987. p. 725-8.

- [58] ATAL, Bishnu S.. Stochastic Gaussian Model for Low-Bit Rate Coding of LPC Area Parameters. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1987. Anais... Tokyo, IEEE, 1987. p. 2404-7.
- [59] KROON, Peter & ATAL, Bishnu S.. Quantization Procedures for the Excitation in CELP Coders. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1987. Anais... Tokyo, IEEE, 1987. p. 1649-52.
- [60] ADOUL, J-P. & MABILLEAU, P. & DELPRAT, M. & MORISSETE, S.. Fast CELP Coding Based on Algebraic Codes. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1987. Anais... Tokyo, IEEE, 1987. p. 1957-60.
- [61] ROSE, R. C. & BARNWELL III, Thomas P.. Quality Comparison of Low Complexity 4800 bps Self Excited and Code Excited Vocoders. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1987. Anais... Tokyo, IEEE, 1987. p. 1927-40.
- [62] LIN, Daniel. Speech Coding Using Efficient Pseudo-Stochastic Block Codes. INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1987. Anais... Tokyo, IEEE, 1987. p. 1354-57.
- [63] RAMACHANDRAN, Ravi P. & KABAL, Peter. Stability and Performance of Pitch Filters in Speech Coders. IEEE Transactions on Acoustics, Speech, and Signal Processing, E.U.A., 35(7):937-46, Jul. 1987.
- [64] DAVIDSON, G. & YONG, M. & GERSHO, A.. Real-Time Vector Excitation Coding of Speech At 4800 bps. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1987. Anais... Tokyo, IEEE, 1987. p. 2189-92.
- [65] LIPPMANN, R.. An Introduction to Computing with Neural Nets. IEEE ASSP Magazine. :4-22. Abr. 1987.
- [66] SOLEYMANI, M. & MORGERA, S.. An Efficient Nearest Neighbor Search Method. IEEE Transactions on Communications, 35(6):677-9. Jun. 1987.
- [67] LOWRY, A. & HOSSAIN, S. & MILLAR, W.. Binary Search Trees for Vector Quantization. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 1987. Anais... Tokyo, IEEE, 1987. p. 51.8.1-4.
- [68] KROON, Peter & DEPRETTERE, Ed F.. A Class of Analysis-by-Synthesis Predictive Coders for High Quality Speech Coding at Rates Between 4.8 and 16 kbits/s. IEEE Journal on Selected Areas in Communications, E.U.A., 6(2):353-63, Fev. 1988.
- [69] SOONG, Frank K. & JUANG, Biing-Hwang. Optimal Quantization of LSP Parameters. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New York, 1988. Anais... New York, IEEE, 1988. p. 394-7.

- [70] KABAL, P. & MONCET, J.-L. & CHU, C. C.. Synthesis Filter Optimization and Coding: Applications to CELP. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New York, 1988. Anais... New York, IEEE, 1988. p. 147-150.
- [71] KROON, Peter & ATAL, Bishnu S.. Strategies for Improving the Performance of CELP Coders at Low Bit Rates. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New York, 1988. Anais... New York, IEEE, 1988. p.151-4.
- [72] DAVIDSON, Grant & GERSHO, Allen. Multiple-Stage Vector Excitation Coding of Speech Waveforms. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New York, 1988. Anais... New York, IEEE, 1988. p. 163-6.
- [73] KITAWAKI, Nobuhiko & NAGABUCHI, Hiromi & ITOH, Kenzo. Objective Quality Evaluation for Low-Bit-Rate Speech Coding Systems. IEEE Journal on Selected areas in Communications, E.U.A., 6(2):242-8, Fev. 1988.
- [74] KLEIJN, W. B. & KRASINSKI, D. J. & KETCHUM, R. H.. Improved Speech Quality and Efficient Vector Quantization in SELP. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New York, 1988. Anais... New York, IEEE, 1988. p. 155-8.
- [75] GRIFFIN, D. & LIM, J.. Multi-band Excitation Vocoder. IEEE Transactions on Acoustics, Speech and Signal Processing, E.U.A., 36(8):1223-35. Ago. 1988.
- [76] SUGAMURA, N. & FARVARDIN, N.. Quantizer Design in LSP Speech Analysis-Synthesis. IEEE Journal on Selected Areas in Communications, E.U.A. 6(2):432-40. Fev. 1988.
- [77] COPPERI, M.. Rule-Based Speech Analysis and Application to CELP Coding. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New York, 1988. Anais... New York, IEEE, 1988. p. 143-6.
- [78] YONG, M. & GERSHO, A.. Vector Excitation Coding with Dynamic Bit Allocation. In: International Conference on Communication, Florida, 1988. Anais... Florida, IEEE, 1988. p. 290-4.
- [79] VARY, Peter et al. Speech Codec for the European Mobile Radio System. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New York, 1988. Anais... New York, IEEE, 1988. p. 227-30.
- [80] VAISEY, J. & GERSHO, A.. Simulated Annealing and Codebook Design. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New York, 1988. Anais... New York, IEEE, 1988. p.1176-9.
- [81] VIDAL, Enrique et al. On the Use of a Metric-Spece Search Algorithm (AESAs) for Fast DTW-Based Recognition of Isolated Words. IEEE Transactions on Acoustics, Speech and Signal Processing, E.U.A., 36(5):651-60. Mai. 1988.
- [82] MOODY, George et al. Evaluation of the "TRIM" ECG Data Compressor. In: Computers in Cardiology, 1988. Anais.... p. 471-4.

- [83] DIMOLITSAS, S.. Objective Speech Distortion Measures and their Relevance to Speech Quality Assessments. IEE Proceedings, Inglaterra, 136-1(5):317-24, Out. 1989.
- [84] MARLOW, S. & BUGGY, B.. Classified Vector Excitation Speech Coding. IEE Proceedings, Inglaterra, 136-1(5):325-7, Out. 1989.
- [85] TRANCOSO, Isabel M. & TRIBOLET, Josw M.. Harmonic Postprocessing of Speech Synthesised by Stochastic Coders. IEE Proceedings, Inglaterra, 136-1(2):141-4, Abr. 1989.
- [86] WANG, Shihua & GERSHO, Allen. Phonetically-Based Vector Excitation Coding of Speech at 3.6 kbps. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Glasgow, 1989. Anais... Glasgow, IEEE, 1989. p. 49-52.
- [87] TANIGUCHI, Tomohiko & UNAGAMI, Shigeyuki & GRAY, Robert M.. Multimode Coding: Application to CELP. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Glasgow, 1989. Anais... Glasgow, IEEE, 1989. p. 156-9.
- [88] JAYANT, N. S. & CHEN, J. H.. Speech Coding with Time-Varying Bit Allocations to Excitation and LPC Parameters. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Glasgow, 1989. Anais... Glasgow, IEEE, 1989. p. 65-8.
- [89] KEMP, David P. & SUEDA, Retha A., & TREMAIN, Thomas E.. An Evaluation of 4800 bps Voice Coders. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Glasgow, 1989. Anais... Glasgow, IEEE, 1989. p. 200-3.
- [90] COX, Richard V. & KLEIJN, W. B. & KROON, Peter. Robust CELP Coders for Noisy Backgrounds and Noisy Channels. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Glasgow, 1989. Anais... Glasgow, IEEE, 1989. p. 739-42.
- [91] KABAL, Peter & RAMACHANDRAN, Ravi P.. Pitch Prediction Filters in Speech Coding. IEEE Transactions on Acoustics, Speech, and Signal Processing, E.U.A., 37(4):467-78, Abr. 1989.
- [92] ATAL, Bishnu S. & COX, Richard V. & KROON, Peter. Spectral Quantization and Interpolation for CELP Coders. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Glasgow, 1989. Anais... Glasgow, IEEE, 1989. p. 69-72.
- [93] FURUI, Sadaoki. Digital Speech Processing, Synthesis, and Recognition. New York, Marcel Dekker, 1989. 390p.
- [94] LIPPMANN, R.. Pattern Classification Using Neural Networks. IEEE Communications Magazine. :47-64. Nov. 1989.
- [95] CAMPBELL, J. & WELCH, V. & TREMAIN, T.. An expandable error-protected 4800 bps CELP Coder (U.S. Federal Standard 4800 bps Voice Coder). In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Glasgow, 1989. Anais... Glasgow, IEEE, 1989. p. 735-8.

- [96] PALIWAL, K. & RAMASUBRAMANIAN, V.. Effect of Ordering the Codebook on the Efficiency of the Partial Distance Search Algorithm for Vector Quantization. IEEE Transactions on Communications, E.U.A., 37(5):538-40. Mai. 1989.
- [97] EQUITZ, W.. A New Vector Quantization Clustering Algorithm. IEEE Transactions on Acoustics, Speech and Signal Processing, E.U.A., 37(10):1568-75. Out. 1989.
- [98] SOLEYMANI, M. & MORGERA, S.. A Fast MMSE Encoding Technique for Vector Quantization. IEEE Transactions on Communications, E.U.A., 37(6):656-9. Jun. 1989.
- [99] ELECTRONICS INDUSTRIES ASSOCIATION. Dual-Mode Mobile Station - Base Station Compatibility Standard. EIA/TIA-IS54. p. 2-9 a 2-65. 1989.
- [100] GALAND, C. & MENEZ, J. & ROSSO, M.. Complexity Reduction of CELP Coders. In: INTERNATIONAL TELECOMMUNICATIONS SYMPOSIUM, Rio de Janeiro, 1990. Anais... Rio de Janeiro, SBT/IEEE, 1990. p. 181-85.
- [101] TRANCOSO, Isabel M. & ATAL, Bishnu S.. Efficient Search Procedures for Selecting the Optimum Innovation in Stochastic Coders. IEEE Transactions on Acoustics, Speech, and Signal Processing, E.U.A., 38(3):385-96, Mar. 1990.
- [102] KLEIJN, W. Bastiaan & KRASINSKI, Daniel J. & KETCHUM, Richard H.. Fast Methods for the CELP Speech Coding Algorithm. IEEE Transactions on Acoustics, Speech and Signal Processing, E.U.A. 38(8):1330-42, Ago. 1990.
- [103] ROSE, Richard C. & BARNWELL, Thomas P.. Design and Performance of an Analysis-by-Synthesis Class of Predictive Speech Coders. IEEE Transactions on Acoustics, Speech, and Signal Processing, E.U.A., 38(9):1489-503, Set. 1990.
- [104] KITAWAKI, Nobuhiko, & NAGABUCHI, Hiromi & TAKA, Masahiro & TAKAHASKI, Kenzo. Speech Coding Technology for ATM Networks. IEEE Communications Magazine, E.U.A., :21-27, Jan. 1990.
- [105] LEE, J. I. & UN, C. K.. On Reducing Computational Complexity of Codebook Search in CELP Coding. IEEE Transactions on Communications. E.U.A., 38(11):1935-7, Nov. 1990.
- [106] KROON, Peter & ATAL, Bishnu S.. Pitch Predictors with High Temporal Resolution. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New Mexico, 1990 Anais... New Mexico, IEEE, 1990. p. 661-4.
- [107] MARQUES, J. S. & TRANCOSO, Isabel M. & Tribolet, Josw M. & ALMEIDA, L. B.. Improved Pitch Prediction with Fractional Delays in CELP Coding. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New Mexico, 1990. Anais... New Mexico, IEEE, 1990. p. 665-8.
- [108] OZAWA, Kazunori. A Hybrid Speech Coding Based on Multi-Pulse and CELP at 3.2kb/s. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New Mexico, 1990. Anais... New Mexico, IEEE, 1990. p. 677-80.

- [109] MANO, Kazunori & MORIYA, Takehiro. 4.8 kbit/s Delayed Decision CELP Coder Using Tree Coding. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New Mexico, 1990. Anais... New Mexico, IEEE, 1990. p. 21-4.
- [110] AKAMINE, Masami & MISEKI, Kimio. CELP Coding with an Adaptive Density Pulse Excitation Model. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New Mexico, 1990. Anais... New Mexico, IEEE, 1990. p. 29-32.
- [111] LAFLAMME, C. & ADOUL, J-P & SU, H. Y. & MORISSETE, S.. On Reducing Computational Complexity of Codebook Search in CELP Coder through the Use of Algebraic Codes. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New Mexico, 1990. Anais... New Mexico, IEEE, 1990. p. 177-80.
- [112] GIBSON, Jerry D. & CHEONG, Yoon C. & WOO, Hong. C. & CHANG, Wen-Whei. A Comparison of Backward Adaptive Prediction Algorithms in Low Delay Speech Coders. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New Mexico, 1990. Anais... New Mexico, IEEE, 1990. p. 237-40.
- [113] GERSON, Ira A. & JASIUK, Mark A.. Vector Sum Excited Linear Prediction (VSELP) Speech Coding at 8 kbps. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New Mexico, 1990. Anais... New Mexico, IEEE, 1990. p. 461-4.
- [114] CASAJÚS-QUIRÓS, F. J. & HERNÁNDES-GÓMES, L. A. & GARCÍA-MATEO, C. Analysis and Quantization Procedures for a Real-Time Implementation of a 4.8 kb/s CELP Coder. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New Mexico, 1990. Anais... New Mexico, IEEE, 1990. p. 609-12.
- [115] ALCAIM, A. & BIGLIERI, E. & BOISSON DE MARCA, J. R.. Quantização Vetorial. CETUC/PUC-RIO, 1990. 50p.
- [116] ALCAIM, A. & BIGLIERI, E. & BOISSON DE MARCA, J. R.. Codificação Paramétrica. CETUC/PUC-RIO, 1990. 109p.
- [117] WANG, S. & GERSHO, A.. Improved Excitation for Phonetically-Segmented VXC Speech Coding Below 4Kb/s. In: GLOBECOM, 1990. Anais..., IEEE, 1990. p. 946-9.
- [118] O' SHAUGHNESSY, D.. Speech Communication: Human and Machine. Addison Wesley. 1990. 567p.
- [119] KRISHNAMURTHY, A. et al. Neural Networks for Vector Quantization of Speech and Images. IEEE Journal on Selected Areas in Communications, 8(8):1449-57, Out. 1990.
- [120] LEE, T. & PETERSON, A.. Adaptive Vector Quantization Using a Self-Development Neural Network. IEEE Journal on Selected Areas in Communications, 8(8):1458-71, Out. 1990.
- [121] HAGEN, R. & HEDELIN, P.. Low-Bit-rate Spectral Coding in CELP, a new LSP-method. Proc. ICASSP, pp. 189-192. 1990.

- [122] SOONG, F. & JUANG, B.. Optimal Quantization of LSP Parameters Using Delayed Decisions. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New Mexico, 1990. Anais... New Mexico, IEEE, 1990. p.185-8.
- [123] PHAMDO, N. & FARVARDIN, N.. Coding of Speech LSP Parameters Using TSVQ with Interblock Noiseless Coding. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, New Mexico, 1990. Anais... New Mexico, IEEE, 1990. p.193-6.
- [124] IWATA, A. & NAGASAKA, Y. & SUZUMURA, N.. Data Compression of the ECG Using Neural Network for Digital Holter Monitor. IEEE Engineering in Medicine and Biology. :53-7. Set. 1990.
- [125] CUSTÓDIO, Ricardo. Codificadores Paramétricos de Sinais de Voz com Excitação Multi-Pulso. Dissertação de Mestrado. Florianópolis, UFSC, 1990. 90p.
- [126] Códigos Corretores de Erros. Revista da Sociedade Brasileira de Telecomunicações, Brasil, 5(1). Jun. 1990.
- [127] JALALEDDINE, Sateh et al. ECG Data Compression Techniques - A Unified Approach. IEEE Transactions on Biomedical Engineering, 37(4):329-43. Abr. 1990.
- [128] SPANIAS, Andreas S. & WU, Frank H.. Speech Coding and Speech Recognition Technologies: A Review. In: INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, Singapura, 1991. Anais... Singapura, IEEE, 1991, p. 572-7.
- [129] OZAWA, Kazunori & MIYANO, Toshiki. 4kb/s Improved CELP Coder with Efficient Vector Quantization. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 213-6.
- [130] WU, Yuhang & HANSEN, Henrik B. & LARSEN, Knud J.. High Performance Coder: A Possible Candidate for the GSM Half-Rate System. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 613-6.
- [131] GRANZOW, W. & ATAL, Bishnu S. & PALIWAL, K. K. & SCHROETER, J.. Speech Coding at 4 kb/s and Lower Using Single-Pulse and Stochastic Models of LPC Excitation. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 217-20.
- [132] WONG, Siu-Wah. An Evaluation of 6.4kbit/s Speech Codecs for INMARSAT-M System. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 629-32.
- [133] MARTINS, José A. & VIOLARO, Fábio. Low Bit Rate LPC Vocoders Using Quantization and Interpolation. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 597-600.
- [134] TANIGUCHI, Tomohiko & JOHNSON, Mark & OHTA, Yasuji. Pitch Sharpening for Perceptually Improved CELP, and the Sparse-Delta Codebook for Reduced Computation. In: INTERNATIONAL

- CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 241-4.
- [135] COPPERI, Marizio. Efficient Excitation Modeling in a Low Bit-Rate CELP Coder. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 233-6.
- [136] HEDELIN, Per & BERGSTROM, Anders. Amplitude Quantization for CELP Excitation Signals. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 225-8.
- [137] LE GUYADER, A. & DI FRANCESCO, R. & LAMBLIN, C.. Derivation of Efficient CELP Coding Algorithms Using the Z-Transform Approach. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 209-12.
- [138] GERSON, Ira A. & JASIUK, Mark A.. Techniques for Improving the Performance of CELP Type Speech Coders. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 629-32.
- [139] HERNANDEZ-GÓMEZ, L. A. & CASAJÚS-QUIRÓS, F. J. & GARCÍA-MATEO, C. & ORTEGA-GARCÍA, J.. Real-Time Implementation and Evaluation of Variable Rate CELP Coders. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 585-8.
- [140] KIPPER, U. & REININGER, H. & WOLF, D.. Improved CELP Coding Using Adaptive Excitation Codebooks. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 237-40.
- [141] AKAMINE, Masami & MISEKI, Kimio. Efficient Excitation Model for Low Bit Rate Speech Coding. INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, Singapura, 1991, Anais..., Singapura, IEEE, 1991. p. 587-9.
- [142] ANDREOTTI, F. G. & MAIORANO, V. & VETRANO, L. & ZANZONI F.. Real-Time Implementation of a High Quality CELP Codec for Digital Mobile Radio System. INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, Singapura, 1991, Anais..., Singapura, IEEE, 1991. p. 324-7.
- [143] TANIGUCHI, T. & AMANO, F. & JOHNSON, M.. Improving the Performance of CELP-Based Speech Coding at Low Bit Rates. INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, Singapura, 1991, Anais..., Singapura, IEEE, 1991. p. 590-3.
- [144] MISEKI, Kimio & AKAMINE, Masami. Adaptive Bit-Allocation Between the Pole-Zero Synthesis Filter and Excitation in CELP. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 229-32.

- [145] DE IACOVO, Rosario D. & SERENO, Daniele. Embedded CELP Coding for Variable Bit-Rate Between 6.4 and 9.6 kbit/s. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 681-4.
- [146] PALIWAL, K. K. & ATAL, Bishnu S.. Efficient Vector Quantization of LPC Parameters at 24 bits/frame. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 661-4.
- [147] NETO, Boaventura M. D. & ALCAIM, Abraham. Um Estudo Comparativo de Dicionários de Excitações para Codificadores CELP. In: 9<sup>o</sup> SIMPÓSIO BRASILEIRO DE TELECOMUNICACÕES. São Paulo, 1991. Anais... São Paulo, USP, 1991. p. 4.1.1-4.1.5.
- [148] MARTINS, José A.. Vocoder LPC com Quantização Vetorial. Dissertação de Mestrado, Campinas, UNICAMP, 1991. 114p.
- [149] SPERANZINI, Irvando Luiz. Codificadores Paramétricos de Sinal de Voz com Excitação Codificada e Filtragem Adaptável. Dissertação de Mestrado, Florianópolis, UFSC, 1991. 107p.
- [150] VASEGHI, S. Finite State CELP for Variable Rate Speech Coding. IEE Proceedings, Inglaterra, 138(6):603-10. Dez. 1991.
- [151] EASTON, M. & GOODYEAR, C.. A CELP Codebook and Search Technique Using a Hopfield Net. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991.
- [152] YELDENER, S. & KONDOZ, A. & EVANS, B.. High Quality Multiband LPC Coding of Speech At 2.4 kbit/s. Electronics Letters, Inglaterra, 27(14):1287-9. 1991.
- [153] SILVA, Adailton, Quantização Vetorial: Aplicações a um Vocoder LPC. Dissertação de Mestrado, Campina Grande, UFPb. 1991. 115p.
- [154] National Communications System. Federal Standard 1016 - "Telecommunications: Analog to Digital Conversion of Radio Voice by 4,800 bit/second Code Excited Linear Prediction (CELP)". Fev. 1991.
- [155] KEMP, D. & COLLURA, J. & TREMAIN, T.. Multi-Frame Coding of LPC Parameters at 600-800 bps. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 609-12.
- [156] LAROA, R. & PHAMDO, N. & FARVARDIN, N.. Robust and Efficient Quantization of Speech LSP Parameters Using Structured Vector Quantizers. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p.641-4.
- [157] GRASS, J. & KABAL, P.. Methods of Improving Vector-Scalar Quantization of LPC Coefficients. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p.657-60.

- [158] HERMANSSON, H. et al. A Speech Codec for Cellular Radio at a Gross Bit Rate of 11.4 kb/s. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p.625-8.
- [159] ANDREOTTI, F. & MAIORANO, V. & VETRANO, L.. A 6.3 kb/s CELP Codec Suitable for Half-Rate System. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 621-4.
- [160] GRUET, C. & POMMARET, F. & DELPRAT, M.. Experiments with a Regular Pulse CELP Coder for the Pan European Half Rate Channel. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 617-20.
- [161] WUM F. & PARHI, K. & GANESAN, K.. Neural Network Vector Quantizer Design Using Sequential and Parallel Learning Techniques. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 637-40.
- [162] JOSHI, R. & POONACHA, P.. A New MMSE Encoding Algorithm for Vector Quantization. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 645-8.
- [163] ORCHAD, Michael. A Fast Nearest-Neighbor Search Algorithm. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 2297 a 2300.
- [164] HARDWICK, John & LIM, Jae. The Application of the IMBE Speech Coder to Mobile Communications. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Toronto, 1991. Anais... Toronto, IEEE, 1991. p. 249-52.
- [165] MOLLICA, Marcelo & ALCAIM, Abraham. Quantizadores Vetoriais Ganho-Forma para Codificação de Forma de Onda de Voz em Presença de Erros no Canal. In: 10<sup>o</sup> SIMPÓSIO BRASILEIRO DE TELECOMUNICACÕES, Brasília, 1992. Anais... Brasília, UnB, 1992. p.
- [166] SILVA, Adailton J. S. & NETO, Benedito G. A. & VIOLARO, Fsbio. Análise da Quantização Vetorial em Vocoder LPC. In: 10<sup>o</sup> SIMPÓSIO BRASILEIRO DE TELECOMUNICACÕES, Brasília, 1992. Anais... Brasília, UnB, 1992. p.
- [167] FERNANDES, Dênis & BERMUDEZ, Jose C. M. & SEARA, Rui. Quantização Estatística do Ganho em Codificadores CELP. In: 10<sup>o</sup> SIMPÓSIO BRASILEIRO DE TELECOMUNICACÕES, Brasília, 1992. Anais... Brasília, UnB, 1992. p.
- [168] HAGEN, R. & HEDELIN, P.. Spectral Coding by LSP Frequencies - Scalar and Segmented VQ - Methods. IEE Proceedings, Inglaterra, 139(2):118-22. Abr. 1992.
- [169] MA, W. et al. A High Quality 2.4 Kb/s Multi-Band LPC Vocoder and its Real-Time Implementation. In: International Symposium on Signals, Systems and Electronics, 1992. Anais.... p. 226-9.

- [170] CALÔBA, L.. Introdução À Computação Neuronal (Minicurso). 9º Congresso Brasileiro de Automática, Vitória, UFES, 1992.
- [171] RAMASUBRAMANIAN, V. & PALIWAL, K.. Fast K-Dimensional Tree Algorithms for Nearest Neighbor Search with Application to Vector Quantization Encoding. IEEE Transactions on Signal Processing, E.U.A., 40(3):518-31. Mar. 1992.
- [172] BOISSON DE MARCA, J. R. & DE MARTINO, E.. Codificadores de Voz para Telefonia Móvel Celular. In: 10º SIMPÓSIO BRASILEIRO DE TELECOMUNICACÕES, Brasília, 1992. Anais... Brasília, UnB, 1992. p.283-8.
- [173] PAKSOY, Erdal & CHAN, Wai-Yip & GERSHO, Allen.. Vector Quantization of Speech LSF Parameters with Generalized Product Codes. In: INTERNATIONAL CONFERENCE ON SPOKEN LANGUAGE PROCESSING, Banff, 1992. Anais... 1992.
- [174] KUO, Chih-Chung & JEAN, Fu-Rong & WANG, Hsiao-Chuan.. Low Bit-Rate Quantization of LSP Parameters Using Two-Dimensional Differential Coding. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, San Francisco, 1992. Anais.... San Francisco, IEEE, 1992. p. I-97-I-100.
- [175] CHAN, Wai-Yip.. The Design of Generalized Product-Code Vector Quantizers. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, San Francisco, 1992. Anais.... San Francisco, IEEE, 1992. p. III-389 - III-392.
- [176] FERNANDES, Dênis, Codificadores CELP: Implementação, Análise e Propostas para Redução de Complexidade e Taxa de Transmissão. Dissertação de Mestrado, Florianópolis, UFSC, 1992. 161p.
- [177] BOTTEMILLER, Robert. Comments on "A New Vector Quantization Clustering Algorithm". IEEE Transactions on Signal Processing, E.U.A., 40(2):455-6. Fev. 1992.
- [178] CHAN, Wai-Yip & GUPTA, Smita & GERSHO, Allen. Enhanced Multistage Vector Quantization by Joint Codebook Design. IEEE Transactions on Communications, E.U.A., 40(11):1693-7. Nov. 1992.
- [179] ZEGER, Kenneth & VAISEY, Jacques & GERSHO, Allen. Globally Optimal Vector Quantizer Design by Stochastic Relaxation. IEEE Transactions on Signal Processing, E.U.A., 40(2):310-22. Fev. 1992.
- [180] YAIR, Eyal & ZEGER, Kenneth & GERSHO, Allen. Competitive Learning and Soft Competition for Vector Quantizer Design. IEEE Transactions on Signal Processing, E.U.A., 40(2):294-309. Fev. 1992.
- [181] National Communications System. Details to Assist in Implementation of Federal Standard 1016 CELP. Technical Information Bulletin 92-1. 1992. 35 p.
- [182] HUSSAIN, Yunus & FARVARDIN, Nariman. Finite-State Vector Quantization Over Noisy Channels and its Application to LSP Parameters. In: INTERNATIONAL CONFERENCE ON

- ACOUSTICS, SPEECH AND SIGNAL PROCESSING, San Francisco, 1992. Anais.... San Francisco, IEEE, 1992. p.II-133-6.
- [183] BHATTACHARYA, B. & LeBLANC, W. & MAHMOUD, S. & CUPERMAN, V.. Tree-Searched Multi-Stage Vector Quantization of LPC Parameters for 4 kb/s Speech Coding. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, San Francisco, 1992. Anais.... San Francisco, IEEE, 1992. p.I-105-8.
- [184] LANCINI, R. & PEREGO, F. & TUBARO, S.. Neural Network Approach for Adaptive Vector Quantization of Images. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, San Francisco, 1992. Anais.... San Francisco, IEEE, 1992. p. II-389-92.
- [185] CHEN, O. & SHEU, B. & FANG, W.. Adaptive Vector Quantizer for Image Compression Using Self-Organization Approach. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, San Francisco, 1992. Anais.... San Francisco, IEEE, 1992. p. II-385-8.
- [186] HUANG, C. -M. et all.. Fast Full Search Equivalent Encoding Algorithms for Image Compression Using Vector Quantization. IEEE Transactions on Image Processing, E.U.A., 1(3). Jul. 1992.
- [187] RAMASUBRAMANIAN, V. & PALIWAL, K.. An Efficient Approximation-Elimination Algorithm for Fast Nearest-Neighbor Search. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, San Francisco, 1992. Anais.... San Francisco, IEEE, 1992. p. I-89-92.
- [188] FERNANDES, Dênis & BERMUDEZ, José Carlos M. & SEARA, Rui. Codificação Paramétrica de Voz Utilizando Excitação Linearmente Combinada. In: 11<sup>o</sup> SIMPÓSIO BRASILEIRO DE TELECOMUNICACÕES, Natal, 1993. Anais... Natal, UFRN, 1993. p. 281-6.
- [189] PAKSOY, Erdal & SRINIVASAN, K. & GERSHO, Allen. Variable Rate Speech Coding with Phonetic Segmentation. A ser publicado no ICASSP'93.
- [190] COLLURA, John & TREMAIN, Thomas.. Vector Quantizer Design for the Coding of LSF Parameters. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. II-29-32.
- [191] KLAUTAU, Aldebaro & BERMUDEZ, José Carlos M. & SEARA, Rui. Codificador CELP com Quantização Vetorial Empregando Busca Rápida. In: 11<sup>o</sup> SIMPÓSIO BRASILEIRO DE TELECOMUNICACÕES, Natal, 1993. Anais... Natal, UFRN, 1993. p. 92-7.
- [192] HUANG, C. & HARRIS, R. A Comparison of Several Vector Quantization Codebook Generation Approaches. IEEE Transactions on Image Processing, 2(1):108-12. Jan. 1993.
- [193] THYSSEN, J. & HANSEN, S.. Using Neural Networks for Vector Quantization in Low Rate Speech Coders. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p.II-431-4.

- [194] PALIWAL, K. & ATAL, B.. Efficient Vector Quantization of LPC Parameters at 24 Bits/frame. IEEE Transactions on Speech and Audio Processing, E.U.A, 1(1)3-14. Jan. 1993.
- [195] HERNÁNDEZ-GÓMEZ, L. & LÓPEZ-GONZALO, E.. Phonetically-Driven CELP Coding Using Self-Organizing Maps. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. II-628-31.
- [196] KLEIJN, W. & KROON, P. & CELLARIO, L. & SERENO, D.. A 5.8 kb/s CELP Algorithm for Cellular Applications. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. II-596-9.
- [197] LUPINI, P. & HASSANEIN, H. & CUPERMAN, V.. A 2.4 kb/s Speech Codec with Class-Dependent Structure. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. II-143-6.
- [198] KUO, Chih-Chung & JEAN, Fu-Rong & Wang, Hsiao-Chuan. Speech Classification Embedded in Adaptive Codebook Search for CELP Coding. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. II-147-50.
- [199] McCREE, Alan & BARNWELL III, Thomas. Implementation and Evaluation of a 2400 bps Mixed Excitation LPC Vocoder. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. II-159-62.
- [200] MOHSENIAN, N. & NASRABADI, N.. Predictive Vector Quantization Using a Neural Network. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. V-245-8.
- [201] MARTINELLI, G. & RICOTTI, L. & MARCONE, G.. Neural Clustering for Optimal KLT Image Compression. IEEE Transactions on Signal Processing, E.U.A, 41(4):1737-9. Abr. 1993.
- [202] TREMAIN, T. & KEMP, D. & COLLURA, J. & KOHLER, M.. Evaluation of Low rate Speech Codes for HF. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. II-163-6.
- [203] KOTROPOULOS, C. & PITAS, I. & MAGLARA, A.. Voronoi Tessellation and Delauney Triangulation Using Euclidean Disk Growing in  $Z^2$ . In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. V-29-32.
- [204] BAKAMIDIS, Stelios. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. V-658-61.
- [205] KOK, C. et all. A VLSI Architecture for Vector Quantization with Optimal Search. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. 675-8.

- [206] da SILVA, Lúcio Martins & ALCAIM, Abraham. Um Modelo CELP com Redução de Complexidade. In: 11º SIMPÓSIO BRASILEIRO DE TELECOMUNICACÕES, Natal, 1993. Anais... Natal, UFRN, 1993. p. 269-74.
- [207] NISHIGUCHI, Masayuki et al. Vector Quantized MBE with Simplified V/UV Division at 3.0 kbps. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. II-151-4.
- [208] DALL'AGNOL, Sônia. Quantização Eficiente dos Parâmetros do Filtro em Codificadores VSELP. Dissertação de Mestrado, Rio de Janeiro, PUC/Rio. 1993. 176p.
- [209] de GARRIDO, Diego & PEARLMAN, W. & FINAMORE, W.. A Clustering Algorithm for Entropy-Constrained Vector Quantizer Design with Applications in Coding Image Pyramids. Submetido à publicação no IEEE Transactions on Circuits and Systems for Video Technology. 1993.
- [210] de MARCA, José Roberto Boisson. An LSF Quantizer for the North-American Half-Rate Speech Coder. A ser publicado no IEEE Transactions on Vehicular Technology. 1993.
- [211] PAN, Jianping & FISCHER, Thomas. Vector Quantization of Speech LSP Parameters Using Trellis Codes and L1-norm Constraints. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. II-17-20
- [212] LÓPEZ-SOLER, J. & FARVARDIN, Nariman. A Combined Quantization-Interpolation Scheme for Very Low Bit Rate Coding of Speech LSP Parameters. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. II-21-4.
- [213] ERZIN, Engin & ÇENTIN, A. Enis. Interframe Differential Vector Coding of Line Spectrum Frequencies. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. II-25-8.
- [214] TANAKA, Y. & TANIGUCHI, T.. Efficient Coding of LPC Parameters Using Adaptive Prefiltering and MSVQ with Partially Adaptive Codebook. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. II-5-8.
- [215] CHEMLA, David & SOONG, Sau-Wah & CHAN, Wai-Yip. Tree-Strucutred Vector Quantization of Speech LSF Parameters. In: SPEECH CODING WORKSHOP, 1993. IEEE.
- [216] XYDEAS, C. & SO, K. A Long History Quantization Approach to Scalar and Vector Quantization of LSF Coefficients. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. Anais.... Minneapolis, IEEE, 1993. p. II-1-4.
- [217] MALVONE, Kevin & FISCHER, Thomas. Enumeration and Trellis-Searched Coding Schemes for Speech LSP Parameters. IEEE Transactions on Speech and Audio Processing, E.U.A., 1(3):304-14. Jul. 1993.

- [218] ELSHAFEI AHMED, M. & AL-SUWAYYEL, M.. Fast Methods for Code Search in CELP. IEEE Transactions on Speech and Audio Processing, E.U.A., 1(3):315-25. Jul. 1993.