ADAILTON JOSÉ SANTOS SILVA

QUANTIZAÇÃO VETORIAL:

APLICAÇÕES A UM VOCODER LPC

Dissertação apresentada ao Curso de Mestrado em Engenharia Elétrica da Universidade Federal da Paraíba, em cumprimento às exigências para obtenção do Grau de Meste.

ÁREA DE CONCENTRAÇÃO: Processamento da Informação

BENEDITO GUIMARÃES AGUIAR NETO, Dr.-Ing.
Orientador

FÁBIO VIOLARO, Dr.
Co-orientador

CAMPINA GRANDE - PB
DEZEMBRO - 1992

S586q Silva, Adailton Jose Santos
Quantizacao vetorial : aplicacoes a um vocoder LPC / Adailton Jose Santos Silva. - Campina Grande, 1992.
115 f. : il.

Dissertacao (Mestrado em Engenharia Eletrica) - Universidade Federal da Paraiba, Centro de Ciencias e Tecnologia.

1. Codificacoes de Voz - 2. Processamento Digital da Voz 3. Comunicacoes Digitais 4. Quantizacao Vetorial em Codificacao 5. Processamento da Informacao 6. Dissertacao I. Aguiar Neto, Benedito Guimaraes, Dr. II. Violaro, Fabio, Dr. III. Universidade Federal da Paraiba - Campina Grande (PB) IV. Título

CDU 621.391(043)

QUANTIZAÇÃO VETORIAL: APLICAÇÃO A UM VOCODER LPC

ADAILTON JOSÉ SANTOS SILVA

DISSERTAÇÃO APROVADA EM 13.12.91

BENEDITO GUIMARÃES AGUIAR NETO, Dr.-Ing., UFPB
Orientador

FABIO VIOLARO, Dr., UNICAMP
Co-orientador

ANTONIO MARCUS NOGUEIRA LIMA, Dr., UFPB
Componente da Banca

ROSÂNGELA MARIA VILAR FRANÇA, Mestre, UFPB
Componente da Banca

CAMPINA GRANDE - PB
DEZEMBRO - 1991

Este trabalho foi realizado no Laboratório de Comunicações Digitais do DECOM/FEE/UNICAMP e contou com o apoio do Conselho Nacional de Pesquisa (CNPq), do Departamento de Engenharia Elétrica da UFPB e do convênio UNICAMP/TELEBRÁS 208/87.

Dedico este trabalho aos meus pais Maria da Penha e Manoel José, à minha esposa Ligia Cristina e ao meu filho Káio César.

AGRADECIMENTOS

- À minha mãe Maria da Penha, que muito tem me incentivado e ajudado nos momentos mais difíceis. Pessoa cujo exemplo de luta tem me servido de referência e motivação para atingir os objetivos até agora alcançados.
- Quero fazer um agradecimento especial ao Prof. Dr. Fábio Violaro que forneceu todo suporte necessário para a realização deste trabalho. Principalmente pela orientação, apoio e confiança depositados nesta dissertação.
- Ao Prof. Dr. Benedito Guimarães A. Neto, por ter viabilizado a realização deste trabalho na Universidade Estadual de Campinas e pela grande ajuda dispensada ao mesmo.
- À minha esposa Lígia Cristina pelo amor, paciência e compreensão. Que soube suportar com firmeza as conseqüências de minha ausência.
- Ao meu filho Káio César, inspiração para este trabalho.
- Ao meu pai Manoel e aos meus irmãos Antonio, Ailton, Alda, Auxiliadora e Telma, e a toda minha família por tudo que tem me proporcionado.
- A Ana Palmira e Maria Marta pela amizade e pelos bons momentos compartilhados.
- A Beatriz, Celina, Sonia, Marta, Rosana, Mônica, Regina, Belquis, Carlos, Claudio, Coradine, Izavan e Gilberto pela contribuição com suas vozes.
- A Miguel, Geber, Josué, Menotti, Egashira, Ramiro, Rosivan, Kênia, Eliane, Leandro, Leo, João, Júnior, Rogério, Eilson, Adrian, Fernando, Hassen e a todos os colegas da Faculdade de Engenharia Elétrica/UNICAMP pela amizade e companheirismo.
- Quero também agradecer aos participantes dos testes subjetivos e a todos que direta ou indiretamente contribuiram para a realização deste trabalho.

RESUMO

A utilização da Quantização Vetorial em codificação de voz tem despertado grande interesse nos pesquisadores da área, devido à possibilidade de codificação dos sinais de voz a taxas tão baixas quanto 800 bits/s.

Neste trabalho são abordados os princípios básicos da Quantização Vetorial, com ênfase e aplicações voltadas para um "VOCODER LPC". São apresentadas uma descrição sucinta do "vocoder" utilizado nas simulações e três medidas de distorção comuns em quantização vetorial (a distorção de erro quadrático e duas versões da distorção de Itakura-Saito). Também são descritos dois algoritmos de projeto de alfabetos de reprodução, o algoritmo LBG e o algoritmo "Product Code". Com estes algoritmos e as três medidas de distorção, são implementados cinco quantizadores vetoriais. São abordados também alguns métodos de geração de alfabetos iniciais, e proposto um método simples para redução de células vazias durante a geração dos alfabetos de reprodução.

São então realizadas avaliações objetivas e subjetivas dos quantizadores implementados e apresentados também os resultados dessas avaliações, sendo que para as avaliações subjetivas, utilizou-se um teste formal baseado no "score" médio de opinião.

ABSTRACT

The use of vector quantization in speech coding has raised a substantial interest on the researchers of this area, because of the possibility of digital coding of speech signals at bit rate as low as 800 bits/s.

The basic principles of the vector quantization are presented in this work, with emphasis and applications aiming at LPC VOCODER. A short description of the LPC VOCODER used in the simulations and three distortion measures more commonly used in vector quantization are presented. Two codebook generation algorithms, the LBG algorithm and the product code algorithm, are also described. Five vector quantizers, using these algorithms and the three distortion measures, are implemented. Some methods for generation of the initial codebooks are also examined and a simple method for empty cells reduction is proposed.

Then, objective and subjective evaluations of the implemented quantizers are realized and the results of these evaluations presented, with a formal test based on Mean Opinion Score (MOS) for the subjective evaluation.

ÍNDICE

CAPÍTULO 1

INTRODUÇÃO

1.1 - CODIFICAÇÃO DIGITAL DE SINAIS DE VOZ

Nas últimas cinco décadas, os princípios fundamentais da análise de voz não têm apresentado grandes inovações, ao contrário da tecnologia aplicada ao processamento de sinais de voz. Até aproximadamente 1960, a análise de voz era quase que exclusivamente realizada com tecnologia analógica; foi então que os pesquisadores começaram a utilizar computadores digitais para agilizar e facilitar, através de simulações, projetos de sistemas de comunicação mais complexos e sofisticados. Como os computadores rapidamente tornaram-se mais poderosos e menos dispendiosos, aliado ao avanço da tecnologia de circuitos integrados (sistemas extremamente complexos podem ser implementados em apenas um "chip" e permitem o processamento em tempo real), ficou evidente que as técnicas de processamento digital ofereciam mais vantagens em relação às analógicas [1]. Atualmente, através de "hardwares" digitais ou de computadores dedicados, as técnicas digitais são predominantemente utilizadas tanto em pesquisas quanto em sistemas de comunicação de voz .

O conceito de *representação* de um sinal de voz é comum a quase todas as áreas de pesquisa em sistemas de comunicação de voz.

Muitas vezes a forma de representação do sinal de voz não é o objetivo principal, mas ainda assim, está implícita na formulação de um determinado problema ou de um projeto de sistema. Como exemplo, pode-se citar a Telefonia, onde as mais diferentes formas de representação (modulação analógica AM e FM, codificação PCM, ADPCM, etc) têm sido utilizadas na trasmissão do sinal através do canal.

Em muitos casos, portanto, deve-se ter bastante cuidado na escolha e no método de representação do sinal de voz. Isto é sempre verdade para áreas tais como: armazenagem ou transmissão da voz, sistemas de resposta automática, síntese de voz, sistemas de verificação e identificação de locutor e sistemas de reconhecimento de voz. Em todas estas áreas, a representação digital tornou-se predominante devido aos motivos já citados nos parágrafos anteriores e também devido ao fato de que todas estas técnicas envolvem uma certa faixa de complexidade e sofisticação que seria praticamente impossível de se implementar com métodos analógicos [2]. A representação digital, além das vantagens apresentadas, ainda oferece mais segurança e integridade nos casos de comunicação privada, através da codificação dos sinais digitais, de modo que só o usuário do sistema possa ter acesso à informação.

Em geral, o custo dos sistemas de transmissão ou armazenagem é diretamente proporcional à quantidade de informação que pode ser efetivamente transmitida ou armazenada. Enquanto este custo decai a cada ano, a demanda para o uso desses serviços cresce a taxas maiores, comprometendo assim a capacidade destes sistemas. Então há a constante necessidade de se minimizar a quantidade de bits exigida para transmitir ou armazenar sinais com uma certa fidelidade ou qualidade [3].

A evolução da tecnologia levou a uma grande variedade de sistemas de codificação de voz. O objetivo principal, na maioria dos sistemas, tem sido a utilização de novas técnicas que possibilitem a representação do sinal de voz, preservando a inteligibilidade, numa forma que seja conveniente para transmissão

ou armazenagem. Tanto para transmissão como para armazenagem do sinal de voz, deseja-se reduzir ao máximo a taxa de codificação, mantendo sua qualidade aceitável e a um baixo custo. Sendo que este custo é, em geral, diretamente proporcional à complexidade dos sistemas de codificação, a qual, por sua vez, está intimamente relacionada ao desempenho e à eficiência destes sistemas.

Nos últimos anos, surgiu uma poderosa técnica de codificação e com grande aplicabilidade - a Quantização Vetorial - uma nova direção na codificação de fonte. Ela foi primeiramente aplicada na análise/síntese de voz e tem reduzido a taxa de "vocoders LPC" a cerca de 800 bits/s com pouca redução na qualidade [5]. Além disso, ela tem conseguido taxas tão baixas quanto 150 bits/s mantendo uma certa inteligibilidade [6]. A quantização vetorial permite um bom compromisso custo/benefício em relação à quantização escalar, já que seu desempenho é relativamente bom em torno de 1 bit/amostra, sendo que é justamente nesta faixa que a quantização escalar passa a degradar severamente o sinal. Conseqüentemente, os quantizadores vetoriais se mostram bons candidatos para diversos ramos de aplicaçãoes em codificação de sinais a baixas taxas, principalmente voz e imagem.

O propósito desta dissertação foi o desenvolvimento e a simulação de alguns algoritmos, baseado nas técnicas de quantização vetorial, de modo a quantizar os parâmetros - ganho e coeficientes de um filtro digital variante no tempo - gerados por um codificador de voz do tipo VOCODER LPC. O quantizador vetorial incidirá diretamente sobre o codificador de voz, tratando cada conjunto de coeficientes referente a um quadro de voz como um vetor e quantizando o conjunto de vetores, procurando manter a qualidade do sinal em níveis aceitáveis.

Embora a qualidade não seja compatível com aplicações em telefonia, o "VOCODER" com quantização vetorial possui uma série de aplicações em sistemas de reconhecimento de voz , reconhecimento de palavras isoladas [3] e em sistemas de resposta automática.

Com relação aos métodos de avaliação empregados, foram

realizados testes objetivos, que consistiram de avaliação da Distorção Média Total e da Relação Sinal Ruído de Quantização, e testes subjetivos, através da utilização do método MOS (Mean Opinion Square) [8].

1.2 - ESTRUTURA DA DISSERTAÇÃO

Após esta rápida introdução a alguns aspectos da representação digital de sinais e da quantização vetorial, no Capítulo 2 são apresentados os aspectos da predição linear e uma descrição suscinta do "VOCODER LPC", abordando seu princípio fundamental de funcionamento e enfatizando apenas o Método da Autocorrelação utilizado na extração dos coeficientes LPC. Por não ser o escopo principal deste trabalho e por não fazer parte dos parâmetros tratados pelos quantizadores vetoriais abordados, o período de "pitch" e o método de estimação relacionado não são apresentados com muitos detalhes.

No capítulo 3 são apresentados alguns tipos de medidas de distorção que podem ser empregadas em sistemas de codificação de voz com quantização vetorial. São então enumeradas propriedades matemáticas dessas medidas e propriedades de caráter subjetivo que influenciam diretamente o desempenho do quantizador vetorial.

O Capítulo 4 é dedicado à apresentação dos conceitos teóricos relacionados à quantização vetorial, sendo explanado o princípio geral de funcionamento de um quantizador vetorial e suas propriedades fundamentais. É feito também um paralelo entre as propriedades da quantização vetorial e da quantização escalar, indicando algumas vantagens da quantização vetorial em relação à escalar.

O Capítulo 5, além de apresentar os principais métodos de geração de "codebook", é dedicado aos algoritmos de busca, abordando suas vantagens e desvantagens em relação à complexidade computacional e ao custo de armazenagem. Nesse Capítulo também é apresentado ainda um novo método de eliminação de células vazias

nos algoritmos de geração de "codebooks".

Finalmente, o Capítulo 6 apresenta os resultados dos testes e simulações realizados, discute as inovações nos métodos de geração dos "codebooks" do Capítulo 5, e também apresenta uma discussão geral, algumas conclusões e recomendações para possíveis trabalhos nesta área.

Salvo indicação em contrário, todos os gráficos e tabelas estatísticas foram obtidos através de procedimentos e programas desenvolvidos especificamente para esta dissertação.

O "Sistema de Análise e Processamento Digital de Voz SAPDV-A" [7] do Laboratório de Comunicações Digitais do Departamento de Comunicações da FEE/UNICAMP foi o suporte fundamental para a realização deste trabalho, onde foram realizados a digitalização do sinal de voz e a geração de arquivos de voz para o desenvolvimento, teste e simulação dos algoritmos, bem como a realização dos testes subjetivos.

1.3 - NOTAÇÃO GERAL E SÍMBOLOS

Neste item são apresentados as notações e os símbolos empregados nos Capítulos seguintes. A notação ou símbolo aqui não encontrado, será explicitada ao longo do texto.

$\mathcal{R}^k$	- Espaço Euclideano K-dimensional;
$\mathbf{x},\mathbf{y}$	- vetores linha no espaço $\mathcal{R}^K$;
sup	- *supremum*: O menor limite superior de um conjunto;
inf	- *infimum*: O maior limite inferior de um conjunto;
$F(\cdot)$	- Função distribuição cumulativa de probabilidade;
$F_n(\cdot)$	- Função distribuição Cumulativa de Probabilidade obtida de uma Seqüência de Treinamento com n vetores;
$Pr(\cdot)$	- Probabilidade de um evento;
$\mathcal{A},\hat{\mathcal{A}}$	- Alfabetos de reprodução (codebook);
$\mathcal{Y},\mathcal{P},S$	- Partição formada por todas as células do quantizador;
$Q\{\mathcal{A},\mathcal{Y}\}$	- Quantizador vetorial Q representado pelo "codebook" $\mathcal{A}$ e pela partição $\mathcal{Y}$;
P_i e Q_j	- Células de filtros e de ganho de um quantizador filtro-ganho.
$S(\mathcal{A},\mathcal{H})$	- Partição de um quantizador filtro-ganho, que depende de $\mathcal{A}$ e de $\mathcal{H}$;
$\mathcal{H}(\mathcal{A},S)$	- Codebook de ganho que depende de $\mathcal{A}$ e de S.

CAPÍTULO 2

VOCODER LPC

2.1 - INTRODUÇÃO

O termo "vocoder" origina-se da contração das palavras "voice" e "coder". Historicamente, o "vocoder", inventado por Homer Dudley em 1928, foi o primeiro sistema de análise e síntese de voz [10]. Desde a sua concepção, muito tempo e esforço têm sido despendidos para melhorar a qualidade do sinal sintetizado.

A largura da faixa de freqüências alocada para um locutor, numa transmissão telefônica, é de aproximadamente 3100 Hz (de 300 a 3400 Hz). Esta porção do espectro de voz é suficiente para assegurar qualidade e inteligibilidade adequadas em várias aplicações. Dudley mostrou que se pode representar o sinal de voz com taxas de codificação muito baixas e os princípios básicos por ele estabelecidos constituem ainda, a base para uma grande parte dos sistemas de representação de voz, sendo o "vocoder" o sistema no qual o sinal de voz é analisado, transformado numa representação paramétrica e depois reconstruído através da utilização dos parâmetros num sintetizador adequado.

As técnicas de predição linear têm produzido um grande impacto no desenvolvimento de sistemas "vocoder", devido ao fato de problemas de modelagem e tratamento da fonte poderem ser

eficientemente resolvidos, resultando numa melhor qualidade da voz sintética, e devido à disponibilidade de tecnologia necessária para implementação com "hardwares" em tempo real. A importância destas técnicas encontra-se na sua capacidade de fornecer uma estimação dos parâmetros da voz com boa precisão e na sua relativa velocidade computacional.

Embora existam vários tipos de "vocoders", especialmente os "vocoders LPC", e variadas técnicas de abordagens e implementações relacionadas, este capítulo limitar-se-á apenas a apresentar as técnicas aplicadas na implementação do "vocoder LPC", discutindo os fundamentos básicos da predição linear de voz, o algoritmo da autocorrelação como método utilizado no cálculo dos coeficientes LPC e os demais tópicos referentes ao codificador de voz. Antes de se discutir o "vocoder" propriamente dito, apresentar-se-ão nos itens seguintes, alguns conceitos fundamentais das técnicas de predição linear.

2.2 - PRINCÍPIOS FUNDAMENTAIS DA PREDIÇÃO LINEAR DE VOZ

Nos últimos anos, técnicas matemáticas de predição linear têm sido aplicadas em problemas de modelamento do comportamento da voz, tornando-se um poderoso método na análise de voz e predominante na estimação dos seus parâmetros e na representação do sinal a baixas taxas para armazenagem ou transmissão. Um impacto inicial ao surgimento da codificação por predição linear, foi sua utilização como um bom aplicativo em estudos de comportamento da voz em laboratórios, facilitando a solução de equações complexas com maior precisão numérica e tornando os sistemas implementáveis em tempo real.

A idéia básica da predição linear é que uma amostra do sinal de voz pode ser aproximada por uma combinação linear das amostras anteriores, e sua filosofia está intimamente relacionada ao modelo de síntese de voz apresentado a seguir.

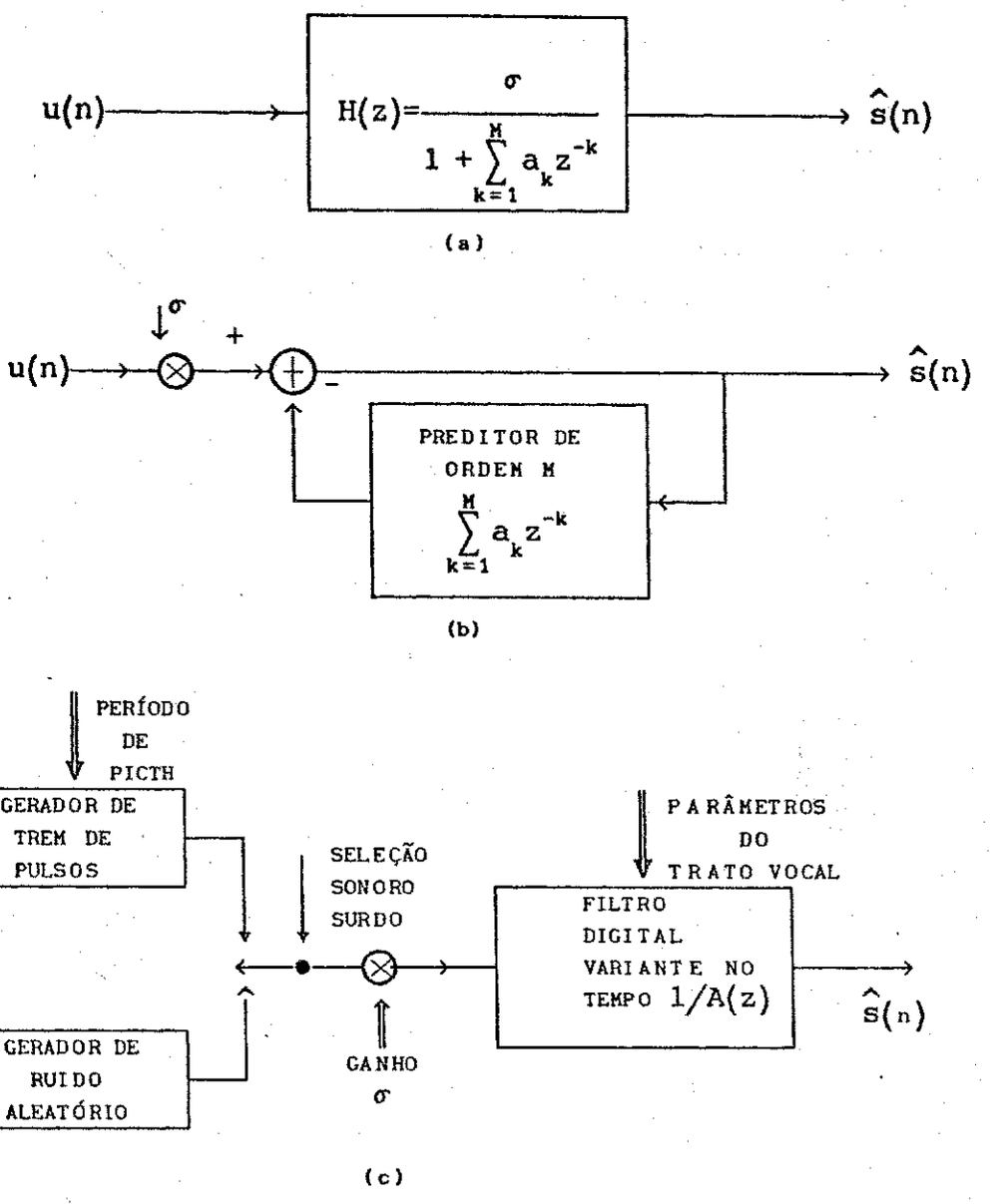


Figura 2.1 - a) Modelo básico do filtro com apenas pólos. b) Modelo discreto com apenas pólos explicitando o preditor de ordem M. c) Modelo de processamento digital para a produção do sinal de voz, onde A(z) é estabelecida pela equação 2.5.

A forma específica deste modelo, apropriada para a discussão da síntese de voz por predição linear, é mostrada na figura 2.1. Neste caso, os efeitos espectrais combinados do formato do pulso glotal, do trato vocal e da impedância de radiação nos lábios são representados por um filtro digital cuja função de transferência pode ser dada por

$$H(z) = \frac{S(z)}{U(z)} = \frac{\sigma}{1 + \sum_{K=1}^{M} a_k z^{-k}} \qquad (2.1)$$

O sistema é excitado por um trem de impulsos periódicos para voz sonora ou por uma sequência de ruído aleatório para sons não sonoros. Assim seus parâmetros são: a seleção surdo/sonoro, o período de pitch, o ganho σ e os coeficientes $\{a_k\}$ do filtro digital. Todos estes parâmetros variam lentamente com o tempo.

O modelo simplificado com apenas pólos é uma representação natural de sons sonoros não nasalizados [9 e 17], já que para esses sons a função de transferência do trato vocal não possui zeros. Já para a representação do trato vocal para os sons surdos e nasais o filtro deveria, adicionalmente, incluir zeros. Entretanto, se o número de pólos é suficientemente elevado, o modelo pode razoavelmente simular os efeitos dos sons nasais e surdos, fornecendo assim uma boa representação de quase todos os sons. As maiores vantagens deste modelo é que o ganho e os coeficientes do filtro H(z) podem ser estimados através de métodos de predição linear de maneira confiável e computacionalmente eficiente.

Para este modelo, como mostra a figura 2.1.b, as amostras da voz s(n) são relacionadas à excitação u(n) pela equação

$$s(n) = -\sum_{k=1}^{M} a_k s(n-k) + \sigma\, u(n) \qquad (2.2)$$

que estabelece a relação de síntese do sinal na predição linear, com σ sendo um fator de ganho.

O preditor linear com coeficientes de predição $\{\alpha_k\}$ é definido como um sistema cuja saída é dada por

$$\hat{s}(n) = -\sum_{k=1}^{M} \alpha_k s(n-k) \qquad (2.3)$$

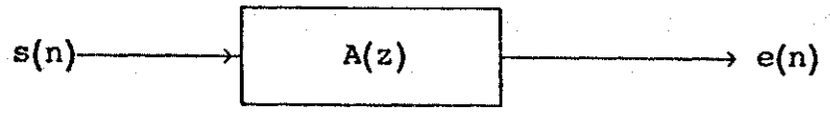


Figura 2.2 - Modelo básico com apenas zeros.

O erro de predição ou sinal residual é definido como

$$e(n) = s(n) - \hat{s}(n) \tag{2.4}$$

com $\hat{s}(n)$ estabelecido pela equação (2.3). Da equação (2.4), pode-se notar que a seqüência do erro de predição é, como mostra a figura 2.2, a saída de um sistema com função de transferência igual a

$$A(z) = 1 + \sum_{k=1}^{M} \alpha_k z^{-k}, \tag{2.5}$$

quando se aplica à sua entrada o sinal de voz.

Das equações (2.3) e (2.4), se o sinal de voz obedece ao modelo da equação (2.2) e se $\alpha_k = a_k$, então $e(n) = \sigma u(n)$. Desse modo, o filtro do erro de predição A(z) será um filtro inverso em relação ao filtro H(z) da equação (2.1), ou seja,

$$H(z) = \frac{\sigma}{A(z)} \tag{2.6}$$

O problema básico da análise por predição linear é a determinação do conjunto de coeficientes $\{\alpha_k\}$ diretamente do sinal de voz, de forma a se obter uma boa estimação das propriedades do sinal através do filtro digital H(z) da equação (2.6). Como o sinal de voz é, por natureza, variante no tempo, o procedimento básico é encontrar o conjunto de coeficientes que minimize o erro quadrático de predição sobre um curto segmento do sinal de voz, considerando-o localmente estacionário. Os coeficientes calculados são então introduzidos na função do sistema H(z) do modelo de produção de voz.

Pode não ser óbvio que este método produza resultados proveitosos, mas ele é justificado por vários motivos. Primeiro, lembrando que se $\alpha_k = a_k$, então $e(n) = \sigma u(n)$. Isto significa que, para sons sonoros, como $u(n)$ é um trem de impulsos periódicos com período igual ao período de "pitch", o erro $e(n)$ será pequeno a maior parte do tempo. A segunda motivação está no fato de que se o sinal é gerado pela equação de síntese (2.2) com coeficientes não variantes no tempo e excitado ou por um simples impulso ou por um ruído branco estacionário, então pode-se mostrar que os coeficientes de predição resultantes da minimização do erro médio quadrático, serão idênticos aos coeficientes da equação (2.2), ou seja, os coeficientes resultantes correspondem exatamente aos do modelo. A terceira e última justificativa é que, para se obter os parâmetros do filtro preditor, este método leva a um conjunto de equações lineares que podem ser facilmente resolvidas com algoritmos computacionalmente eficientes.

O erro quadrático de predição para um curto intervalo de tempo é definido como

$$\mathcal{E}_n = \sum_m \left[e_n(m) \right]^2 \qquad (2.7)$$

onde $e_n(m) = e(m+n)$ com $e(\cdot)$ definido pela equação (2.4).

Das equações (2.3) e (2.4), tem-se que

$$\mathcal{E}_n = \sum_m \left[s_n(m) + \sum_{k=1}^{M} \alpha_k s_n(m-k) \right]^2 \qquad (2.8)$$

onde

$$s_n(m) = s(m+n) \qquad (2.9)$$

Por enquanto os limites das equações (2.7) e (2.8) não foram definidos. No entanto, como a análise é realizada num curto segmento do sinal, isto é, num curto intervalo de tempo onde o sinal de voz é considerado estacionário, esses limites sempre serão finitos.

Minimizando a equação (2.8) em relação a a_k como segue

$$\frac{\partial \mathcal{E}_n}{\partial a_i} = 0, \qquad i = 1,2,....,M \tag{2.10}$$

obtém-se as seguintes equações

$$-\sum_m s_n(m-i)s_n(m) = \sum_{k=1}^{M} \hat{a}_k \sum_m s_n(m-i)s_n(m-k), \quad 1 \le i \le M \tag{2.11}$$

onde $\hat{a}_k$ são os valores de a_k que minimizam $\mathcal{E}_n$. Como (2.11) possue solução única, a partir de agora se utilizará a notação $a_k=\hat{a}_k$ para representar os coeficientes ótimos que minimizam $\mathcal{E}_n$.

Escrevendo-se (2.11) sob forma mais compacta, tem-se

$$\sum_{k=1}^{M} a_k f(i,k) = - f(i,0) \qquad i= 1,2,.....,M \tag{2.12}$$

com

$$f(i,k) = \sum_m s_n(m-i)s_n(m-k) \tag{2.13}$$

O erro quadrático mínimo de predição obtido das equações (2.8) e (2.11) pode ser expresso como

$$\mathcal{E}_n = f_n(0,0) + \sum_{k=1}^{M} a_k f_n(0,k) \tag{2.14}$$

Até agora não foram explicitamente indicados os limites do somatório nas equações (2.7), (2.8) e (2.11). Entretanto, deve-se notar que os limites do somatório em (2.11) serão iguais aos limites assumidos pelo erro médio quadrático de predição nas equações (2.7) e (2.8). Como a análise é realizada em um curto intervalo de tempo, os limites devem ser sempre finitos. Há dois métodos básicos relacionados a esta questão: o método da autocorrelação e o da covariância, os quais provêm das considerações dos limites e da definição do segmento do sinal $s_n(m)$.

Neste trabalho, abordar-se-á apenas o método da autocorrelação, para o cálculo dos coeficientes do filtro preditor do codificador de voz que será especificado no item 2.3.

2.2.1 - MÉTODO DA AUTOCORRELAÇÃO

Para se calcularem os coeficientes de predição linear, deve-se primeiro calcular $f_n(i,k)$ para $1 \leq i \leq M$ e $0 \leq k \leq M$, e depois resolver a equação (2.12). Uma maneira de se determinarem os limites nos somatórios das equações (2.7), (2.8) e (2.11) é assumir que o sinal $s_n(m)$ seja nulo fora do intervalo $0 \leq m \leq N-1$, ou seja

$$s_n(m) = s(m+n)\ w(m) \tag{2.15}$$

onde $w(m)$ é uma janela de comprimento finito que é zero fora do intervalo $0 \leq m \leq N-1$. Para este caso, pode-se expressar $\mathcal{E}_n$ como

$$\mathcal{E}_n = \sum_{m=0}^{N+M-1} e_n^2(m) \tag{2.16}$$

Os limites na expressão para $f_n(i,k)$ na equação (2.13) são idênticos àqueles da equação (2.16). Assim,

$$f_n(i,k) = \sum_{m=0}^{N+M-1} s_n(m-i)\ s_n(m-k) \tag{2.17}$$

$$= \sum_{m=0}^{N-1-(i-k)} s_n(m)\ s_n(m+i-k) \qquad 1 \leq i \leq M \text{ e } 0 \leq k \leq M \tag{2.18}$$

Para este caso, $f_n(i,k)$ é igual à função de autocorrelação para um curto intervalo de tempo [9,12 e 13], ou seja

$$f_n(i,k) = R_n(i-k) \tag{2.19}$$

com

$$R_n(k) = \sum_{m=0}^{N-1-k} s_n(m)\ s_n(m+k) \tag{2.20}$$

Como $R_n(k)$ é uma função par, tem-se que

$$\phi_n(i,k) = R_n(|i-k|) \qquad i = 1,2,...,M \text{ e } k = 0,1,...,M \qquad (2.21)$$

Portanto, a equação (2.12) pode ser representada como

$$\sum_{k=1}^{M} a_k R_n(|i-k|) = - R_n(i) \qquad 1 \leq i \leq M \qquad (2.22)$$

Igualmente, o erro quadrático de predição é expresso como

$$\mathcal{E}_n = R_n(0) + \sum_{k=1}^{M} a_k R_n(k) \qquad (2.23)$$

Em forma matricial, o conjunto de equações produzido pela equação (2.22) para i=1,2,..,M, conhecidas como equações de Yuler-Walker, é apresentado como

$$\begin{bmatrix} R_n(0) & R_n(1) & R_n(2) & \dots & R_n(M-1) \\ R_n(1) & R_n(0) & R_n(1) & \dots & R_n(M-2) \\ R_n(2) & R_n(1) & R_n(0) & \dots & R_n(M-3) \\ \vdots & \vdots & \vdots & & \vdots \\ R_n(M-1) & R_n(M-2) & R_n(M-3) & .. & R_n(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_M \end{bmatrix} = (-1) \begin{bmatrix} R_n(1) \\ R_n(2) \\ R_n(3) \\ \vdots \\ R_n(M) \end{bmatrix} \qquad (2.24)$$

A matriz de autocorrelação acima é uma matriz M X M "Toeplitz", ou seja, simétrica e com os elementos iguais ao longo das diagonais. O método de solução para estas equações será apresentado no ítem 2.3.

2.2.2 - DETERMINAÇÃO DO GANHO

Pode-se relacionar o ganho σ ao sinal de excitação e ao erro de predição através das equações (2.2) e (2.4). Assim o sinal de excitação, σ u(n), pode ser representado por

$$\sigma u(n) = s(n) + \sum_{k}^{M} a_k s(n-k) \qquad (2.25)$$

enquanto que

$$e(n) = s(n) + \sum_{k=1}^{M} \alpha_k s(n-k) \qquad (2.26)$$

Especificamente, quando $\alpha_k = a_k$ então

$$e(n) = \sigma\, u(n) \qquad (2.27)$$

Isto implica dizer que o sinal de entrada é proporcional ao erro de predição, com σ sendo a constante de proporcionalidade. A equação (2.27) é apenas aproximada, pois supõe que o sinal de voz segue exatamente o modelo da equação (2.1). Adicionalmente, não é possível calcular σ de maneira confiável a partir do próprio sinal de erro. Uma suposição mais razoável é feita em termos da energia. Considera-se a energia do sinal de erro igual à energia do sinal de excitação, ou seja,

$$\sigma^2 \sum_{m=0}^{N-1} u^2(m) = \sum_{m=0}^{N-1} e^2(m) = \mathcal{E}_n. \qquad (2.28)$$

Há dois casos de interesse para a excitação. Para sons sonoros assume-se que $u(n) = \delta_T(n)$, isto é, a excitação é uma seqüência de impulsos periódicos espaçados pelo período de "pitch" T . Para sons surdos assume-se que u(n) seja um ruído branco, estacionário, de média nula e de variância unitária.

Com base nestas suposições, deve-se determinar o ganho da equação (2.28). Para sons sonoros, tem-se que

$$\mathcal{E}_n = \sigma^2 \sum_{m=0}^{N-1} \delta_T^2(n) \qquad (2.29)$$

$$\mathcal{E}_n = \frac{\sigma^2 N}{T} \qquad (2.30)$$

$$\sigma^2 = \frac{\mathcal{E}_n T}{N} \qquad (2.31)$$

com $\mathcal{E}_n$ dada pela equação (2.23). Por conveniência, denotar-se-á $\mathcal{E}_n = \alpha$ e assim a equação (2.31) fica

$$\sigma = \sqrt{\frac{T}{N}\,\alpha} \qquad (2.32)$$

Para o caso de sons surdos, tem-se da equação (2.28) que

$$\sigma^2\, N = \alpha \qquad (2.33)$$

$$\sigma = \sqrt{\frac{\alpha}{N}} \qquad (2.34)$$

Como as constantes $\sqrt{T/N}$ na equação (2.32) e $\sqrt{1/N}$ na equação (2.34) não influenciam no projeto dos quantizadores vetoriais, é comum utilizar-se $\sigma^2 = \alpha$, resultado originado das formulação de combinação espectral da predição linear [12]. Assim, o ganho pode ser calculado por

$$\sigma = \sqrt{\alpha}\ . \qquad (2.35)$$

Na síntese, entretanto, são levadas em conta as constantes de proporcionalidade.

Das equações (2.5) e (2.6) que definem o modelo básico de análise e síntese respectivamente, nota-se que as formulações da predição linear das amostras de voz equivalem a um modelo linear de produção de voz. A importância deste procedimento está no fato de que os parâmetros de análise/síntese podem ser obtidos diretamente da forma de onda do sinal de voz [12].

2.3 - O CODIFICADOR DE VOZ

O codificador de voz ou "vocoder LPC", como usualmente é chamado, é um sistema simplificado baseado no processo natural de produção de voz que consiste, como mostram as figuras 2.3.a e 2.3.b, de um transmissor que executa a análise LPC e a extração do "pitch", de um canal por onde os parâmetros são enviados, e de um

receptor que decodifica os parâmetros e, a partir destes, sintetiza o sinal de voz.

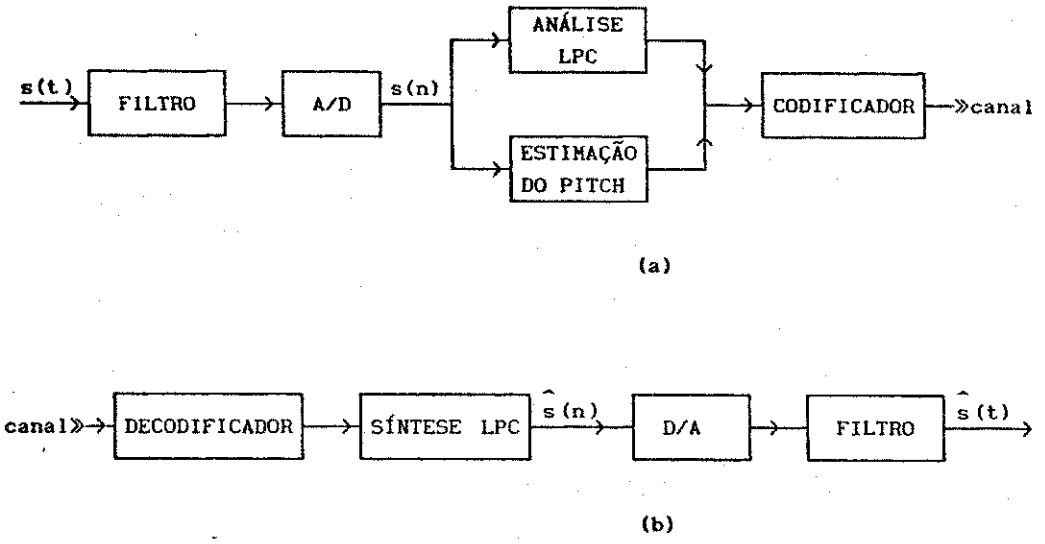


Figura 2.3 - Diagramas de blocos de um vocoder LPC. a) Transmissor e b) Receptor.

No transmissor, o sinal de voz s(t), como mostra a figura 2.3.a, é filtrado e depois é realizada a conversão analógico/digital. Então o sinal é processado quadro a quadro, com o tamanho do quadro de análise geralmente fixo. Para cada quadro, é feita uma estimação sonoro/surdo e estimado o período de "pitch" para o quadro sonoro. A análise é feita para obter-se os coeficientes do filtro H(z). Além disso, é calculado um parâmetro de ganho σ representando a energia do sinal de voz. Um processo de codificação é então aplicado, transformando os parâmetros analisados com o objetivo de minimizar a degradação na voz sintetizada para um número específico de bits disponíveis para sua codificação.

No receptor, os parâmetros transmitidos são decodificados numa versão quantizada. Um sinal de excitação é então construído a partir da decisão sonoro/surdo e do período de "pitch". Este sinal de excitação é aplicado a um filtro de síntese $H(z) = \sigma/A(z)$. Como mostra a figura 2.3.b, as amostras $\hat{s}(n)$ são então processadas num conversor digital/analógico e num filtro passa-baixas para gerar o

sinal de voz sintético $\hat{s}(t)$.

2.3.1 - TRATAMENTO DO SINAL DE VOZ

A digitalização do sinal de voz foi efetuada no Sistema de Análise e Processamento Digital de Voz, SAPDV-A [7]. Neste sistema, o sinal analógico é primeiramente limitado de 0 a 3,4 KHz por um filtro passa-baixas elíptico, com uma atenuação menor que 0,1 dB abaixo de 3,4 KHz e com atenuação acima de 34 dB para freqüências maiores que 4,6 KHz. Logo após, o sinal é amostrado a uma taxa constante de 8 KHz e quantizado linearmente em 12 bits (4096 níveis). As amostras são então transferidas ao disco rígido do microcomputador através de uma interface GPIB padrão IEEE-488. A duração do sinal de voz a ser digitalizado está limitada à quantidade disponível de memória no disco. O SAPDV-A também possui recursos para a reprodução acústica dos arquivos digitalizados, facilitando assim as avaliações subjetivas, bem como facilidade para a monitoração dos sinais em osciloscópios.

As gravações foram realizadas no Laboratório de Comunicações Digitais, que não dispõe de nenhum tipo de isolamento acústico. Entretanto, tais gravações foram feitas na ausência de ruídos provocados por motores de automóveis, condicionadores de ar, etc.

2.3.2 - ANÁLISE LPC

Como uma limitação prática e computacional, geralmente é desejável usar, principalmente em quantização vetorial, um número mínimo de parâmetros para modelar precisamente as características significantes do sinal de voz. A ordem do filtro preditor para o modelo LPC está diretamente relacionada à precisão desejada do trato vocal e depende da freqüência de amostragem escolhida para representar o sinal de voz. Naturalmente, a ordem do filtro deve ser escolhida de maneira a se obter uma boa representação de todos os formantes presentes no sinal , bem como uma boa representação, com o filtro de apenas pólos, dos sons nasais e surdos. De [9] tem-se a seguinte aproximação

$$M \cong \frac{f_s}{1000} \qquad (2.36)$$

onde M é a ordem do filtro preditor e f_s é a freqüência de amostragem. A equação (2.36) indica que deve existir pelo menos uma secção cilíndrica sem perdas, no modelo do tubo acústico que representa o trato vocal, para cada KHz da freqüência de amostragem [9 e 12]. Com isso, escolheu-se M = 8 já que f_s = 8 KHz.

Para compensar a queda de 6 dB/oitava resultante da combinação do espectro do pulso glotal e da impedância de irradiação no espectro do sinal de voz, o mesmo sofreu um processamento (pré-ênfase) no transmissor através de um filtro simples de apenas zeros da forma $1 - mz^{-1}$ com m = 0,9 estabelecendo a seguinte relação

$$s'(n) = s(n) - 0,9\ s(n-1) \qquad (2.37)$$

onde s'(n) é o sinal com pré-ênfase. Com a pré-ênfase, consegue-se um "vocoder" com preditor de ordem M com mesmo desempenho de um "vocoder" com preditor de ordem M+1 sem pré-ênfase.

Logo a seguir o sinal sofre um janelamento, ou seja, a seqüência resultante é multiplicada por uma janela de Hamming H_n dada pela equação (2.38). Esta janela possui a característica de fornecer maior ênfase nas amostras localizadas no seu centro e atenuar suavemente o sinal nas suas extremidades, de modo a manter as características espectrais do centro do quadro e eliminar as transições abruptas de suas extremidades. Com isso consegue-se diminuir o erro de predição quanto se tenta estimar , no ínicio da janela, o valor de amostras não nulas a partir de amostras nulas. Idem no fim da janela, quando se tenta estimar amostras nulas a partir de amostras não nulas.

$$H_n = \begin{cases} 0,54 - 0,46.\cos(2Pn/(N-1)), & 0 \le n < N \\ 0, \text{ caso contrário.} \end{cases} \qquad (2.38)$$

Utiliza-se também, como forma de se evitar grandes flutuações dos parâmetros calculados em cada quadro, a superposição da janela de análise com as janelas adjacentes. Assim os parâmetros do quadro de análise são influenciados pelos parâmetros dos quadros anterior e posterior.

Como visto no ítem 2.2, a função de sistema do filtro do modelo digital de produção de voz é dada por

$$H(z) = \frac{\sigma}{1 + \sum_{i=1}^{M} a_i z^{-i}} \qquad (2.39)$$

Vê-se que, além da seleção sonoro/surdo e do período de "pitch", deve-se calcular o ganho σ e os M coeficientes do filtro digital.

Efetuado o cálculo dos coeficientes de autocorrelação num intervalo de curta duração, através da equações (2.24) e utilizando-se o algoritmo recursivo de Levinson-Durbin [13 e 16] apresentado a seguir, podem-se calcular os coeficientes do filtro digital.

Algoritmo de Levinson-Durbin

$$E^{(0)} = R(0) \qquad (2.40)$$

$$k_i = \left[-R(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j) \right] / E^{(i-1)} \qquad 1 \le i \le M \qquad (2.41)$$

$$a_i^{(i)} = k_i \qquad (2.42)$$

$$a_j^{(i)} = a_j^{(i-1)} + k_i a_{i-j}^{(i-1)} \qquad 1 \le j \le i-1 \qquad (2.43)$$

$$E^{(i)} = (1-k_i^2)\, E^{(i-1)} \qquad (2.44)$$

As equações (2.41) a (2.44) são resolvidas recursivamente para $i = 1,2,\ldots,M$ e a solução é dada por

$$a_j = a_j^{(M)} \qquad 1 \le j \le M \qquad (2.45)$$

Os valores $\{a_j^{(i)};\ j=1,2...,i\}$ são os coeficientes ótimos do preditor de ordem i. Da equação (2.44) pode-se escrever

$$E^{(i)} = R(0) \prod_{j=1}^{i} (1-k_j^2), \qquad (2.46)$$

onde $E^{(i)}$ é a anergia residual $\mathcal{E}_n$ na i-ésima iteração. O erro mínimo $E^{(i)}$ deve decrescer quando a ordem do preditor aumenta. De (2.46), isto significa que deve-se ter a seguinte condição para a estabilidade do filtro com apenas pólos:

$$| k_i | < 1 \qquad (2.47)$$

Os valores intermediários k_i que aparecem no algoritmo são conhecidos como coeficientes de reflexão [13 e 16]. Na literatura estatística, em modelamento autoregressivo, os negativos de k_i são conhecidos como coeficientes parcor (partial correlation) [16]. Pode-se mostrar que a equação (2.47) estabelece a condição necessária e suficiente para o filtro de apenas pólos H(z) ser estável, ou seja, todos os pólos estarem dentro do círculo unitário. Em síntese de voz, a estabilidade do filtro é um fator muito importante, porque a instabilidade pode levar a *estalos* e *estouros* no sinal sintetizado. Se $\{|k_i| \ge 1,\ i=1,2,..,M\}$, então ocorrerão pólos sobre ou fora da circunferência de raio unitário, o que é uma condição de instabilidade.

Dividindo-se a equação (2.46) pela energia R(0) do sinal, obtem-se o erro residual mínimo normalizado

$$V^{(i)} = \frac{E^{(i)}}{R(0)} = \prod_{j=1}^{i} (1-k_j^2) \qquad (2.48)$$

Das equações (2.46) e (2.47), tem-se que

$$0 < V^i \le 1 \qquad (2.49)$$

No método da autocorrelação, o algoritmo de Levinson-Durbin garante a estabilidade do filtro representado pelos coeficientes produzidos. Após a obtenção dos coeficientes do filtro preditor, o ganho s é eficientemente calculado através das equações (2.32) e (2.34) ou (2.35).

Em alguns casos, a quantização direta dos coeficientes a_i pode levar a instabilidade. É comum se realizar então um teste nos coeficientes de reflexão associados, e para tal o cálculo dos coeficientes de reflexão, a partir dos coeficientes do filtro preditor, pode ser realizado a partir do seguinte algoritmo [13]:

$$a_j^{(M)} = a_j \; ; \qquad 1 \le j \le M \tag{2.50}$$

$$k_i = a_i^{(i)} \; ; \qquad \text{para } i = M, M-1, \ldots, 2, 1 \tag{2.51}$$

$$a_j^{(i-1)} = \left[a_j^{(i)} - a_i^{(i)} a_{i-j}^{(i)} \right] / \left(1 - k_i^2\right) \qquad 1 \le j \le i-1 \tag{2.52}$$

Detecção do Período de Pitch

A detecção do período de "pitch", ou equivalentemente, estimação da freqüência fundamental de vibração das cordas vocais, é um processo essencial em uma grande variedade de sistemas de processamento de voz. Devido à importância da estimação do "pitch", diferentes algoritmos têm sido propostos [18]. Os detectores de "pitch" são largamente utilizados em sistemas de verificação e identificação de locutor e em sistemas "vocoder".

Basicamente, o detector de "pitch" é um dispositivo que executa a decisão sonoro/surdo e fornece, para sons sonoros, uma medida do período de "pitch". Os algoritmos de detecção podem, ordinariamente, ser divididos em três grandes categorias: i) os algoritmos que usam principalmente as propriedades do sinal no domínio do tempo; ii) os que utilizam as propriedades do sinal no domínio da freqüência; iii) os algoritmos híbridos, que aplicam as propriedades do sinal nos dois domínios.

Os detectores que operam no domínio do tempo atuam diretamente na forma de onda do sinal para estimar o "pitch". São mais freqüentes, para este caso, as medidas de vales e picos do sinal, as medidas do sinal de autocorrelação [18], da função AMDF (Average Magnitude Difference Function), etc.

O grupo dos que operam no domínio da freqüência usa a propriedade de que se o sinal é periódico no domínio do tempo, então ele pode ser visto como a convolução de um trem de impulsos com a resposta impulsiva do trato vocal. Em freqüência tem-se o produto das transformadas Z da excitação e da resposta impulsiva, sendo que a transformada Z da excitação é periódica com período igual período de "pitch". Assim, uma medida do espectro de freqüência do sinal pode ser feita para se estimar o "pitch".

A classe dos detectores híbridos incorpora tanto as características das abordagens no domínio do tempo como as no domínio da freqüência. Por exemplo, um detetor de "pitch" híbrido pode usar as técnicas no domínio da freqüência para obter um sinal com espectro de freqüência plano e então usar a medida de autocorrelação para estimar o período de "pitch".

Todos os algoritmos propostos têm suas limitações e pode-se assegurar que, dentre os disponíveis atualmente, nenhum pode oferecer resultados satisfatórios numa grande faixa de locutores, aplicações e ambientes de operação, ou seja, são bons apenas em casos particulares.

O "vocoder LPC" utilizado na geração dos parâmetros, nas simulações dos quantizadores vetoriais e nas avaliações subjetivas, foi implementado com o algoritmo de detecção e encadeamento do período de "pitch" proposto por Schäfer-Vincent [19], que realiza a análise do sinal de voz no domínio do tempo. Por não ser o escopo principal desta dissertação e devido à não quantização do período de "pitch" pelos quantizadores vetoriais abordados, os detalhes deste algoritmo não são apresentados.

2.3.3 - SÍNTESE LPC

No receptor, o sinal sintetizado é obtido a partir da equação (2.2). Para sons surdos, observa-se do item 2.2.2, que para a excitação u(n) do filtro H(z) pode-se utilizar um sinal de ruído com espectro plano, média nula e variância unitária. Verificou-se que é possível utilizar poucos níveis de ruído no sinal de excitação u(n), sem afetar a qualidade subjetiva do sinal sintético.

Para sons sonoros, a excitação u(n) consiste em uma seqüência de impulsos periódicos espaçados pelo período de "pitch" T. Como esse sinal não apresenta média nula, a voz sintetizada pode apresentar um ruído de baixa freqüência, principalmente quando se utiliza pré-ênfase/de-ênfase. Isto ocorre porque o filtro de de-ênfase apresenta um ganho elevado na freqüência zero. Pode-se entretanto obter um sinal de excitação com média nula utilizando-se o sinal expresso por

$$u(n) = \begin{cases} 1, \; n=0, \; T, \; 2T, \; \ldots \\ \\ -1/(T-1), \; \text{caso contrário,} \end{cases} \qquad (2.53)$$

onde T é o período de "pitch".

Após sua reconstrução, o sinal sintético é submetido a uma de-ênfase através de um filtro com apenas zeros da forma $1/(1-mz^{-1})$ com m = 0.9, para compensar a pré-ênfase no transmissor.

CAPÍTULO 3

MEDIDAS DE DISTORÇÃO

3.1 - INTRODUÇÃO

A medida de distorção é uma função de atribuição de um valor não negativo para o par entrada/saída de um sistema. A distorção entre o sinal original ou entrada e o sinal de reprodução ou saída indica o custo resultante da representação do sinal original por um sinal quantizado. Para uma função de distorção ser utilizada em sistemas de processamento de voz, ela deve possuir algumas características fundamentais tais como: 1) Significância subjetiva, ou seja, pequenos valores na distorção devem resultar numa boa qualidade da voz, assim como grandes valores devem indicar uma péssima qualidade do sinal; 2) Ser analiticamente tratável, de modo que possa ser analizada através de métodos matemáticos convencionais e não muito complexos; 3) Tratabilidade computacional, no sentido da função de distorção poder ser eficientemente calculada e aplicada em sistemas operando em tempo real.

Várias medidas de distorção, em uso atualmente, são tratáveis e algumas até relevantes subjetivamente. Entretanto, muitos pesquisadores têm descoberto que um decréscimo de poucos decibéis na distorção é completamente perceptível pelo ouvido em algumas

situações e em outras não [3]. Tem-se notado também que, enquanto as medidas de distorção objetivas são indispensáveis e úteis em sistemas de codificação de voz, há a necessidade de se fazer constantes testes de qualidade subjetiva para indicar o desempenho dos sistemas.

Neste capítulo, exibir-se-ão algumas das principais medidas de distorção atualmente utilizadas em sistemas de codificação de voz, bem como as propriedades matemáticas relacionadas. Inicialmente serão feitas algumas suposições teóricas necessárias para que se possa utilizar a medida de distorção na quantização vetorial e serão abordadas várias propriedades relevantes na implementação dos quantizadores vetoriais.

Devido ao fato de na literatura de processamento digital de voz terem surgido muitas denominações para as várias versões da distorção de Itakura-Saito, denotar-se-á como versão 1 da distorção de Itakura-Saito, de aplicação restrita à codificação de voz por predição linear, a medida apresentada no item 3.4, e como versão 2, a distorção apresentada no item 3.5.

3.2 - CONSIDERAÇÕES PRELIMINARES

Sejam os vetores $\mathbf{x}$ e $\mathbf{y} \in \mathcal{R}^K$. A distorção ou custo $d(\mathbf{x},\mathbf{y})$ de se representar $\mathbf{x}$ por $\mathbf{y}$ é uma função que assume valores reais não negativos e que satisfaz às seguintes hipóteses:

a) Para quaisquer $\mathbf{x},\mathbf{y} \in \mathcal{R}^K$ com $\mathbf{x}$ fixo, $d(\mathbf{x},\mathbf{y})$ é uma função convexa de $\mathbf{y}$, ou seja, para $\mathbf{y}_1$, $\mathbf{y}_2 \in \mathcal{R}^K$, $\lambda \in (0,1)$,

$$d(\mathbf{x} , \lambda\mathbf{y}_1 + (1-\lambda)\, \mathbf{y}_2) \leq \lambda\, d(\mathbf{x},\mathbf{y}_1) + (1-\lambda)\, d(\mathbf{x},\mathbf{y}_2). \qquad (3.1)$$

Se a desigualdade é estabelecida, então $d(\mathbf{x},\mathbf{y})$ é estritamente convexa em $\mathbf{y}$ [21].

b) Para quaisquer $\mathbf{x},\mathbf{y} \in \mathcal{R}^K$ com $\mathbf{x}$ fixo e o conjunto de vetores $\{\mathbf{y}_i ;\ i=1,2,\cdots,n\}$, se $\mathbf{y}_i$ diverge para algum i então a distorção $d(\mathbf{x},\mathbf{y}_i)$ também diverge.

c) d é localmente limitada, ou seja, para quaisquer conjuntos $\mathcal{B}_1$, $\mathcal{B}_2 \subseteq \mathcal{R}^K$,

$$\sup_{x\in\mathcal{B}_1, y\in\mathcal{B}_2} d(x,y) < \infty \ . \qquad (3.2)$$

A hipótese (a) é a propriedade principal, sendo as propriedades (b) e (c) condições para evitar possíveis problemas de consistência matemática. A seguir, alguns exemplos de medidas que satisfazem essas propriedades.

3.2.1 - NORMAS COMO MEDIDAS DE DISTORÇÃO

Seja $\|u\|$ uma norma em $\mathcal{R}^K$ de um vetor $u \in \mathcal{R}^K$. Seja Δ qualquer função convexa não constante em $[0,\infty)$ com $\Delta(0) = 0$ (isto assegura que Δ é não decrescente [20]). Então qualquer medida de distorção da forma

$$d(x,y) = \Delta\Big(\|x - y\|\Big), \qquad (3.3)$$

satisfaz (a)-(c). Como alguns exemplos de normas tem-se:

1) As normas l_p ou Holder de um vetor u,

$$\|u\|_p = \left\{ \sum_{i=1}^{K} |u_i|^p \right\}^{1/p}, \qquad (3.4)$$

onde p é um número real entre $[1,\infty)$ e K é a dimensão do vetor [21].

2) As normas Holder em $(-\pi, -\pi)$ de uma função g que assume valores complexos,

$$\|g\|_p = \left\{ \int_{-\pi}^{\pi} |g(\theta)|^p \, d\theta/2\pi \right\}^{1/p}, \qquad (3.5)$$

com $\|g\|_p \leq \|g\|_q$ se $1 \leq p \leq q$. Esta norma é utilizada para a definição da versão 2 da distorção de Itakura Saito.

3) As normas l_r que são expressas por

$$D(u) = \|u\|_r^r , \quad r \geq 1. \tag{3.6}$$

Se D é estritamente convexa, então d(x,y) é estritamente convexa em y [20]. A classe das normas l_r inclui as distorções de r-ésima potência no espaço euclideano K-dimensional da forma

$$d_r(x,y) = \|x - y\|_r^r = \sum_{i=1}^{K} |x_i - y_i|^r, \quad r \geq 1. \tag{3.7}$$

É muito comum se definir d_r como uma medida de distorção média por dimensão. Assim, a equação (3.7) fica

$$d_r(x,y) = \frac{1}{K} \sum_{i=1}^{K} |x_i - y_i|^r, \quad r \geq 1. \tag{3.8}$$

3.2.2 - PRODUTO INTERNO COMO MEDIDA DE DISTORÇÃO

Seja (x|y) um *produto interno* em $\mathcal{R}^K$. Seja D como no item anterior. A distorção d(x,y) = D((x-y|x-y)) satisfaz (a)-(c), já que por definição $(x|x)^{1/2}$ é a norma l_r com r = 2 em $\mathcal{R}^K$ [20,21]. O exemplo mais importante de um *produto interno* em $\mathcal{R}^K$ é expresso por

$$(x|y) = xBy^t = \sum_{i=1}^{K} \sum_{j=1}^{K} x_i y_j B_{ij} , \tag{3.9}$$

onde y^t é o vetor transposto de y e B é uma matriz de ponderação K x K definida positiva, podendo ser fixa ou não. Daí tem-se que

$$d_B(x,y) = (x - y)B(x - y)^t, \tag{3.10}$$

é uma distorção quadrática ponderada pela matriz B, no caso desta ser fixa. Uma medida de distorção muito utilizada em sistemas de comunicação de voz, proposta por Itakura e Saito [20,24] e que segue a equação (3.10) com uma matriz B variável será apresentada

no item 3.4. Outra medida baseada na equação anterior e muito utilizada em problemas de classificação e análise de códigos, incluindo classificação de voz, é a distorção de Mahalanobis [3,23] dada por

$$d_G(x,y) = (x - y)G^{-1}(x - y)^t, \tag{3.11}$$

obtida a partir da equação (3.10) com $B = G^{-1}$ e G sendo a matriz de covariância do vetor K-dimensional de entrada **x**.

É interessante notar que, se **B** na equação (3.10) é uma matriz identidade, então esta equação se reduz a

$$d_B(x,y) = (x-y)(x-y)^t = \|x-y\|_2^2, \tag{3.12}$$

ou seja, a distorção ponderada se torna uma distorção de erro quadrático.

É importante também observar que, se a matriz **B** na equação (3.10) for simétrica, além de ser definida positiva, então **B** pode ser decomposta na forma

$$B = P\ P^t. \tag{3.13}$$

Assim os vetores $\hat{x}$ e **y** podem ser transformados em um novo conjunto de vetores $\tilde{x}$ e $\tilde{y}$ expressos por

$$\tilde{x} = P\ x, \quad \tilde{y} = P\ y \tag{3.14}$$

e

$$\begin{aligned} d_B(x,y) &= (x - y)B(x - y)^t \\ &= (Px - Py)(Px - Py)^t \\ &= (\tilde{x} - \tilde{y})(\tilde{x} - \tilde{y})^t \\ &= \|\tilde{x} - \tilde{y}\|_2^2 = d_2(\tilde{x},\tilde{y}). \end{aligned} \tag{3.15}$$

Com isso, a distorção quadrática ponderada entre os vetores originais **x** e **y** é igual à distorção quadrática entre os vetores transformados $\tilde{x}$ e $\tilde{y}$. Apesar de não ter sido utilizada neste

trabalho, pode ser vantajoso para propósitos computacionais fazer a transformação estabelecida pela equação (3.14) em todos os vetores antes ou durante a quantização vetorial. A matriz de covariância da distorção de Mahalanobis descrita anteriormente e a matriz de autocorrelação da distorção de Itakura-Saito descrita no item 3.4, possuem a característica de simetria e são matrizes definidas positivas. Então para estas matrizes, pode-se utilizar a transformação de (3.14) e usar a distorção de erro quadrático na quantização vetorial.

3.3 - DISTORÇÃO DE ERRO QUADRÁTICO

A medida de distorção mais comum em sistemas de codificação de voz é a distorção de erro quadrático ou de distância euclideana, originada da norma l_r. Sua popularidade provém justamente do fato de sua simplicidade e tratabilidade matemáticas. Fazendo r = 2 na equação (3.8) obtém-se

$$d_2(x,y) = \frac{1}{K}\sum_{i=1}^{K} |x_i - y_i|^2. \qquad (3.16)$$

Neste caso, os espaços de entrada e de reprodução são planos euclideanos K-dimensionais e d_2 é denominada distorção de erro quadrático [3].

3.4 - DISTORÇÃO DE ITAKURA-SAITO (VERSÃO 1)

Itakura e Saito [20,24] propuseram uma medida de distorção para ser utilizada na quantização dos coeficientes LPC. Esta medida pode ser expressa como

$$d_R(x,y) = (x - y)R(x - y)^t, \qquad (3.17)$$

onde $$R = \left\{ r(|i-k|)/r(0),\ 0 \le i,k \le M-1 \right\} \qquad (3.18)$$

é, como apresentada no Capítulo anterior, a matriz de autocorrelação normalizada, simétrica e definida positiva, do sinal

de voz. Para este caso, **x** é o vetor linha que representa os coeficientes ótimos do filtro preditor $\mathbf{x} = \{a_k, 1 \leq k \leq M\}$ e **y** corresponde aos coeficientes quantizados. Pode-se mostrar também que esta medida satifaz às propriedades (a)-(c) do item 3.2.

É importante salientar que **R** na equação (3.17) é na realidade uma matriz de ponderação como definido no item 3.2.2. Entretanto, diferente da matriz **B** na eq. (3.10), aqui **R** não é simplesmente uma matriz de ponderação de erro, visto que varia na mesma proporção do vetor de entrada **x**. Esta variação da distorção de Itakura-Saito é uma medida não simétrica com respeito a seus argumentos, ou seja, $d_R(\mathbf{x},\mathbf{y}) \neq d_R(\mathbf{y},\mathbf{x})$ e, diferente da distorção quadrática ponderada, não correponde a uma distância, isto é, não é uma distorção métrica. Justamente pelo fato de não ser métrica, esta versão da distorção de Itakura-Saito é mais significante subjetivamente, já que há uma maior influência das características espectrais e da variação do sinal na medida de distorção.

3.5 - DISTORÇÃO DE ITAKURA-SAITO (VERSÃO 2)

A versão 2 da distorção de Itakura-Saito, classificada como uma medida de distorção espectral [27], representa uma indicação de diferença entre o espectro de potência do modelo de entrada e o espectro de potência do modelo de reprodução, com o modelo podendo ser um processo aleatório ou determinístico. Para os fins deste trabalho, adordar-se-á apenas a parte referente aos processo aleatórios, como é o caso do sinal de voz.

Inicialmente, faz-se necessário estabelecer algumas considerações preliminares relacionadas aos modelos envolvidos, bem como algumas propriedades de matrizes importantes ao desenvolvimento da medida de distorção.

Seja $f(q)$ a função correspondente à densidade espectral do sinal de voz. A densidade espectral $f(q)$ é uma função par não negativa definida como

$$f(q) = \sum_{n=-\infty}^{\infty} r(n)e^{-jn\Theta}, \qquad (3.19)$$

cujos coeficientes de Fourier definem a sequência de autocorrelação

expressa por

$$r(n) = \int_{-\pi}^{\pi} f(\theta) e^{jn\Theta} \frac{d\theta}{2\Pi} \ . \qquad (3.20)$$

Se r(n) é a função de autocorrelação num curto segmento do sinal, então $f(\theta)$ é a função densidade espectral de energia. Nesta dissertação em particular, a autocorrelação é definida como no Capítulo 2 e expressa por:

$$r(n) = \begin{cases} \sum_{i=0}^{M-1-n} x(i)x(i+n), \ n = 0,1,..,M \\ 0 \ , \text{ caso contrário} \ . \end{cases} \qquad (3.21)$$

Associado a cada função de autocorrelação, há uma matriz de autocorrelação (M+1) x (M+1) indicada por

$$R(f) = \left\{ r(|i-j|), \ 0 \le i,j \le M \right\} \ . \qquad (3.22)$$

Como R(f) é tipicamente uma matriz "Toeplitz", algumas propriedades matemáticas importantes baseadas nas características "Toeplitz" da matriz são apresentadas a seguir:

A primeira propriedade estabelece que, para qualquer inteiro positivo n, existe uma *forma Toeplitz* associada à função densidade espectral que é definida por:

$$T_n(\mathbf{a}) = \int_{-\pi}^{\pi} \left| \sum_{k=0}^{n} a_k e^{-j\Theta} \right|^2 f(\theta) \frac{d\theta}{2\Pi}$$

$$= \sum_{k=0}^{n} \sum_{l=0}^{n} a_k a_l r(k-l) = \mathbf{a} \ R(f) \mathbf{a}^t, \qquad (3.23)$$

onde $\mathbf{a} = \{a_0, a_1, \ldots, a_n\}$ é um vetor genérico e $\mathbf{a}^t$ é o vetor transposto de **a**. Por conveniência representar-se-á, em alguns

casos, $R(f)$ por R e $f(\theta)$ por f, sem perda de generalidade.

Definindo o termo entre módulos da equação (3.23) como o polinômio $A_n(z)$, ou seja,

$$A_n(z) = \sum_{k=0}^{n} a_k z^{-k}, \qquad (3.24)$$

com $z = e^{j\theta}$ e $a_0 = 1$, pode-se interpretar a equação (3.23) como a média, em θ, do produto $|A_n(e^{j\Theta})|^2 f(\theta)$, que é a energia resultante da passagem de um sinal com densidade espectral de energia f através do filtro digital representado por $A_n(z)$.

O valor mínimo de $T_n(a)$ para n e $f(\theta)$ fixos, sujeito à imposição de que $a_0 = 1$, será representado por $\sigma_f^2(n)$ e de [27] é expresso por

$$\sigma_f^2(n) = \det R_n(f)/\det R_{n-1}(f). \qquad (3.25)$$

onde R_n e R_{n-1} são matrizes de autocorrelação de ordens n e n - 1 respectivamente.

A minimização de $\sigma_f^2(n)$ para o cálculo do polinômio $A_n(z)$ que representa a, pode ser expressa analiticamente em termos de polinômios ortogonais [27] ou encontrada equivalentemente através de técnicas tais como o algoritmo de Levinson-Durbin descrito no Capítulo 2. Aqui, denota-se $A_n(z)$ como um polinômio de ordem n com $a_0 = 1$ que minimiza a equação (3.23). De interesse é o fato de que o polinômio de minimização $A_n(z)$ pode ser usado com $\sigma_f^2(n)$ para modelar a densidade espectral $f(\theta)$ na integral da *forma Toeplitz*. Seja $G_n(z)$ algum polinômio da forma

$$G_n(z) = \sum_{k=0}^{n} g_k z^{-k}, \qquad (3.26)$$

onde $g = \{g_0, g_1, \ldots, g_n\}$ é um vetor linha cujo transposto é representado por g^t. Então,

$$T_n(g) = \int_{-\pi}^{\pi} \left|G_n(e^{j\Theta})\right|^2 f(\theta)\, \frac{d\theta}{2\pi}$$

$$= \int_{-\pi}^{\pi} \left|G_n(e^{j\Theta})\right|^2 \frac{\sigma_f^2(n)}{\left|A_n(e^{j\Theta})\right|^2}\, \frac{d\theta}{2\pi} \ . \qquad (3.27)$$

Em análise por predição linear, $\sigma_f^2(n)/|A_n(e^{j\Theta})|^2$ corresponde ao modelo sem quantização, $|G_n(e^{j\Theta})|^2$ corresponde ao modelo quantizado e $T_n(g)$ é a energia residual de predição. Assim a equação (3.27) é vista como uma representação no domínio da freqüência do que se pode chamar de propriedade de combinação de correlação [12,27] do modelo $\sigma_f(n)/A_n(z)$ e é dessa propriedade que são originadas as distorções espectrais, inclusive as distorçõs de Itakura-Saito.

Outra propriedade que tem sido muito utilizada na literatura de predição linear é a noção de erro de predição de *um passo* que é representado por

$$\sigma^2 \triangleq \lim_{n\to\infty} \sigma^2(n) = \exp\left\{ \int_{-\pi}^{\pi} \ln\left[f(\theta)\right] \frac{d\theta}{2\pi} \right\} , \qquad (3.28)$$

onde σ_f^2 foi, por conveniência, abreviado por σ^2.

3.5.1 - MEDIDAS DE DISTORÇÃO ESPECTRAL

Estas medidas são mais facilmente definidas no domínio espectral, embora sua avaliação seja freqüentemente realizada sem referência a aquele domínio. A medida de distorção espectral é uma função de dois espectros, f e $\hat{f}$ por exemplo; a qual assinala um valor $d(f,\hat{f})$ não negativo para indicar a distorção ao se usar $\hat{f}$ para representar f. As medidas mais comuns são as que empregam a norma l_p na diferença $(f - \hat{f})$. Estas medidas são métricas ou distâncias no sentido de que elas satisfazem as propriedades de simetria $d(f,\hat{f}) = d(\hat{f},f)$ e de desigualdade triangular

$$d(f,g) \leq d(f,h) + d(h,g). \tag{3.29}$$

Entretanto, as distorções espectrais, como é o caso da versão 2 da distorção de Itakura-Saito considerada neste item, dependem apenas do logaritmo do espectro, ou conseqüentemente, da razão de espectros, e assim são medidas que dependem apenas da razão

$$d(f,\hat{f}) = d(1,\hat{f}/f) = d(f/\hat{f},1). \tag{3.30}$$

Outros aspectos importantes são os escalonamentos usados nas distorções espectrais e que consistem na normalização e na otimização em relação ao ganho. A distorção de ganho normalizado é definida por

$$d^*(f,\hat{f}) \triangleq d(f/\sigma^2,\hat{f}/\hat{\sigma}^2), \tag{3.31}$$

onde σ^2 e σ^2 são os ganhos ou erros de predição de *um passo* para f e $\hat{f}$ respectivamente, como definido na equação (3.28). A distorção de ganho otimizado é definida por

$$d'(f,\hat{f}) \triangleq \min_{\lambda \geq 0} d(f,\lambda\hat{f}). \tag{3.32}$$

Por definição, $d(f,\hat{f}) \geq d'(f,\hat{f})$ e para medidas satisfazendo a equação (3.30), pode-se ver que

$$d^*(f,\hat{f}) = d(f/\sigma^2,\hat{f}/\hat{\sigma}^2) \geq d'(f,\hat{f}). \tag{3.33}$$

A distorção de ganho normalizado é útil nos casos de consideração separada dos modelos normalizados dos filtros e dos valores do ganho, citando-se como exemplo o algoritmo "product code" [26] ulilizado na geração separada de alfabetos de reprodução de ganho e de coeficientes do filtro, que será descrito no capítulo 5.

Baseada na norma Holder, equação (3.5), a versão 2 da distorção de Itakura-Saito [27] é dada por

$$d_{IS}(f, \hat{f}) = \left\| (f/\hat{f}) - \ln(f/\hat{f}) - 1 \right\|_1, \qquad (3.34)$$

onde o termo -1 foi introduzido para assegurar $d_{IS}(\cdot) \geq 0$ uma vez que $u - \ln(u) - 1 \geq 0$, para todo real u. Usando-se a equação (3.5), a equação (3.34) pode ser reescrita como

$$d_{IS}(f, \hat{f}) = \int_{-\pi}^{\pi} (f/\hat{f}) \frac{d\theta}{2\pi} - \ln(\sigma^2/\hat{\sigma}^2) - 1, \qquad (3.35)$$

onde σ^2 e $\hat{\sigma}^2$ são os ganhos ou erros de predição de *um passo* de f e $\hat{f}$ respectivamente, de acordo com a equação (3.28).

Usando-se o fato de que a média geométrica é menor ou igual à média aritmética, pode-se escrever

$$\int_{-\pi}^{\pi} (f/\hat{f}) \frac{d\theta}{2\pi} \geq \exp \int_{-\pi}^{\pi} \ln(f/\hat{f}) \frac{d\theta}{2\pi} = \sigma^2/\hat{\sigma}^2. \qquad (3.36)$$

Aplicando a equação (3.36) na equação (3.35), obtém-se a desigualdade $d_{IS}(f,\hat{f}) \geq d_{IS}(\sigma^2,\hat{\sigma}^2)$. Em outras palavras, para ganhos espectrais, o espectro constante produz as menores distorções.

A aplicação da versão 2 da distorção de Itakura-Saito d_{IS} em predição linear torna-se mais evidente se f é a densidade espectral das amostras da voz e $\hat{f}$ o espectro do modelo de reprodução da forma

$$\hat{f}(\theta) = \alpha/|\hat{A}(e^{j\Theta})|^2 \qquad (3.37)$$

$$\hat{A}(z) = \sum_{k=0}^{M} \hat{a}_k z^{-k} \qquad (3.38)$$

com $\hat{a}_0 = 1$. De (3.27) e (3.35) tem-se que

$$d_{IS}(f,\alpha/|\hat{A}|^2) = \frac{1}{\alpha} T_M(\hat{a}) - \ln(\sigma_f^2/\alpha) - 1, \qquad (3.39)$$

onde $\hat{a} = (1,\hat{a}_1,\hat{a}_2,\ldots,\hat{a}_M)$. Para escolher $\hat{a}$ e α que minimize a

expressão (3.39), $T_M(\hat{a})$ deve ser minimizada para se obter o valor mínimo $\sigma_f^2(M)$ ocorrendo em $\hat{A}(z) = A_M(z)$. Então a energia residual α é escolhida para minimizar o resultado, ocorrendo em $\alpha = T_M(\hat{a}) = \sigma_f^2(M)$. Isto é justamente um problema de minimização da *forma Toeplitz* como descrito no início do item 3.5.

Restritamente, se $\tilde{f}$ pertencer ao conjunto de todos os filtros autoregressivos de ordem M como descrito nas equações (3.37) e (3.38), d_{IS} é minimizada por $\tilde{f} = \sigma_f^2(M)/|A_M|^2$. Para esta escolha de $\tilde{f}$, pode-se trocar f por $\tilde{f}$ nas integrais em que f ocorrer usando a propriedade de combinação de correlação da predição linear, estabelecida no domínio da freqüência por (3.27). Em particular, se $\tilde{f}$ é um subconjunto restrito do conjunto de filtros autoregressivos de ordem M e $\tilde{f} = \sigma_f^2(M)/|A_M|^2$ então, por uso direto da distorção, pode-se encontrar que

$$d_{IS}(f, \hat{f}) = d_{IS}(f, \tilde{f}) + d_{IS}(\tilde{f}, \hat{f}) \qquad (3.40)$$

Nestes casos, a relação é melhor que a propriedade da desigualdade triangular, já que a igualdade é satisfeita. De (3.40) pode-se também notar que a distorção total é minimizada se primeiro minimizar $d_{IS}(f,\tilde{f})$ em $\tilde{f}$ e depois a parte restante $d_{IS}(\tilde{f},\hat{f})$ em $\hat{f}$. Isto fica claro se se trocar $d_{IS}(f,\tilde{f})$ por seu valor mínimo

$$d_{IS}(f, \tilde{f}) = \ln\left[\sigma_f^2(M)/\sigma^2\right], \qquad (3.41)$$

um valor independente do modelo final $\hat{f}$ e uma distorção devido apenas ao uso de um modelo finito $\tilde{f}$ no lugar de um modelo infinito f. A distorção restante $d_{IS}(\tilde{f},\hat{f})$ em (3.40) pode então ser expressa por

$$d_{IS}(\tilde{f}, \hat{f}) = T_M(\hat{a})/\hat{\sigma}^2 - \ln\left[\sigma_f^2(M)/\hat{\sigma}^2\right] - 1. \qquad (3.42)$$

Para separar o ganho e o modelo normalizado durante o processo de minimização de $d_{IS}(\tilde{f},\hat{f})$, pode-se utilizar a equação dada por

$$d_{IS}(\tilde{f},\ \hat{f}) = d'_{IS}\left(\tilde{f},\ 1/|\hat{A}|^2\right) + d^*_{IS}\left(T_M(\hat{a}),\ \hat{\sigma}^2\right) \qquad (3.43)$$

onde $$d'_{IS}\left(\tilde{f},\ 1/|\hat{A}|^2\right) = \ln\left[T_M(\hat{a})/\ \sigma^2_{\tilde{f}}(M)\right] \qquad (3.44)$$

e $$d^*_{IS}\left(T_M(\hat{a}),\ \hat{\sigma}^2\right) = T_M(\hat{a})/\ \hat{\sigma}^2 - \ln\left[T_M(\hat{a})/\ \hat{\sigma}^2\right] - 1 \qquad (3.45)$$

são as distorções de Itakura-Saito de ganho otimizado e ganho normalizado respectivamente. Esta propriedade de separação da versão 2 da distorção de Itakura-Saito, representada pela equação (3.40), é muito útil quando deseja-se obter quantizadores vetoriais filtro-ganho, como o caso dos algoritmos "product code" abordados no Capítulo 5.

Para propósitos computacionais, a avaliação numérica de $T_M(\hat{a})$ nas equações anteriores é convenientemente expressa como

$$T_M(\hat{a}) = \sum_{k=0}^{M}\sum_{l=0}^{M} \hat{a}_k \hat{a}_l\, r(k-l)$$

$$= r(0)r_{\hat{a}}(0) + 2\sum_{n=1}^{M} r(n)r_{\hat{a}}(n) \qquad (3.46)$$

onde $$r_{\hat{a}}(n) = \sum_{k=0}^{M-n} \hat{a}_k \hat{a}_{k+n}\ , \quad n = 1,2,\ldots,M \qquad (3.47)$$

Observa-se que o uso da versão 2 da distorção de Itakura-Saito para a seleção do modelo de reprodução mais *semelhante*, é um processo de minimização da energia residual equivalente à análise LPC. Supõe-se assim que esta medida de distorção seja subjetivamente significante para aplicações em sistemas de codificação de voz [27].

CAPÍTULO 4

QUANTIZAÇÃO VETORIAL: CONCEITOS TEÓRICOS

4.1 - INTRODUÇÃO

A quantização independente de cada valor ou parâmetro de um sinal é denominada quantização escalar, enquanto que a quantização de um conjunto ou bloco de parâmetros é denominada quantização por bloco, quantização por combinação de códigos ou ainda quantização vetorial. A quantização vetorial é apresentada como um processo de remoção de redundância que faz uso efetivo de quatro propriedades interrelacionadas dos parâmetros do vetor, que são: dependência linear (correlação), dependência não linear, forma da função densidade de probabilidade (fdp) e dimensionalidade vetorial. Diferente pois da quantização escalar que utiliza apenas a dependência linear e a forma da fdp.

Neste Capítulo são apresentadas várias propriedades para o algoritmo do projeto de quantizadores vetoriais baseadas num modelo probabilístico da fonte ou numa longa seqüência de treinamento. São apresentadas ainda condições na fonte e nas medidas de distorção sobre as quais o algoritmo é bem definido e converge para um mínimo local. É mostrado também que se o processo aleatório é estacionário e ergódico, então no limite quando $n \to \infty$, a execução do algoritmo

numa distribuição de amostras de uma seqüência de treinamento de tamanho n produzirá o mesmo resultado como se o mesmo algoritmo fosse executado numa distribuição *verdadeira* (modelo probabilístico ideal). Assim, o quantizador projetado para trabalhar numa seqüência de treinamento suficientemente longa, também fornecerá bom desempenho em seqüências futuras que não foram utilizadas no projeto.

4.2 - CONSIDERAÇÕES PRELIMINARES

Seja $\mathbf{X} = (X_i;\ i = 1,\cdots,K)$ um vetor aleatório de valores reais descrito por uma função de distribuição cumulativa F definida no espaço Euclideano $\mathcal{R}^K$ como $F(\mathbf{x}) = Pr(X_i \leq x_i;\ i=1,..,K)$ e $\mathbf{x} = (x_i;\ i=1,\cdots,K)$ um vetor determinístico que representa os valores assumidos pelo vetor aleatório. Um quantizador vetorial de N níveis (quantizador por blocos ou quantizador K-dimensional) $Q = \{\hat{A}, \mathcal{Y}\}$ para $\mathcal{R}^K$ consiste de:

a) Um alfabeto de reprodução ou "codebook" $\hat{A} = \{\mathbf{y}_i;\ i = 1,2,..,N\}$ com $\mathbf{y}_i \in \mathcal{R}^K$ denominados vetores de reprodução ou palavras código;

b) O mapeamento $q: A \rightarrow \hat{A}$ definido por $q(\mathbf{x}) = \mathbf{y}_i$ se $\mathbf{x} \in S_i$;

c) O conjunto ou partição $\mathcal{Y} = \{S_i;\ i = 1,2,...,N\}$ de A, onde $S_i = \{\mathbf{x}: q(\mathbf{x}) = \mathbf{y}_i\}$ são subconjuntos de A.

Se um quantizador Q de N níveis é aplicado em um vetor $\mathbf{X}$ e $Pr(\mathbf{X} \in S_i) = 0$ para algum i, então pode-se remover $\mathbf{y}_i$ de $\hat{A}$ e S_i de $\mathcal{Y}$ e formar um quantizador de N-1 níveis sem afetar o desempenho do sistema. Uma opção alternativa , que será apresentada com mais detalhes no Capítulo 5, é fazer com que $\mathbf{y}_i$ seja o ponto de distorção mínima entre todos $\mathbf{y}_j$ para $j < i$, ou seja, para a célula vazia S_i é designado um vetor $\mathbf{y}_i$ que é o centróide de todos os centróides anteriores gerados durante a execução do algoritmo. Em relação à distorção média total, várias simulações e experimentos mostraram que este método produz um mínimo local de distorção menor que aquele onde simplesmente remove-se $\mathbf{y}_i$ de $\hat{A}$ e S_i de $\mathcal{Y}$.

A figura 4.1 mostra um exemplo de um particionamento do espaço bidimensional (K=2) para o processo da quantização vetorial. A

célula S_i é representada pela região fechada por linhas escuras. Qualquer vetor de entrada **x** que se encontre dentro da célula S_i é quantizado como $\mathbf{y}_i$. Os pontos representam as posições dos vetores de reprodução correspondentes a cada célula . Para este exemplo, o número de vetores é N = 18.

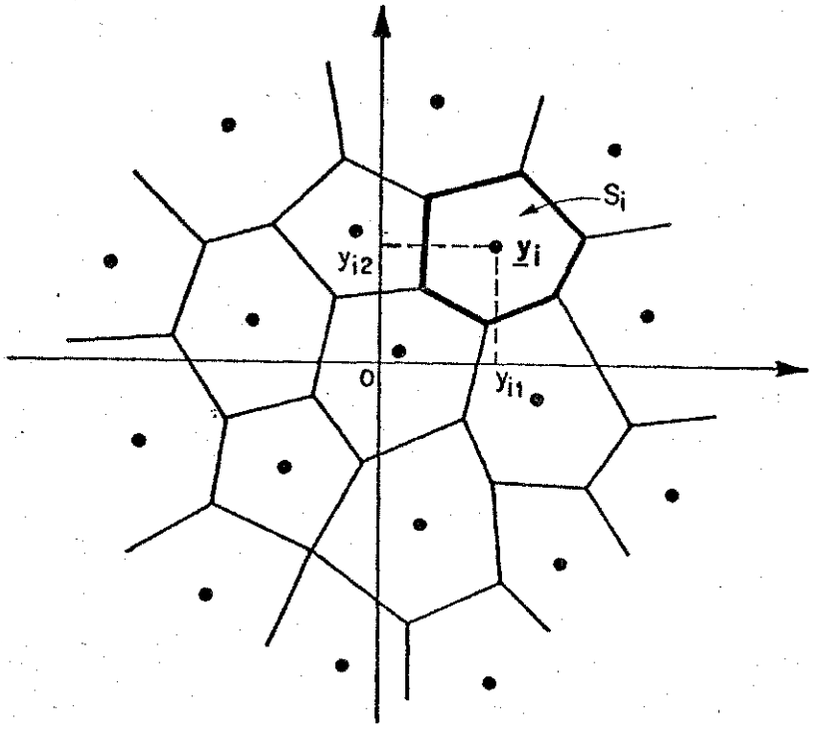


Figura 4.1- Particionamento do espaço bidimensional em N=18 células. As formas da célula podem ser muito diferentes uma das outras.

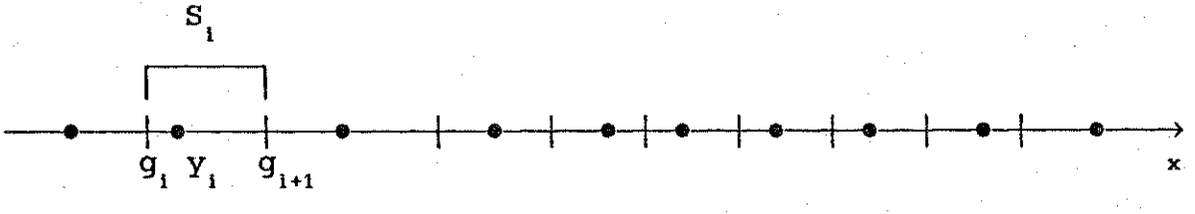


Figura 4.2- Particionamento do eixo real em N=10 células ou intervalos para a quantização escalar (k=1).

Para o caso de K=1, a quantização vetorial se reduz à quantização escalar, como mostra a figura 4.2 que apresenta um particionamento do eixo real para a quantização ecalar. Neste caso,

como na quantização vetorial, qualquer valor de entrada x dentro do intervalo S_i é quantizado como y_i. Enquanto na quantização escalar as células são apenas segmentos do eixo real, na quantização vetorial elas podem ter diferentes formas. Esta liberdade de se gerar células com várias formas no espaço multidimensional dá à quantização vetorial uma importante vantagem sobre a quantização escalar.

4.3 - PROPRIEDADES DE QUANTIZADORES ÓTIMOS

Seja Q um quantizador vetorial como no item anterior. A distorção média do quantizador Q aplicado em um vetor aleatório **X** com distribuição F é definida por

$$D(Q,F) = E\left[d(X,q(X))\right]$$

$$= \sum_{i=1}^{N} \int_{S_i} d(\mathbf{x},y_i)\,dF(\mathbf{x}) \qquad (4.1)$$

onde $E(\cdot)$ denota o valor esperado em relação a F e $d(\cdot)$, definida como no Capítulo anterior, representa uma função de custo ou distorção da representação de **x** como $q(\mathbf{x})$.

Um quantizador Q^* de N níveis é dito ser *ótimo* em F se $D(Q^*,F) \leq D(Q,F)$ para todos os possíveis quantizadores Q com N ou menos níveis [20,24]. Um quantizador Q^* é *localmente ótimo* se $D(Q^*,F)$ é um mínimo local [20], o que indica que pequenas pertubações em seu "codebook" ou partição não podem diminuir a distorção média.

Tem-se também que nenhuma partição pode produzir menor distorção média que a partição obtida pelo mapeamento de **x** em $\mathbf{y} \in \hat{A}$ que minimiza a distorção $d(\mathbf{x},\mathbf{y})$, ou seja,

$$\mathbf{x} \in S_i \text{ se } d(\mathbf{x},y_i) \leq d(\mathbf{x},y_j) \qquad (4.2)$$

para todo $i \neq j$ e $1 \leq j \leq N$.

Seja agora um quantizador $Q = \{\hat{A}, \mathcal{Y}\}$ onde varia-se o alfabeto $\hat{A}$ com uma partição $\mathcal{Y} = \{S_i\}$ fixa. Pode-se mostrar então que se $Pr(\mathbf{x} \in S_i) \neq 0$, existe um vetor de distorção mínima ou centro de gravidade $\hat{\mathbf{x}}(S_i)$ que é o centróide da célula S_i. O que implica dizer que a distorção média na equação (4.1) é mínima quando $\mathbf{y}_i$ é o centróide da célula S_i.

Assim, para qualquer partição $\mathcal{Y} = \{S_i;\ i = 1,\cdots,N\}$, define-se $\hat{\mathbf{x}}(\mathcal{Y}) = \{\mathbf{x}(S_i);\ i = 1,2,\cdots,N\}$, onde $\hat{\mathbf{x}}(\mathcal{Y})$ é o alfabeto de reprodução ótimo para a partição $\mathcal{Y}$ no sentido de que nenhum outro alfabeto possa fornecer menor distorção.

Seja agora uma medida de distorção satisfazendo as propriedades (a) e (b) do Capítulo anterior. Pode-se então mostrar que as condições necessárias para que um quantizador $Q = \{\hat{A}, \mathcal{Y}\}$ seja ótimo para F são: Que a partição seja ótima para o alfabeto de reprodução ($\mathcal{Y} = \mathcal{P}(\hat{A})$) e que o alfabeto de reprodução seja ótimo para a partição ($\hat{A} = \hat{\mathbf{x}}(\mathcal{Y})$).

Para se obter um quantizador ótimo, deve-se encontrar um ponto estacionário (mínimo global) da distorção média D(Q,F). Como não se consegue um mínimo global, é preciso encontrar um quantizador que pelo menos reúna as condições necessárias para um desempenho ótimo. Para isso, define-se um operador de mapeamento T_F, que mapeia um alfabeto de reprodução $\hat{A} = \{\mathbf{y}_i\}$ em um outro alfabeto $T_F\hat{A} \triangleq \hat{\mathbf{x}}(\mathcal{P}(\hat{A}))$. Se T_F mudar pontos sem redução na distorção, isto é, se os vetores $\mathbf{y}_i$ já são pontos de distorção mínima para $S_i \in \mathcal{P}(\hat{A})$, então assinala-se $\hat{\mathbf{x}}(S_i) = \mathbf{y}_i$.

Das suposições citadas anteriormente, uma condição necessária para um quantizador $\hat{A}$ ser ótimo é que ele seja um ponto fixo[1] de T_F. E se a partição $\mathcal{P}(T_F\hat{A})$ possuir uma célula com probabilidade zero (célula vazia), então esta célula e o vetor de reprodução correpondente podem ser removidos produzindo um quantizador com N-1 níveis.

Do operador de mapeamento T_F tem-se que $D(T_F\hat{A},F) \leq D(\hat{A},F)$, com igualdade estabelecida se e somente se $\hat{A}$ é um ponto fixo [20]. Isto

[1] As propriedades de um quantizador de ponto fixo são apresentadas no item 4.4.

assegura que, em relação à distorção média, os quantizadores obtidos após o mapeamento T_F serão melhores ou no pior caso iguais aos quantizadores antes do mapeamento. Pode ser mostrado que se a distorção d(**x**,**y**) é diferenciável em **y** e se não há probabilidade nos limites das células, isto é, se

$$\Pr\left[d(X,y_i) = d(X,y_j)\right] = 0 \qquad (4.3)$$

para todo i ≠ j, então $\hat{A}$ também é um ponto estacionário de $D(\hat{A},F)$. Deste modo, se $D(\hat{A},F)$ tiver um único ponto estacionário que é um mínimo global, então um ponto fixo é também um mínimo global. A equação (4.3) é satisfeita se F é *absolutamente contínua*. Em muitos casos, (4.3) também é satisfeita para distribuições discretas, ou seja, mesmo que $D(\hat{A},F)$ não seja absolutamente diferenciável, ela pode ser diferenciável nos pontos fixos.

4.4 - ALGORITMO DE PONTO FIXO

O seguinte algoritmo é uma generalização natural do Método I de Lloyd para K dimensões [20] e a medida de distorção deve satisfazer as propriedades (a)-(c) do Capítulo 3.

Algoritmo

1) Inicialização: Dado um alfabeto inicial $\hat{A}_0$ de N níveis, um limiar e ≥ 0 e uma uma distribuição F tal que $\Pr(X \in S_i) \neq 0$ para $S_i \in \mathcal{P}(\hat{A}_0)$ e $D(\hat{A}_0,F) < \infty$, faça m=1;

2) Dado $\hat{A}_{m-1}$, forme $\hat{A}_m = T_F\hat{A}_{m-1} = T_F^m\hat{A}_0$;

3) Calcule $D(\hat{A}_m,F)$;

4) Calcule a partição $\mathcal{P}(\hat{A}_m)$ e $\Pr(X \in S_i)$ para $S_i \in \mathcal{P}(\hat{A}_m)$. Se para M valores de i $\Pr(X \in S_i) = 0$, então faça N = N-M e remova os M níveis y_i de $\hat{A}_m$;

5) Se $\{D(\hat{A}_{m-1},F) - D(\hat{A}_m,F)\} \leq e$, pare com o alfabeto final. Caso contrário, faça m=m+1 e retorne ao passo (2).

O teste do passo (5) pode ser substituído por uma condição de percentagem da forma

$$\left\{\left[D(\hat{A}_{m-1},F) - D(\hat{A}_m,F)\right] / D(\hat{A}_m,F)\right\} \leq \epsilon. \qquad (4.4)$$

O passo (4) remove do quantizador quaisquer células vazias e seus correspondentes níveis. Assim o algoritmo pode produzir um quantizador tendo um número de níveis menor que o tamanho original N. Na prática, esta singularidade pode ocorrer e tal problema está intimamente relacionado com o tamanho da seqüência de treinamento. Se isto acontecer, o algoritmo simplesmente terá atingido um ponto onde todos os N níveis não são necessários para se obter uma redução na distorção.

Quando $D(\hat{A}_{m-1},F) = D(\hat{A}_m,F)$, $\hat{A}_{m-1}$ é um ponto fixo de T_F. Nesse caso, futuras iterações do algoritmo não mudarão o alfabeto de reprodução. Se essa condição é satisfeita, o algoritmo tem convergido para um ponto fixo.

Certas modificações no algoritmo de ponto fixo poderão ser úteis para alguns casos particulares. Por exemplo, pode ser mostrado que se a distorção $d(x,\cdot)$ é diferenciável e estritamente convexa, então uma condição necessária para $\hat{A} = \{y_i\}$ ser globalmente ótimo é que as células da partição $\mathcal{P}(\hat{A})$ não se toquem [20], o que significa que a equação (4.3) deve ser satisfeita. Esta possibilidade não está excluída quando $\hat{A}$ é um alfabeto fixo e se a distribuição F tem componentes discretas. Pode-se, portanto, adicionar passos no algoritmo para testar as probabilidades nos limites das células. Se alguma célula se tocar com outra, pode-se obter um melhor desempenho mudando o critério de parada, redeclarar os pontos limite e continuar o algoritmo. Este raro comportamento nunca ocorreu nas simulações de Linde *et alli* [24], [20], nem nas simulações realizadas para este trabalho.

Tendo o algoritmo atingido o ponto fixo, pode-se pertubar $\hat{A}$ para verificar se é possível melhoramentos adicionais, ou seja, *agita-se* o alfabeto de reprodução final libertando-o do ponto fixo ou mínimo local, para verificar se a distorção ainda pode ser minimizada de modo a produzir um mínimo local ainda melhor.

Como a distorção média $D(T_F^m \hat{A}_0,F)$ é não crescente em m, então, desde que seja não negativa, deve ter um limite da forma

$$D_{\infty}(\hat{A}_0,F) \triangleq \lim_{m=\infty} D(T_F^m \hat{A}_0,F) \qquad (4.5)$$

Espera-se que $T_F^m \hat{A}_0 = \{y_i(m)\}$ possa convergir para um ponto fixo $\hat{A}_\infty = \{y_i\}$ no sentido de que $y_i(m) \to y_i$, para os valores de i não eliminados no passo 4. Assim, quando $\hat{A}_\infty$ existe diz-se que o algoritmo converge para um ponto fixo. Mas o limite na equação (4.5) pode não existir e o algoritmo pode não convergir para um ponto fixo [20].

Seja uma medida de distorção satisfazendo as propriedades (a)-(c) do Capítulo anterior e um vetor aletório **X** com um alfabeto finito $A \subseteq \mathcal{R}^K$. Então para e ≥ 0, pode-se demonstrar que o algoritmo de ponto fixo converge para um ponto fixo em um número finito de iterações, ou seja, há um ponto fixo $\hat{A}^*$ e $M < \infty$ tal que $T_F^M \hat{A}_0 = \hat{A}^*$ [20].

Dada uma distribuição F com um alfabeto finito A e um alfabeto inicial $\hat{A}_0$, então, das suposições apresentadas, há um alfabeto de reprodução limite ou ponto fixo $\hat{A}_\infty = \hat{A}_\infty(\hat{A}_0,F)$ tal que $T_F^m \hat{A}_0 \to \hat{A}_\infty$ e o limite é obtido para m finito. No caso particular de uma distribuição amostral F_n obtida de uma seqüência de treinamento $\{x_j;\ j=1,2,\cdots,n\}$, há um $M(n) < \infty$ tal que

$$\lim_{m\to\infty} T_{Fn}^M \hat{A}_0 = T_{Fn}^{M(n)} \hat{A}_0 \triangleq \hat{A}(n), \qquad (4.6)$$

é um ponto fixo para T_{Fn} [20].

4.5 - PROPRIEDADES ASSINTÓTICAS PARA LONGAS SEQÜÊNCIAS DE TREINAMENTO

Neste item caracterizar-se-á o comportamento assintótico do algoritmo aplicado a uma distribuição amostral (F_n) baseada na seqüência de treinamento de tamanho n, quando $n \to \infty$. Em particular, se n é grande o quantizador $\hat{A}(n)$ da equação (4.6) projetado usando F_n provavelmente produzirá uma distorção $D(\hat{A}(n),F)$ quando aplicado a uma fonte *verdadeira*, que será aproximadamente $D_\infty(\hat{A}_0,F)$, o limite de distorção atingido se o algoritmo fosse executado numa

distribuição *verdadeira* com o mesmo alfabeto inicial.

Considere-se que o vetor aleatório **X** seja descrito por uma distribuição *verdadeira* F mas desconhecida e que tenha um alfabeto contínuo. Neste caso, permite-se observar uma seqüência de treinamento de vetores $\{\mathbf{x}_j;\ j=1,2,\cdots,n\}$, produzida por uma fonte estacionária e ergódica. Assim, uma estimativa natural da distribuição $F(\mathbf{x})$ com $\mathbf{x} \in \mathcal{R}^K$, é a distribuição amostral $F_n(\mathbf{x})$ definida como segue: Dado uma seqüência de treinamento $\{\mathbf{x}_j;\ j=1,2,\cdots,n\}$, definem-se um alfabeto $A_n = \{\mathbf{x}_j;\ j=1,2,\cdots,n\} \subseteq \mathcal{R}^K$ e uma medida de probabilidade b_n em $\mathcal{R}^K$ por

$$b_n(F) = \sum_j \frac{1}{n}\ ,\quad \mathbf{x}_j \in F, \tag{4.7}$$

que assinala a probabilidade 1/n para cada vetor na seqüência de treinamento. Seja então F_n a distribuição amostral correspondente à medida b_n [20], uma aplicação do teorema da ergodicidade [35] estabelece que pode-se conseguir uma seqüência de treinamento tal que $F_n \to F$ quando $n \to \infty$.

Para estabeler as propiedades assintóticas, é necessário que a medida de distorção satisfaça as propriedades (a)-(c) do Capítulo 3 e que $d(\mathbf{x},\mathbf{y})$ seja uma função estritamente convexa de $\mathbf{y}$, de modo que os pontos de distorção mínima sejam únicos. Naturalmente, escolhe-se uma seqüência de tamanho infinito de vetores $\{\mathbf{x}_j;\ j=1,2,\cdots\}$ produzida por uma fonte estacionária e ergódica. Para cada n, seja F_n uma distribuição amostral em $\mathcal{R}^K$ abrangida por $\{\mathbf{x}_j;\ j=1,2,\cdots,n\}$. Para um dado alfabeto inicial $\hat{A}_0$ e cada valor n, o algoritmo de ponto fixo pode ser executado em $\hat{A}_0$ usando F_n para se obter quantizadores $\{T_F^m \hat{A}_0;\ m=1,2,\cdots\}$, com distorção convergindo para

$$\lim_{n\to\infty} D(T_F^m \hat{A}_0, F) = D_\infty(\hat{A}_0, F). \tag{4.8}$$

Das suposições dadas, implica que se $T_F^m \hat{A}$ convergir para um ponto fixo, então $\hat{A}_\infty$ também será um ponto estacionário.

A propriedade de continuidade mais importante do algoritmo com relação à distribuição amostral é a seguinte [20]: se $\mathcal{A}$ é um alfabeto limitado, então, para $m = 1,2,\cdots$ a seqüência de treinamento é tal que

$$\lim_{n\to\infty} T^m_{F_n}\mathcal{A}_0 = T^m_F\mathcal{A}_0, \tag{4.9}$$

$$\lim_{n\to\infty} D(T^m_{F_n}\mathcal{A}_0, F) = D(T^m_F\mathcal{A}_0, F). \tag{4.10}$$

A convergência na equaçãao (4.9) é garantida apenas para os vetores $\mathbf{y}_i \in T^m_F\mathcal{A}_0$ cujas células em $\mathcal{P}(T^m_F\mathcal{A}_0)$ são não vazias, ou seja, a distorção média continua sendo finita para o alfabeto de reprodução resultante quando alguns códigos "não utilizáveis" são removidos no passo 4.

As equações (4.9) e (4.10) estabelecem que, para uma seqüência de treinamento longa, a execução do algoritmo em F_n deverá produzir resultados bem próximos daqueles produzidos se o mesmo algoritmo fosse executado em uma distribuição *verdadeira* F, e que o desempenho em dados futuros gerados por outra fonte, também estacionária e ergódica, deverá ser próximo ao desempenho produzido pelo algoritmo com dados gerados pela fonte da seqüência de treinamento.

O próximo Capítulo apresenta os métodos de geração de alfabetos de reprodução, onde são aplicados os conceitos teóricos abordados neste Capítulo.

CAPÍTULO 5

PROJETO DE ALFABETOS DE REPRODUÇÃO

5.1 - INTRODUÇÃO

O alfabeto de reprodução ou "codebook", como usualmente é chamado, é o principal componente de um quantizador vetorial. Ele é formado por um conjunto de vetores de referência[1] e tem sua importância devido ao fato de que são seus vetores de referência que deverão bem representar, através de um critério de seleção do vetor mais próximo ou mais semelhante, os vetores de entrada a serem codificados. No caso da predição linear, onde os vetores são coeficientes ou de um filtro preditor ou de autocorrelação (às vezes incluindo o parâmetro de ganho), entende-se como "mais próximo", o vetor que apresentar a menor distorção métrica ou distância e como "mais semelhante", o vetor cuja distorção

[1] Conhecidos também como vetores de reprodução, palavras-código, vetores de saída ou ainda vetores de reconstrução. Na literatura de reconhecimento de padrões, também são chamados de modelos ou padrões de referência.

espectral (não métrica) seja a menor entre todos os outros vetores[2].

No Capítulo anterior foram apresentados de forma generalizada, os conceitos e propriedades teóricas da quantização vetorial, incluindo o algoritmo de projeto de "codebooks" *localmente ótimos.* Aqui, aqueles conceitos e propriedades serão usados para produzir quantizadores vetoriais com aplicações em codificação de voz por predição linear.

Os métodos de projeto de alfabetos de reprodução, que requerem um processo iterativo envolvendo um grande número de operações, também são conhecidos como *treinamento* ou *povoamento* do "codebook". Vários algoritmos têm sido propostos, entretanto, neste Capítulo, abordar-se-ão apenas os seguintes tipos: 1) O algoritmo LBG (Linde, Buzo e Gray) [24], também conhecido como algoritmo "K-Means" [3]; 2) O algoritmo "Product Code" [25,26] ou Produto de "Codebooks", onde são abordados os métodos de otimização conjunta, otimização separada e otimização individual. Também é apresentado um método, baseado nas técnicas de otimização, para eliminação de células vazias. Preliminarmente, é imprescindível a formulação do problema da quantização vetorial.

5.2 - CONSIDERAÇÕES PRELIMINARES

Como visto no Capítulo anterior, um quantizador K-dimensional ou vetorial é um operador de mapeamento $q(\cdot)$, que designa para cada vetor de entrada $\mathbf{x}$ um vetor de referência $\mathbf{y} = q(\mathbf{x})$.

Sejam uma medida de distorção como no Capítulo 3 e os vetores $\mathbf{X}$ e $\mathbf{x}$ como definidos no Capítulo anterior. Então pode-se avaliar o desempenho de um sistema através da distorção média $E[d(\mathbf{X},q(\mathbf{X}))]$ entre o vetor de entrada e o vetor de reprodução. O sistema

[2] O Capítulo 3 apresenta as diferenças entre as distorções métrica e não métrica. Pode-se citar como exemplo de distorção métrica, a distorção de erro quadrático e de não métrica, as distorções de Itakura-Saito.

fornecerá bom desempenho se apresentar pequenos valores de distorção média. Na prática, é importante a média temporal ou amostral para representar a distorção média total do sistema. A média temporal é definida como

$$D = \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} d\left(\mathbf{x}_j, q(\mathbf{x}_j)\right). \qquad (5.1)$$

Se o vetor $\mathbf{x}_j$ é estacionário e ergódico[3], então existe o limite em (5.1) e ele é igual à média estatística expressa pela equação (4.1).

Há duas condições necessárias para se atingir o estado ótimo de um quantizador vetorial. A primeira condição é que o quantizador ótimo seja projetado usando um critério de distorção mínima ou regra do "vizinho mais próximo" expressa pela equação (4.2). A segunda condição é que cada vetor de referência deve ser escolhido de maneira a minimizar a distorção média na célula S_i.

Como visto no Capítulo anterior, tais vetores são os centróides da célula S_i e escreve-se $\mathbf{y}_i = \text{cent}(S_i)$. O cálculo do centróide para uma célula depende da medida de distorção utilizada no quantizador. Os cálculos dos centróides referentes a cada medida de distorção apresentada no Capítulo 3, são apresentados no Apêndice A. Como na prática não se tem conhecimento da distribuição F, utiliza-se então uma seqüência de treinamento que é formada por um conjunto de vetores de instrução $\{\mathbf{x}_j;\ j=1,\cdots,L\}$. Se um subconjunto S_i contém NS_i vetores de instrução , então a distorção média D_i da célula S_i pode ser expressa por

$$D_i = \frac{1}{NS_i} \sum_{\mathbf{x}\in S_i} d(\mathbf{x}, \mathbf{y}_i) \qquad (5.2)$$

onde NS_i é o número de vetores em S_i.

---

[3] A Ergodicidade permite substituir médias estatísticas por médias temporais (amostrais) [3].

5.3 - ALGORITMO LBG

O algoritmo LBG (Linde, Buzo e Gray) [24] pode ser visto como uma aplicação do algoritmo generalizado de Lloyd a modelos probabilísticos com distribuição conhecida ou modelos com distribuição desconhecida, mas em que se dispõe de uma longa seqüência de treinamento. O algoritmo é consistente quando a função distribuição de probabilidade tem componentes discretas, como é o caso da distribuição amostral obtida da seqüência de treinamento.

Seja um quantizador $q(\cdot)$ descrito pelo alfabeto de reprodução $\hat{A} = \{y_i;\ i=1,2,\cdots,N\}$ e pela partição $S = \{S_i;\ i=1,2,\cdots,N\}$. Foi visto que se uma partição $\mathcal{P}(\hat{A}) = \{P_i;\ i=1,2,\cdots,N\}$ é construída obedecendo à regra na equação (4.2) e se $\hat{A}$ é um alfabeto de reprodução fixo $\hat{A}$, a melhor partição possível é $\mathcal{P}(\hat{A})$. Foi mostrado também que se os pontos de distorção mínima (centróides) existem, então para uma partição fixa $S = \{S_i;\ i=1,2,\cdots,N\}$ nenhum alfabeto de reprodução $\hat{A} = \{y_i;\ i=1,2,\cdots,N\}$ pode produzir menor distorção média que o alfabeto de reprodução $x(S) \triangleq \{\hat{x}(S_i);\ i=1,2,\cdots,N\}$ contendo os centróides dos subconjuntos de S, e que esses centróides existem para todos os conjuntos S_i com probabilidade diferente de zero (conjuntos não vazios) e para todas as medidas de distorção apresentadas no Capítulo 3. Se a probabilidade de um conjunto S_i é zero, então o centróide pode ser definido por uma maneira arbitrária.

Naturalmente, as propriedades apresentadas sugerem um método, que é apresentado a seguir, para se projetar bons quantizadores a partir de um dado quantizador inicial, melhorando-o iterativamente.

O Algoritmo

1) Inicialização: Dados um limiar de distorção $\varepsilon \geq 0$, um alfabeto de reprodução inicial $\hat{A}_0$ com N níveis e uma distribuição conhecida F, faça $m = 0$ e $D_{-1} = \infty$.;

2) Dado um alfabeto $\hat{A}_m = \{y_i,\ i=1,2,\cdots,N\}$, encontre suas partições de distorção mínima $\mathcal{P}(\hat{A}_m) = \{S_i;\ i=1,2,\cdots,N\}$:

$$x \in S_i \text{ se } d(x,y_i) \leq d(x,y_j) \text{ para todo } j \neq i.$$

Calcule a distorção média resultante

$$D_m = D\left(\{\hat{A}_m, \mathcal{P}(\hat{A}_m)\}\right) = E\left(\min_{y \in \hat{A}_m} d(X,y)\right);$$

3) Se $(D_{m-1} - D_m)/D_m \leq e$, pare com $\hat{A}_m$ e $\mathcal{P}(\hat{A}_m)$ representando o quantizador final. Caso contrário, continue;

4) Calcule o alfabeto de reprodução ótimo $\hat{x}(\mathcal{P}(\hat{A}_m)) = \{\hat{x}(S_i); i=1,2,\cdots,N\}$ para $\mathcal{P}(\hat{A}_m)$. Faça $\hat{A}_{m+1} \underline{D} \hat{x}(\mathcal{P}(\hat{A}_m))$, m = m+1 e retorne ao passo (2).

Se em alguma iteração existir uma célula S_i tal que $Pr(X \in S_i) = 0$, então o algoritmo assinala um vetor arbitrário como centróide e continua. Regras alternativas são possíveis e algumas delas são apresentadas no item 5.5.

Foi mostrado no Capítulo anterior que a condição necessária para um quantizador ser ótimo é que ele seja um quantizador de ponto fixo. E se não existe probabilidade nos limites das células, ou seja, se

$$Pr\left(d(X,y_i) = d(X,y_j), \text{ para todo } j \neq i\right) = 0, \qquad (5.4)$$

então o quantizador é localmente ótimo. Este é sempre o caso para distribuições contínuas, mas pode ser violado para distribuições discretas. Contudo, este fato nunca ocorreu nas simulações de [24] nem nas simulações realizadas para este trabalho.

O fato do algoritmo LBG ser válido para distribuições puramente discretas, tem sua maior importância em aplicações onde não se possui *a priori* uma descrição probabilística ideal do processo a ser quantizado. Assim, como apresentado a seguir, pode-se projetar um quantizador dispondo-se apenas de uma seqüência de treinamento suficientemente longa.

Dada uma seqüência de treinamento $\{x_j; j = 1,2,\cdots,L\}$, a distorção média temporal ou amostral pode ser escrita por

$$\frac{1}{L}\sum_{n=1}^{L} d\left(x_n, q(x_n)\right). \qquad (5.5)$$

Essa é exatamente a distorção esperada $E_{F_n}(d(X,q(X)))$ com relação a uma distribuição amostral F_n determinada pela seqüência de treinamento, ou seja, a distribuição que atribui a probabilidade m/L ao vetor que ocorre m vezes na seqüência de treinamento. Assim pode-se projetar um quantizador que minimiza a distorção média temporal para a seqüência de treinamento executando o algoritmo na distribuição amostral F_n. Com isso, o cálculo da distorção média resultante no passo (2) fica modificado para

$$D_m = D\left(\{\hat{A}_m, \mathcal{P}(\hat{A}_m)\}\right) = \frac{1}{L}\sum_{j=1}^{L} \min_{y\in\hat{A}_m} d(x_j, y). \qquad (5.6)$$

Para atualizar o alfabeto de reprodução no passo (4) é necessário encontrar os vetores $\hat{x}(S_i)$ que minimizam a distorção condicional associada a cada célula S_i. Para a distorção de erro quadrático, $\hat{x}(S_i)$ é o centro de gravidade Euclideano ou centróide da célula S_i, ou seja,

$$\hat{x}(S_i) = \frac{1}{NS_i}\sum_j x_j \ , \quad x_j \in S_i \qquad (5.7)$$

onde NS_i representa o número de vetores de treinamento na célula S_i. Se $NS_i = 0$, podem-se usar as sugestões ou o algoritmo que serão apresentados no item 5.5. A derivação do centróide relativo a cada tipo de medida de distorção é apresentada no Apêndice A.

Os métodos de se gerar o alfabeto de reprodução inicial são apresentados no item 5.6 deste Capítulo. Em cada passo do algoritmo acima, a distorção média deve reduzir ou, no pior caso, não se alterar. Foi apresentado no Capítulo anterior que, sujeito a algumas condições matemáticas, o algoritmo converge para um *mínimo local*. Uma otimização global pode ser conseguida modificando-se o alfabeto de reprodução inicial para diferentes conjuntos de vetores de referência, repetindo-se o algoritmo para as várias inicializações e escolhendo-se o alfabeto resultante da menor distorção média total.

5.4 - ALGORITMO PRODUCT CODE[4]

O algoritmo "product code" ou produto de codebooks, foi introduzido para reduzir a complexidade e a quantidade de memória exigidas no processo de quantização vetorial. Neste método, o quantizador vetorial é organizado como o produto cartesiano de dois alfabetos de reprodução sendo, no caso de codificação por predição linear, um "codebook" de escalares para o ganho do filtro LPC e outro de vetores representando os coeficientes do filtro.

O quantizador total não é ótimo, já que ele possui uma forma particular. Entretanto, pode-se obter uma redução no tamanho do alfabeto de reprodução a um custo de aumento na distorção total. Por exemplo, se 5 bits são designados para representar o ganho e 10 bits para representar os coeficientes do filtro, então o tamanho do alfabeto de reprodução é reduzido de 2^{15} (32768) vetores de referência, para 2^5+2^{10} (1056) utilizando um quantizador filtro-ganho com busca plena.

O algoritmo "product code" é aplicado para quantizar o filtro e o ganho produzidos pela análise LPC. A diferença básica entre este e o algoritmo LBG é que neste caso há dois "codebooks" em questão, um para o ganho e outro para o filtro, enquanto que o LBG processa apenas um "codebook". Entretanto, os "codebooks" do algoritmo "product code" são otimizados pelo próprio algoritmo LBG.

O modelo de apenas pólos é expresso como $\{\sigma(M)/(1+a_1z^{-1}+a_2z^{-2}+\cdots+a_Mz^{-M})\}$ e representado por $\sigma(M)/A_M(z)$. Aqui, os parâmetros do modelo $\{\sigma(M)\ a_1\ a_2\ \cdots\ a_M\}$ formam o vetor de entrada. Seja y_{ij} o vetor de reprodução constituído pelos parâmetros $\{\hat{\sigma}_j\ \hat{a}_{i,1}\ \hat{a}_{i,2}\ \cdots\ \hat{a}_{i,M}\}$ onde $\hat{\sigma}_j$ é obtido de um conjunto de escalares $\{\sigma_j; j=1,2,\cdots,N_2\}$ que é o alfabeto de reprodução de ganho e $\{\hat{a}_{i,1}\ \hat{a}_{i,2}\ \cdots\ \hat{a}_{i,M}\}$ representa um dos modelos do conjunto de filtros $\{1/A_i(z); i=1,2,\cdots,N_1\}$ que é o

---

[4] Também chamado de quantizador "gain-shape" (filtro-ganho) e considerado como uma generalização do caso de ganho separado do algoritmo BGGM [25].

alfabeto de reprodução de filtros. Considere um quantizador que é formado pelo produto cartesiano de um conjunto finito de vetores e um conjunto de escalares também finito, como descrito anteriormente. Seja, $\mathcal{Z} = \{(\hat{\mathbf{a}}_i, \hat{\sigma}_j);\ i = 1,2,\cdots,N_1 \text{ e } j = 1,2,\cdots,N_2\}$ onde $\hat{\mathcal{A}} = \{\hat{\mathbf{a}}_i;\ i=1,2,\cdots,N_1\}$ é conjunto de vetores em $\mathcal{R}^K$, $\hat{\mathcal{H}} = \{\hat{\sigma}_j;\ j=1,2,\cdots,N_2\}$ é um conjunto de escalares de valores reais não negativos e $\mathcal{Z} = \hat{\mathcal{A}} \times \hat{\mathcal{H}}$ denota o produto cartesiano. Como tem-se o produto cartesiano de dois "codebooks", então cada par filtro-ganho forma um vetor de reprodução num total de $(N_1)x(N_2)$ vetores.

Referindo-se à variação 2 da distorção de Itakura-Saito, a distorção entre o vetor $\mathbf{x}$ e o vetor $\mathbf{y}_{ij}$ pode ser expressa por (vide equação (3.42))

$$d_{IS}^{M}(\mathbf{x},\mathbf{y}_{ij}) = \left\{ \frac{\hat{\mathbf{a}}_i \mathbf{R}(\mathbf{x}) \hat{\mathbf{a}}_i^t}{\hat{\sigma}_j^2} + \ln(\hat{\sigma}_j^2) \right\} - \ln\left[\sigma^2(M)\right] - 1. \quad (5.8)$$

onde $\sigma^2(M) = \alpha_M$ e com d_{IS}^{M} denominada distorção de Itakura-Saito modificada [26,29]. Observe que a minimização de d_{IS}^{M} em $\mathbf{y}_{ij}$ para um certo $\mathbf{x}$, é equivalente à minimização apenas dos termos entre chaves da equação (5.8).

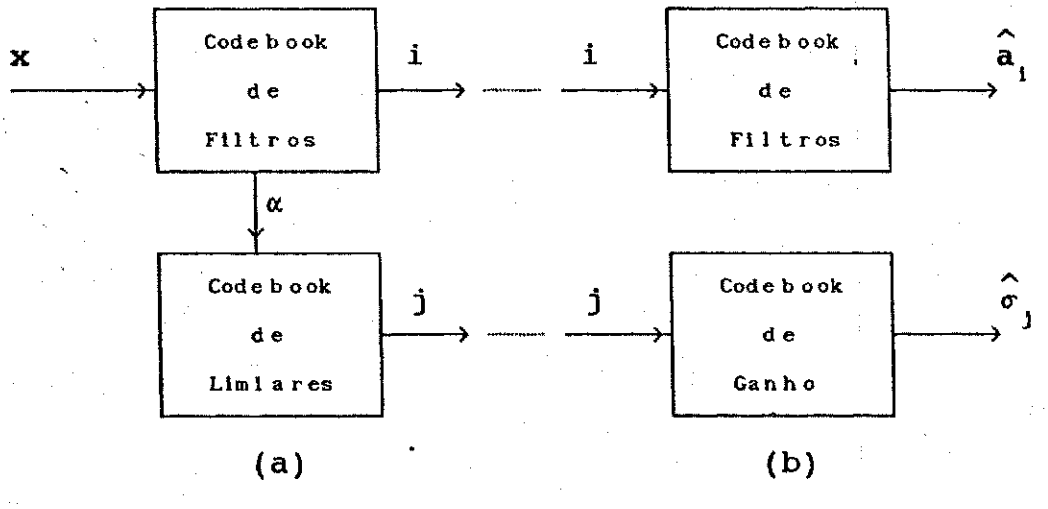


Figura 5.1 - Esquema geral em blocos de um quantizador filtro-ganho para codificação de voz por predição linear. (a) Codebooks codificadores de filtros e ganho; (b) Codebooks decodificadores.

Como mostra a figura 5.1, que apresenta o diagrama de blocos completo de um quantizador filtro-ganho, a procura do modelo mais semelhante ao modelo de entrada consiste em encontrar-se o par (i,j) que minimize o termo entre chaves de (5.8). Isto pode ser feito em dois passos:

1) Selecione o vetor $\hat{\mathbf{a}}_i$ que minimize $\hat{\mathbf{a}}_i \mathbf{R}(\mathbf{x}) \hat{\mathbf{a}}_i^t$.

2) Dado $\alpha = \min \hat{\mathbf{a}}_i \mathbf{R}(\mathbf{x}) \hat{\mathbf{a}}_i^t$, selecione o ganho $\hat{\sigma}_j$ que minimize $\alpha/\hat{\sigma}_j^2 + \ln(\hat{\sigma}_j^2)$.

Com isso, o passo (1) pode ser visto como a procura de um filtro inverso em um conjunto de filtros inversos, que produz a menor energia residual quando aplicado ao quadro de voz ou vetor de entrada. O passo (2) realiza uma quantização escalar do valor α que é a energia residual mínima. O cálculo da energia residual $\hat{\mathbf{a}}_i \mathbf{R}(\mathbf{x}) \hat{\mathbf{a}}_i^t$ pode ser realizado pela equação (3.46) que é expressa por

$$\hat{\mathbf{a}}_i \mathbf{R}(\mathbf{x}) \hat{\mathbf{a}}_i^t = r_{\mathbf{x}}(0) r_{\hat{\mathbf{a}}_i}(0) + 2 \sum_{n=1}^{M} r_{\mathbf{x}}(n) r_{\hat{\mathbf{a}}_i}(n) \qquad (5.9)$$

onde,

$$r_{\hat{\mathbf{a}}_i}(n) = \sum_{k=0}^{M-n} \hat{a}_{i,k} \hat{a}_{i,k+n} \qquad (5.10)$$

e para melhor eficiência computacional, no "codebook" transmissor, cada filtro inverso pode ser armazenado como $\{r_{\hat{\mathbf{a}}_i}(0), 2r_{\hat{\mathbf{a}}_i}(1), \cdots, 2r_{\hat{\mathbf{a}}_i}(M)\}$.

A seleção do ganho $\hat{\sigma}_j$ para minimizar $\alpha/\hat{\sigma}_j^2 + \ln(\hat{\sigma}_j^2)$ é realizada comparando a energia residual α com um conjunto de limiares. Assim, se o "codebook" de ganho contém os escalares $\{\hat{\sigma}_j^2;\ j=1,2,\cdots,N_2\}$, com

$$\hat{\sigma}_1^2 < \hat{\sigma}_2^2 < \hat{\sigma}_3^2 < \cdots < \hat{\sigma}_{N_2}^2,$$

então, para este "codebook", deve-se encontrar um conjunto de limiares $\{\beta_j^2;\ j=1,2,\cdots,N_2-1\}$ de modo que

$$\hat{\sigma}_1^2 < \beta_1^2 < \hat{\sigma}_2^2 < \beta_2^2 < \cdots < \beta_{N_2-1}^2 < \hat{\sigma}_{N_2}^2. \qquad (5.11)$$

No codificador, é armazenado um conjunto de (N_2-1) limiares, e a energia residual α é comparada com cada limiar para se determinar a célula em que ela se encontra e gerar o índice para transmissão. Por exemplo, se $\alpha \leq \beta_1^2$, então o índice j = 1 é transmitido, ou se $\beta_6^2 < \alpha \leq \beta_7^2$ o índice j = 7 é transmitido. No decodificador, o índice é usado para recuperar o ganho $\hat{\sigma}_j^2$. De [25] o limiar de ganho entre $\hat{\sigma}_j^2$ e $\hat{\sigma}_{j+1}^2$ é expresso por

$$\beta_j^2 = \frac{\ln(\hat{\sigma}_{j+1}^2 / \hat{\sigma}_j^2)}{(1/\hat{\sigma}_j^2)-(1/\hat{\sigma}_{j+1}^2)} \qquad (5.12)$$

Quando $\hat{\sigma}_{j+1}^2$ é muito próximo de $\hat{\sigma}_j^2$, como no caso de uma quantização fina, então a equação (5.12) não é eficiente numericamente devido à subtração no seu denominador. Uma alternativa que é muito sugerida é utilizar a expansão em série de Taylor expressa por

$$\beta_j^2 = (\hat{\sigma}_j^2 + \hat{\sigma}_{j+1}^2)\left[\frac{1}{2} - \frac{\delta^2}{3} - \frac{\delta^4}{15} - \frac{\delta^6}{35} - \cdots\right] \qquad (5.13)$$

onde

$$\delta \triangleq (\hat{\sigma}_{j+1}^2 - \hat{\sigma}_j^2)/(\hat{\sigma}_{j+1}^2 + \hat{\sigma}_j^2). \qquad (5.14)$$

A medida de distorção d_{IS}^M na equação (5.9) pode também ser escrita como

$$\left[\ln\left(\frac{\hat{a}_i R(x)\hat{a}_i^t}{\sigma^2(M)}\right)\right] + \left[\frac{\hat{a}_i R(x)\hat{a}_i^t}{\hat{\sigma}_j^2} - \ln\left(\frac{\hat{a}_i R(x)\hat{a}_i^t}{\hat{\sigma}_j^2}\right) - 1\right]. \qquad (5.15)$$

O termo do lado esquerdo entre colchetes não depende de $\hat{\sigma}_j$ e é chamado distorção de ganho otimizado ou distorção de Itakura [26,27]; o termo do lado direito é um valor não negativo e igual a

zero se e somente se $\hat{\sigma}_j^2 = \hat{a}_i R(x) \hat{a}_i^t$. Assim, o termo direito é a contribuição de distorção quando o ganho $(\hat{a}_i R(x) \hat{a}_i^t)^{1/2}$ é quantizado por $\hat{\sigma}_j$. Então a distorção de Itakura-Saito modificada pode ser representada por

$$d_{IS}^M(x;y_{ij}) = d_{IS}^M(x;\hat{a}_i,\hat{\sigma}_j) = d'(x;\hat{a}_i) + d''\left(\sigma^*(x,\hat{a}_i);\hat{\sigma}_j\right) \qquad (5.16)$$

onde d' é a distorção de ganho otimizado referida também como distorção de filtro e d'' é a contribuição devido à quantização de $\sigma^*(x,\hat{a}_i)$ por $\hat{\sigma}_j$, referida também como distorção de ganho.

5.4.1 - OTIMIZAÇÃO CONJUNTA

Seja $\hat{q}(x)$ um quantizador arbitrário do espaço de entrada em um produto de "codebooks" $\hat{A} \times \hat{\mathcal{H}}$. Sejam as partições $S = \{S_{ij};\ i=1,2,\cdots,N_1$ e $j=1,2,\cdots,N_2\}$ onde $S_{ij} = \{x \in \mathcal{R}^K;\ \hat{q}(x) = (\hat{a}_i,\hat{\sigma}_j)\}$. Seja $P_i = \{\bigcup_j S_{ij};\ j=1,2,\cdots,N_2\}$, a célula do espaço de entrada que mapeia no vetor $\hat{a}_i$ e $Q_j = \{\bigcup_i S_{ij};\ i=1,2,\cdots,N_1\}$, a célula do espaço de entrada que mapeia no ganho $\hat{\sigma}_j$. Sejam também o mapeamento de filtros $\hat{a}(x) = \hat{a}_i$ para $x \in P_i$ e o mapeamento de ganho $\hat{\sigma}(x) = \hat{\sigma}_j$ para $x \in Q_j$. Denotar-se-á a distorção esperada de um quantizador filtro-ganho $\hat{A} \times \hat{\mathcal{H}}$ e uma partição S por $D(\hat{A},\hat{\mathcal{H}},S)$, já que ela depende tanto do "codebook" de filtros como do "codebook" de ganho e da partição S.

Dado um quantizador filtro-ganho, de acordo com a regra de mapeamento de distorção mínima, pode-se escrever

$$D(\hat{A},\hat{\mathcal{H}},S) = E\left\{d\left(X;\hat{a}(X),\hat{\sigma}(X)\right)\right\}$$

$$\geq E\left\{\min_{ij} d(X;\hat{a}_{ij},\hat{\sigma}_j)\right\}. \qquad (5.17)$$

Seja $S^+(\hat{A},\hat{\mathcal{H}})$ uma partição de distorção mínima de $\mathcal{R}^K$ para $\hat{A} \times \hat{\mathcal{H}}$, isto é, $S^+(\hat{A},\hat{\mathcal{H}}) = \{S_{ij}^+,\ i=1,2,\cdots,N_1$ e $j=1,2,\cdots,N_2\}$ onde $d(x;\hat{a}_i,\hat{\sigma}_j) \leq d(x;\hat{a}_k,\hat{\sigma}_l)$ para $x \in S_{ij}^+$. Como no Capítulo 4, a partição de distorção mínima não é necessariamente única, mas pode-se impor um critério de parada arbitrário, e parar o processo

obtendo partições localmente ótimas. De (5.17) tem-se que

$$D(\hat{A},\hat{\mathcal{H}},S) \geq D\left(\hat{A},\hat{\mathcal{H}},S^{+}(\hat{A},\hat{\mathcal{H}})\right). \qquad (5.18)$$

A equação (5.18) mostra que se consegue melhorar o quantizador $q(\cdot)$, otimizando-se a partição em relação aos dois alfabetos.

Se a otimização é feita em relação ao "codebook" de filtros e a distorção mínima é atingida por $u_i^* \in A$ para cada $i=1,2,\cdots,N_1$, pode-se definir $\hat{A}^{+} = \{u_i^*;\ i=1,2,\cdots,N_1\}$ com $\hat{A}^{+}$ dependendo apenas das partições de S e do "codebook" de ganho $\hat{\mathcal{H}}$, sendo representado por $\hat{A}^{+}(S,\hat{\mathcal{H}})$. A desigualdade na equação (5.18) pode então ser expressa por

$$D(\hat{A},\hat{\mathcal{H}},S) \geq D\left(\hat{A}^{+}(\hat{\mathcal{H}},S),\hat{\mathcal{H}},S\right). \qquad (5.19)$$

A equação (5.19) indica que se pode melhorar o quantizador $q(\cdot)$, otimizando-se o alfabeto de filtro em relação ao alfabeto de ganho e à partição.

Analogamente, em relação ao ganho, se a minimização da distorção média total é obtida em $\lambda_j^* \geq 0$ para $i=1,2,\cdots,N_2$, pode-se definir $\hat{\mathcal{H}}^{+} = \{\lambda_j^*;\ j=1,2,\cdots,N_2\}$. Neste caso, a dependência é em relação à partição S e ao "codebook" de filtros $\hat{A}^{+}$ e escreve-se $\hat{\mathcal{H}}^{+}(\hat{A},S)$. A desigualdade pode agora ser expressa por

$$D(\hat{A},\hat{\mathcal{H}},S) \geq D\left(\hat{A},\hat{\mathcal{H}}^{+}(\hat{A},S),S\right). \qquad (5.20)$$

Já com esta equação, pode-se melhorar o quantizador otimizando-se o alfabeto de ganho em relação ao alfabeto de filtros e à partição.

As desigualdades em (5.18), (5.19) e (5.20) sugerem passos pelos quais o quantizador pode ser melhorado iterativamente. O seguinte algoritmo usa esses passos para otimizar o quantizador filtro-ganho:

Algoritmo de Otimização Conjunta

1) Inicialização: Dado N_1, N_2, $\varepsilon \geq 0$, um produto de "codebooks" inicial $\mathcal{Z}_0 = (\hat{A}_0 \times \hat{\mathcal{H}}_0)$ e uma distribuição $F(\mathbf{x})$. Faça $m=0$, $D_{-1} = \infty$;

2) Calcule as partições ótimas $S^{+}(\hat{A}_m,\hat{\mathcal{H}}_m)$;

3) Calcule a distorção média $D_m = D(\hat{A}_m, \hat{\mathcal{H}}_m, S^*(\hat{A}_m, \hat{\mathcal{H}}_m))$. Se $\{(D_{m-1} - D_m)/ D_m\} \leq e$, pare com o quantizador final descrito por $(\hat{A}_m, \hat{\mathcal{H}}_m, S^*(\hat{A}_m, \hat{\mathcal{H}}_m))$. Caso contrário, continue;

4) Calcule o alfabeto ótimo de filtros $\hat{A}_{m+1} = \hat{A}(\hat{\mathcal{H}}_m, S^*(\hat{A}_m, \hat{\mathcal{H}}_m))$ e a partição ótima $S^*(\hat{A}_{m+1}, \hat{\mathcal{H}}_m)$;

5) Calcule o alfabeto ótimo de ganho $\hat{\mathcal{H}}_{m+1} = \hat{\mathcal{H}}_m(\hat{A}_{m+1}, S^*(\hat{A}_m, \hat{\mathcal{H}}_m))$, faça m = m+1 e retorne ao passo (2).

5.4.2 - OTIMIZAÇÃO SEPARADA: ALGORITMO BGGM

O algoritmo BGGM (Buzo, Gray, Jr., Gray, e Markel) [25] é visto como um caso particular de um quantizador "product code". Neste algoritmo, o quantizador é projetado otimizando os dois "codebooks" separadamente. Primeiro é projetado o "codebook" de filtros usando o algoritmo LBG com a distorção d' das equações (5.15) e (5.16). Então, com $\hat{\mathbf{a}}(\mathbf{x})$ representando o mapeamento ótimo do espaço de entrada em um alfabeto de filtros, é projetado o "codebook" de ganho aplicando também o algoritmo LBG com a distorção d" e com o ganho s^* obtido entre o quadro de voz original e o vetor quantizado, como na equação (5.16). Como o alfabeto de filtros é projetado independentemente do alfabeto de ganho, supõe-se que o par filtro-ganho não seja ótimo no sentido da minimização da distorção total d. Entretanto, este método tem mostrado bons resultados em codificação de voz por predição linear [25,26].

5.4.3 - OTIMIZAÇÃO INDIVIDUAL

Seja $\hat{\mathbf{a}}(\mathbf{x})$ um mapeamento arbitrário do espaço de entrada em um alfabeto de filtros $\hat{A}$. Seja a partição de filtros $\mathcal{P} = \{P_i; i=1,2,\cdots,N_1\}$ onde $P_i = \{\mathbf{x} \in \mathcal{R}^K: \hat{\mathbf{a}}(\mathbf{x})=\hat{\mathbf{a}}_i\}$. Para o quantizador de filtros descrito por $(\hat{A},\mathcal{P})$, denota-se a distorção média de filtros por $D'(\hat{A},\mathcal{P})$. Como na equação (5.17), observe que

$$D'(\hat{A},\mathcal{P}) = E\left[d'(X;a(\mathbf{x}))\right] \geq E\left[\min_i d'(X;\hat{\mathbf{a}}_i)\right]. \qquad (5.21)$$

Seja a partição ótima de filtros $\mathcal{P}^+(\hat{A}) = \{P_i^+ ; i=1,2,\cdots,N_1\}$ onde $d'(\mathbf{x};\hat{\mathbf{a}}_i) \leq d'(\mathbf{x};\hat{\mathbf{a}}_k)$ para $\mathbf{x} \in \mathcal{P}^+$. Assim,

$$D'(\hat{A},\mathcal{P}) \geq D'\left(\hat{A},\mathcal{P}^+(\hat{A})\right). \qquad (5.21)$$

A equação (5.21) indica que se pode melhorar o alfabeto de filtros do quantizador $q(\cdot)$, otimizando-se apenas a partição em relação ao próprio alfabeto.

Com o ganho ótimo σ^* dado como na equação (5.16) e se a distorção mínima é atingida por algum $\mathbf{u}_i^*$ em $\hat{A}$, então pode-se definir $\hat{A}^+ = \{\mathbf{u}_i^*; i=1,2,\cdots,N_1\}$. Como $\hat{A}^+$ depende da partição $\mathcal{P}$ e do ganho σ^*, que por sua vez volta a depender de $\hat{A}$ (esta dependência será abordada na determinação dos centróides no item 5.4.4), escreve-se então $\hat{A}^+(\mathcal{P},\hat{A})$. Assim, a desigualdade na equação (5.30) pode ser expressa por

$$D'(\hat{A},\mathcal{P}) \geq D'\left(\hat{A}^+(\mathcal{P},\hat{A}),\mathcal{P}\right). \qquad (5.23)$$

A equação (5.23) mostra que se pode melhorar um quantizador $q(\cdot)$, otimizando-se o alfabeto de filtros em relação a ele mesmo e à partição.

As desigualdades (5.22) e (5.23) fornecem passos pelos quais a parte relativa ao alfabeto de filtros de um quantizador "product code" pode ser otimizada. A equação (5.22) sugere a otimização da partição $\mathcal{P}$ para o "codebook" $\hat{A}$. Já a desigualdade (5.23) sugere a troca de $\hat{A}$ pelo alfabeto de filtros que é ótimo para a partição $\mathcal{P}$ e o mapeamento σ^* que é introduzido por $\mathcal{P}$ e $\hat{A}$. O seguinte algoritmo usa iterativamente estes passos para otimizar o alfabeto de filtros de um quantizador filtro-ganho.

Algoritmo de Otimização Individual

1) Inicialização: Dado N_1, $\varepsilon \geq 0$, um "codebook" de filtros inicial $\hat{A}_0$ e uma distribuição $F(\mathbf{x})$, faça $m=0$, $D_{-1} = \infty$;

2) Calcule as partições ótimas $\mathcal{P}^+(\hat{A}_m)$;

3) Calcule a distorção média de filtros $D'_m = D'(\hat{A}_m,\mathcal{P}^+(\hat{A}_m))$. Se

$\{(D'_{m-1} - D'_m)/D'_m\} \leq \varepsilon$, pare com o quantizador de filtros final descrito por $(\hat{A}_m, \mathcal{P}^+(\hat{A}_m))$. Caso contrário, continue;

4) Calcule o alfabeto ótimo de filtros $\hat{A}_{m+1} = \hat{A}^+(\mathcal{P}^+(\hat{A}_m), \hat{A}_m)$, faça m = m+1 e retorne ao passo (4).

Este algoritmo é idêntico à parte de otimização de filtros do algoritmo BGGM, exceto no passo (3). Aqui o "codebook" otimizado depende tanto do "codebook" corrente bem como da partição, ao contrário do BGGM onde a dependência é apenas da partição, conforme será esclarecido no item a seguir.

Como os três algoritmos são válidos para distribuições puramente discretas, então todos eles são úteis quando se desejam projetar quantizadores filtro-ganho baseados numa longa seqüência de treinamento $\{\mathbf{x}_m;\ m=1,2,\cdots,L\}$ de vetores obtidos do sinal de voz.

5.4.4 - DETERMINAÇÃO DOS CENTRÓIDES

Para atualizar o alfabeto de reprodução de filtro no passo (4) de cada algoritmo, é necessário encontrar o vetor $\mathbf{u}_i^*$ que minimiza a distorção condicional associada a cada célula de filtros P_i. Seguindo a terminologia do algoritmo LBG, chamar-se-ão os vetores $\mathbf{u}_i^*$ de centróides das células P_i. Note, entretanto , que $\mathbf{u}_i^*$ depende da célula P_i como também do mapeamento do ganho ($\hat{\sigma}$ ou $[\sigma^*]^+$ para otimizações conjunta ou individual respectivamente). Seja $P_i = \{\mathbf{x}_m;\ m=1,2,\cdots,NP_i\}$ o conjunto com NP_i vetores em P_i. Então, com relação à distorção d' e para um ganho arbitrário $\sigma(\mathbf{x})$, os centróides ou pontos de distorção mínima são os conjuntos de coeficientes obtidos, através do algoritmo de Levinson-Durbin, da matriz de autocorrelação média entre todas as matrizes de autocorrelação correspondentes a cada vetor na célula P_i, depois que são normalizadas por $\sigma^2(\mathbf{x})$. A matriz média é expressa por

$$\bar{\mathbf{R}}_i = \frac{1}{NP_i} \sum_{m=1}^{NP_i} \mathbf{R}_m/\sigma^2(\mathbf{x}), \qquad (5.24)$$

ou seja, o centróide é o modelo LPC do espectro médio normalizado

por $\sigma^2(\mathbf{x})$. Assim, quadros de baixa energia influenciarão no cálculo do centróide mais que os quadros de alta energia, já que o valor de $\sigma^2(\mathbf{x})$ é proporcional à energia do quadro de voz. Para o algoritmo de otimização conjunta, $\sigma(\mathbf{x}) = \hat{\sigma}(\mathbf{x}) = \hat{\sigma}_j$ para $\mathbf{x} \in Q_j$, que é o mapeamento de ganho para o vetor de entrada $\mathbf{x}$. Para o algoritmo de otimização individual, $\sigma(\mathbf{x}) = [\sigma^*(\mathbf{x},\hat{\mathbf{a}}_i(\mathbf{x}))]^+$ para a célula P_i, que é calculado a partir do filtro quantizado. E para o algoritmo BGGM, $\sigma(\mathbf{x}) = \sigma_M(\mathbf{x})$, que é ganho ótimo resultante da energia residual calculada a partir do filtro ótimo (sem quantização).

Para atualizar o alfabeto de reprodução de ganho no passo (5) do algoritmo de otimização conjunta, é necessário encontrar os valores λ_j^* que minimizam a distorção associada a cada célula Q_j. Referir-se-á a λ_j^* como o centróide da célula Q_j e que depende tando do mapeamento de filtros $\hat{\mathbf{a}}(\mathbf{x})$ como de Q_j. Seja $Q_j = \{\mathbf{x}_m; m=1,2,\cdots,NQ_j\}$ o conjunto de vetores em Q_j. Então, com relação à distorção d", os centróides ou pontos de distorção mínima são os escalares expressos por

$$\lambda_j^* = \left(\frac{1}{NQ_j} \sum_{m=1}^{NQ_j} \alpha_m \right)^{1/2}, \tag{5.25}$$

onde $\alpha_m = \hat{\mathbf{a}}(\mathbf{x})\mathbf{R}_m(\mathbf{x})\hat{\mathbf{a}}^t(\mathbf{x})$ é a energia residual calculada pela passagem do sinal de voz $\mathbf{x}$ pelo filtro $\hat{\mathbf{a}}(\mathbf{x})$. Para o cálculo do centróide na otimização do alfabeto de reprodução de ganho do algoritmo BGGM, é utilizado a mesma expressão da equação (5.25) lembrando-se apenas que a otimização do alfabeto de ganho é realizada separadamente da otimização do alfabeto de filtros. A derivação dos centróides para estes algoritmos é apresentada no Apêndice A. Dos três algoritmos "product code", o de otimização conjunta tem apresentado desempenho melhor que o de otimização individual, que por sua vez apresentou desempenho superior ao de otimização separada (BGGM).

5.5 - TRATAMENTO DE CÉLULAS VAZIAS

Durante a execução dos algoritmos de projeto de alfabetos de

reprodução, podem sugir células com probabilidade zero (células vazias). Regras alternativas para a eliminação dessas células são possíveis e, na prática, podem até superar o procedimento onde se assinala um vetor arbitrário para essas células. A seguir são apresentadas as alternativas sugeridas em [24]:

a) Pode-se simplesmente remover a célula vazia S_i e o vetor de reprodução correspondente e continuar o algoritmo com um quantizador de (N-1) níveis sem afetar o desempenho;

b) Pode-se ainda designar para S_i , o centróide da célula S_{i-1} da iteração anterior ou de outra célula qualquer S_j não vazia;

c) Pode-se também, como alternativa mais simples, re-executar o algoritmo com condições iniciais diferentes.

A alternativa (a) não afeta o desempenho. Por outro lado, é subtraído um nível do quantizador. Na sugestão (b), a situação é análoga a (a) já que é obtido um quantizador com no mínimo dois níveis iguais, ficando assim com vetores de reprodução redundantes. A sugestão (c) seria a melhor entre as outras, não fosse o tempo muito alto de processamento na geração dos quantizadores, tornando o processo tedioso. A seguir é apresentado um método simples mas computacionalmente eficiente para tratamento de células vazias e que, nas várias simulações e experiências, tem mostrado resultados satisfatórios e inclusive produzindo mínimos locais melhores que os obtidos com as opções (a)-(c).

O Método

1) Faça C_i representar uma célula formada por todos os vetores de reprodução relacionados às células anteriores geradas durante a execução de qualquer algoritmo de projeto de alfabeto de reprodução, ou seja, $C_i = \{\hat{\mathbf{x}}(S_j)$; para todo $j < i\}$;

2) Se $Pr(X \in S_i) = 0$, então assinale para S_i o centróide da célula C_i, isto é,

$$\hat{\mathbf{x}}(S_i) = \text{cent}(C_i), \qquad (5.26)$$

onde o centróide $\text{cent}(C_i)$ é calculado de acordo com a medida de distorção utilizada no projeto do alfabeto.

Neste método, $\hat{\mathbf{x}}(S_i)$ é um vetor de distorção mínima em relação

ao vetores $\hat{x}(S_j)$ para $j < i$, já que as condições de otimização são satisfeitas. Assim, $\hat{x}(S_i)$ é o vetor que melhor representa os vetores $\hat{x}(S_j)$ e como os vetores $\hat{x}(S_j)$ são os vetores que melhor representam os vetores da seqüência de treinamento, a distorção média total deve decrescer ou permanecer inalterada. Para $Pr(X \in S_i) = 0$ no nível $i = 2$, reduz-se ao caso sugerido em (b). Alguns resultados referentes a este método são apresentados no Capítulo 6.

5.6 - ALFABETOS INICIAIS

Todos os algoritmos de geração de "codebooks", apresentados no itens anteriores, requerem um "codebook" inicial. Para a geração do alfabeto inicial são propostos dois procedimentos básicos: Um inicia com o alfabeto já com tamanho exato e o outro produz recursivamente alfabetos maiores a partir de alfabetos menores. Este item apresenta algumas técnicas relacionadas a estes dois procedimentos.

Alfabetos Aleatórios

Este é o método de geração de "codebooks" mais simples e consiste em se escolher, aleatoriamente, uma quantidade de vetores da seqüência de treinamento correspondente ao tamanho exato do "codebook" a ser otimizado. Para dados altamente correlacionados, uma opção natural é selecionar vários vetores largamente espaçados da seqüencia de treinamento. Outra alternativa é selecionar os vetores da seqüência que fornecerem maior distorção entre si.

Alfabetos "Product Code"

O nome adotado para este método se confunde com nome do método de projeto de quantizadores filtro-ganho. Como no item 5.5., o nome "product code" provém justamente do fato de que o método utiliza o produto cartesiano, só que para produzir alfabetos iniciais.

Este método parte de "codebooks" menores tanto em dimensão como em número de níveis e pode ser modelado como segue: Seja $\{A_i;$

$i=1,2,\cdots,P\}$ uma coleção de codebooks individuais, cada um consistindo de M_i vetores de dimensão K_i (ou escalares se $K_i = 1$) com taxa $R_i = \log_2(M_i)$ bits/vetor. Assim, o alfabeto "product code" $\mathcal{A}$ é definido como a coleção de todas as $M=\prod_i M_i$ combinações possíveis dos P vetores ou escalares, obtidos sucessivamente dos P "codebooks" $\mathcal{A}_i$. A dimensão do "producto code" é $K = \sum_{i=1}^{P} K_i$, a soma das dimensões dos "codebooks" individuais. O "product code" pode ser representado matematicamente como um produto cartesiano [3], ou seja,

$$\mathcal{A} = \bigtimes_{i=1}^{P} \mathcal{A}_i .$$

Para melhor compreensão deste método, considere-se o seguinte exemplo: Sejam os "codebooks" de escalares $\mathcal{A}_1 = \{a_j;\ j=1,\cdots,4\}$ e $\mathcal{A}_2 = \{b_m;\ m=1,\cdots,4\}$ com a_j e $b_m \in \mathcal{R}$. Para este caso particular de "product code", tem-se que $K_1 = K_2 = 1$, $M_1 = M_2 = 4$ e $R_1 = R_2 = 2$, com o "codebook" final $\mathcal{A}$ dado por

$$\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 = \left\{(a_j, b_m);\ 1 \le j,m \le 4\right\}.$$

Então $\mathcal{A}$ tem $M = M_1 M_2 = 16$ vetores com dimensão $K = K_1 + K_2 = 2$ e conseqüentemente $R = 4$ bits/vetor.

Assim, usando-se K vezes o produto cartesiano em um quantizador escalar com taxa R/K bits por dimensão, produz-se um quantizador de taxa R bits/vetor e com vetores K-dimensionais.

Divisões Sucessivas (Splitting)

Esta técnica é baseada no segundo procedimento, ou seja, ela produz alfabetos maiores a partir de alfabetos menores mas apenas em relação ao número de níveis. O método consiste nos seguintes passos: Primeiro, encontra-se o alfabeto ótimo com um nível, que é o centróide de toda seqüência de treinamento. Através de uma pertubação no centróide calculado, é produzido um alfabeto com dois níveis; isto pode ser visto como a divisão do vetor-centróide inicial em dois vetores. Após a divisão, o alfabeto com dois níveis

é otimizado pelo algoritmo LBG produzindo dois vetores-centróide. Novamente, estes dois vetores-centróide são pertubados, produzindo um alfabeto com quatro níveis e este novo alfabeto é otimizado pelo algoritmo LBG. Sucessivamente, o algoritmo é repetido até que se atinja o número de níveis desejado. A figura 5.2 ilustra os passos deste algoritmo para a otimização de um quantizador com quatro níveis ou taxa dois bits/vetor.

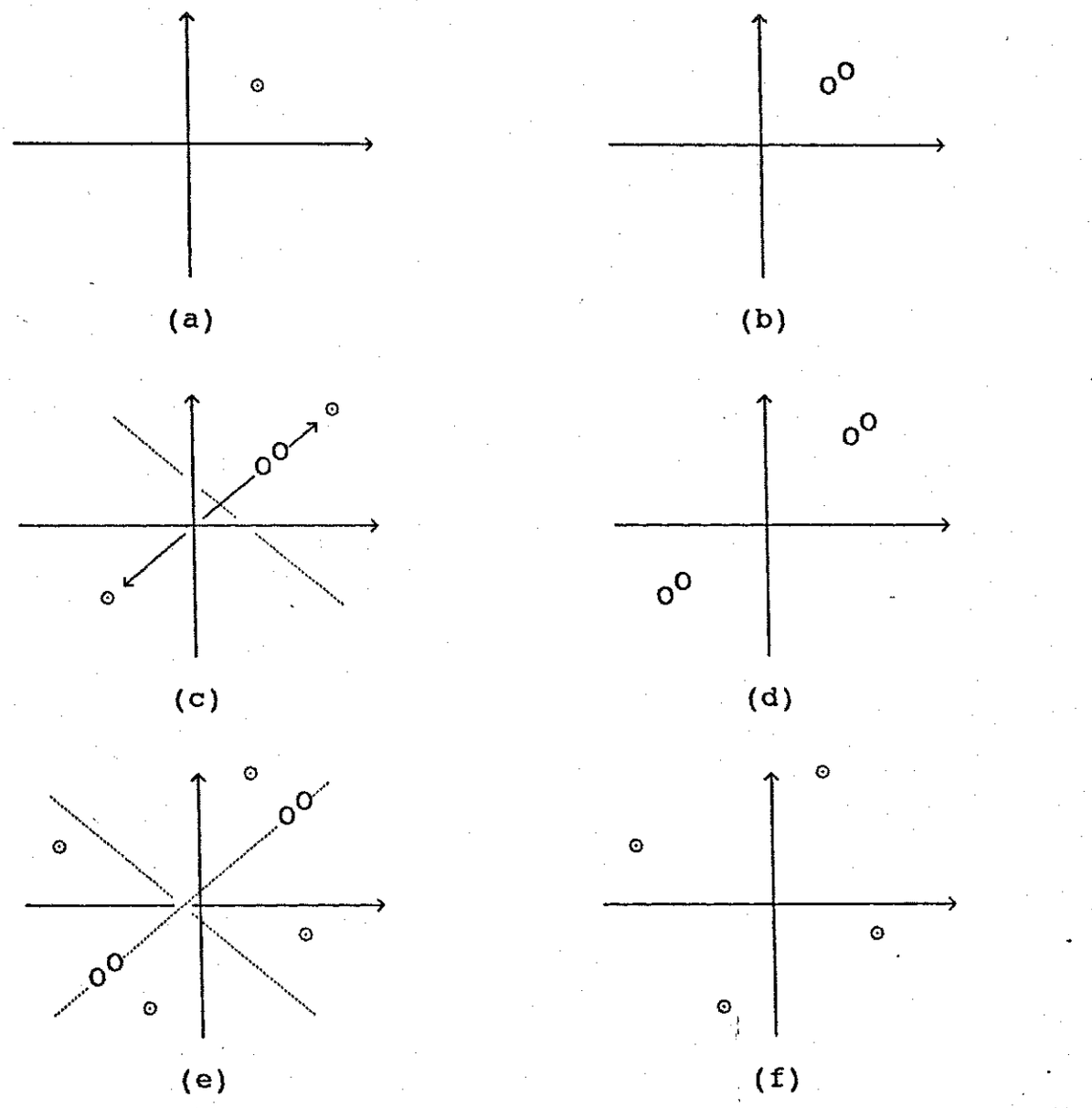


Figura 5.2 - Divisões sucessivas. "⊙" representa o centróide ou vetor de distorção mínima e "O" representa o vetor após a pertubação, antes de ser otimizado. (a) Alfabeto ótimo de taxa zero

(centróide da seqüência de treinamento); (b) Alfabeto de taxa um, após a divisão do centróide inicial; (c) Alfabeto de taxa um otimizado; (d) Alfabeto de taxa dois após a divisão de cada vetor em dois; (e) Alfabeto ótimo de taxa dois ilustrando os vetores pertubados; (f) Alfabeto ótimo final com quatro níveis (taxa dois).

O algoritmo considera os quantizadores de M níveis com $M = 2^R$, $R = 0,1,2,\cdots$, e continua até obter um "codebook" inicial para o quantizador de N níveis. Os seguintes passos descrevem o método de divisões sucessivas:

1) Inicialização: Faça M=1 e defina $\hat{A}_0(1) = \hat{\mathbf{x}}(A)$ como o centróide de toda seqüência de treinamento;

2) Dado um alfabeto de reprodução $\hat{A}_0(M)$ contendo M vetores $\{\mathbf{y}_i;\ i=1,\cdots,M\}$, pertube cada vetor $\mathbf{y}_i$ de maneira a obter-se dois vetores próximos $\mathbf{y}_i+\mathbf{p}$ e $\mathbf{y}_i-\mathbf{p}$, onde p é um vetor de pertubação fixo. Assim, produz-se um alfabeto $\tilde{A}$ com 2M vetores. Faça M=2M;

3) Se M = N, faça $\hat{A}_0 = \tilde{A}(M)$ e pare com $\hat{A}_0$ sendo o alfabeto inicial para o quantizador com N níveis. Caso contrário, execute o algoritmo LBG para o quantizador com M níveis para otimizar $\tilde{A}(M)$ e produzir um alfabeto de reprodução inicial $\hat{A}_0(M)$ ótimo. Retorne ao passo (2).

Para melhor eficiência computacional, o passo (2) pode ser substituído pelo seguinte passo:

2) Dado um alfabeto de reprodução $\hat{A}_0(M)$ contendo M vetores $\{\mathbf{y}_i;\ i=1,\cdots,M\}$, pertube cada vetor $\mathbf{y}_i$ de maneira a obter-se dois vetores próximos $\mathbf{y}_i$ e $(d \circ \mathbf{y}_i)$, onde d é um escalar de pertubação fixo. Assim, produz-se um alfabeto $\tilde{A}$ com 2M vetores. Faça M=2M. d é uma constante que dependerá da percentagem de pertubação inicial desejada. Por exemplo, para uma pertubação inicial de 1%, tem-se que d = 1.01. Com essa modificação, o vetor original é repetido para assegurar que a distorção total não aumente.

5.7 - MÉTODOS DE BUSCA

A figura 5.3 ilustra o diagrama básico de um quantizador vetorial.

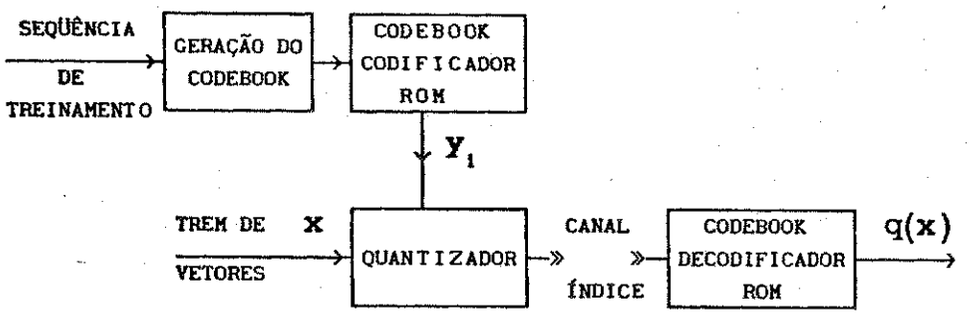


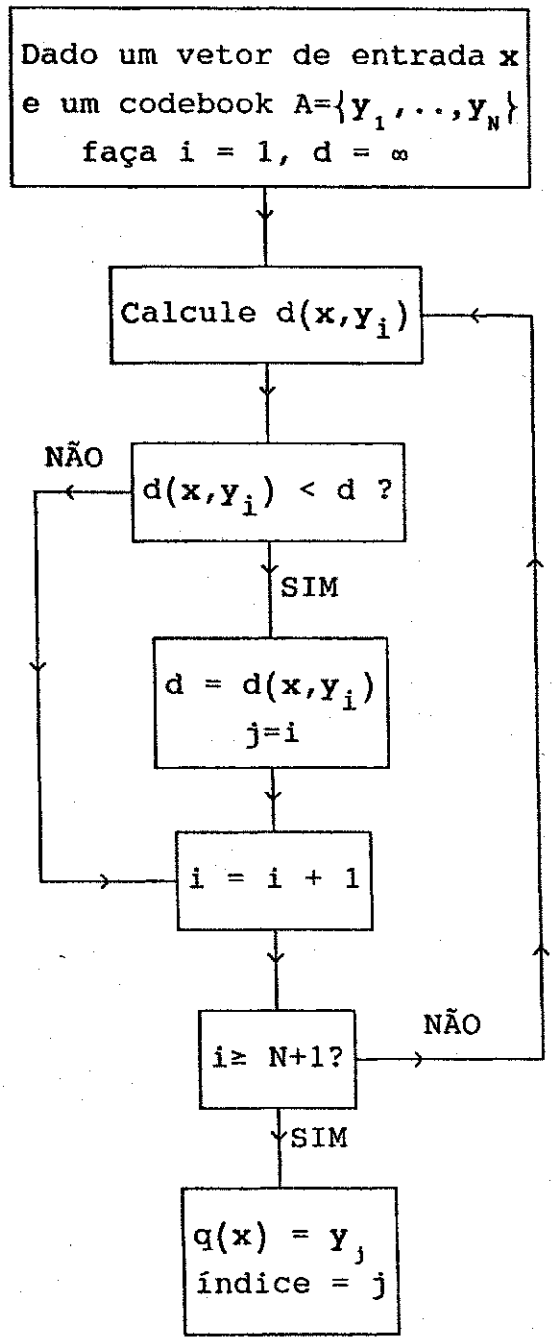
Figura 5.3 - Diagrama básico de um Quantizador Vetorial.

Os métodos de geração de "codebook" foram apresentados nos itens anteriores. No "codebook" codificador, através de um método de busca, é selecionado o vetor mais próximo ou mais semelhante ao vetor de entrada e transmitido o índice deste vetor. Este índice é utilizado para a recuperação do vetor de reprodução no "codebook" decodificador. Há vários métodos para a seleção do vetor ótimo no "codebook" codificador, no entanto serão apresentados apenas os algoritmos de busca plena e busca em árvore.

5.7.1 - BUSCA PLENA

No método de busca plena ou "full search", é calculada exaustivamente a distorção entre o vetor de entrada e todos os vetores de reprodução. É então selecionado o vetor de reprodução que apresentar a menor distorção e o índice correspondente é transmitido. O seguinte fluxograma apresenta os passos do método de busca plena:

Fluxograma do Método de Busca Plena



Para um quantizador K-dimensional de N níveis, o número de cálculos de distorção para cada vetor de entrada é N. Como há vários tipos de medidas de distorção, assume-se que para cada cálculo de distorção é necessário um total de K

multiplicações/adições (isto é verdade para a distorção de erro quadrático e uma versão da distorção de Itakura-Saito). Sendo assim, o custo computacional C_c para quantizar cada vetor de entrada é

$$C_c = N\,K. \tag{5.27}$$

Se cada vetor de reprodução é codificado em $B = R.K = \log_2 N$ bits para transmissão, então

$$C_c = K.2^{RK} \tag{5.28}$$

onde R é o número de bits por dimensão. Assim o custo computacional cresce exponencialmente com a dimensão e com o número de bits por dimensão.

Outro custo importante na quantização é o custo de armazenagem. Assumindo uma locação de memória por parâmetro de um vetor como medida de armazenagem, o custo de armazenagem C_A é expresso por

$$C_A = K.N = K.2^{RK} \tag{5.29}$$

Igualmente ao custo computacional, o custo de armazenagem também cresce exponencialmente com o a dimensão e com o número de bits por dimensão.

5.7.2 - BUSCA EM ÁRVORE

Para contornar o problema da complexidade computacional nos quantizadores vetoriais de busca plena, reduzindo o número de cálculos necessários para encontrar o vetor de reprodução ótimo, Buzo *etti ali* [25] propôs o algoritmo "tree search" ou busca em árvore. Neste método, o "codebook" é organizado em forma de árvore durante o seu processo de geração, particionando-se o espaço de tal maneira que a busca pelo vetor de reprodução de distorção mínima exija a comparação com $\log_2 N$ vetores em vez de N vetores. Para o exemplo de uma árvore binária uniforme, o espaço K-dimensional é

dividido em dois subespaços usando-se o método de divisões sucessivas; então cada um dos subespaços é dividido em dois subespaços adicionais. Este processo continua até que o espaço seja dividido em N subespaços, atingindo assim o número de níveis desejado. Como mostra a figura 5.4, exemplo de um caso com N = 8, há um centróide associado a cada divisão binária.

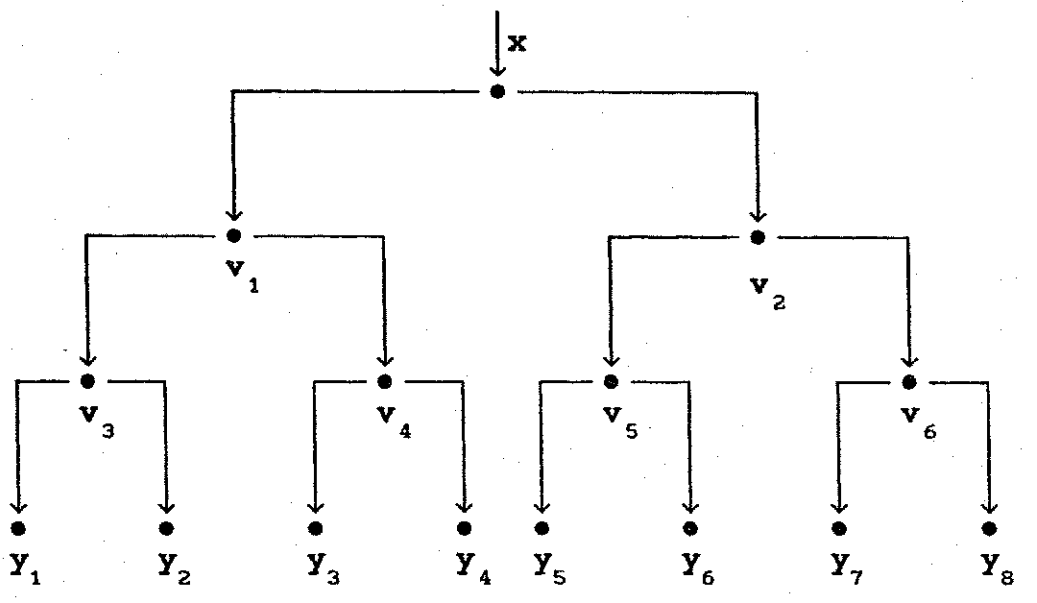


Figura 5.4 - Árvore binária uniforme. Os vetores $\mathbf{v}_i$ são vetores de referência intermediários que são comparados com o vetor de entrada **x**. Os vetores $\mathbf{y}_1$ são os vetores finais de reprodução.

Na primeira divisão binária, $\mathbf{v}_1$ e $\mathbf{v}_2$ são os centróides dos subespaços. Na segunda divisão, há quatro subespaços com centróides de $\mathbf{v}_3$ a $\mathbf{v}_6$. Após a terceira divisão, os centróides de cada subespaço são os vetores de reprodução.

O vetor de entrada é quantizado buscando-se ao longo da árvore, o caminho que fornecer menor distorção em cada nó. Assim **x** é comparado com $\mathbf{v}_1$ e $\mathbf{v}_2$. Se $d(\mathbf{x},\mathbf{v}_1) < d(\mathbf{x},\mathbf{v}_2)$, por exemplo, então o caminho para $\mathbf{v}_1$ é escolhido. Depois, **x** é comparado com com $\mathbf{v}_3$ e $\mathbf{v}_4$. Por exemplo, se $d(\mathbf{x},\mathbf{v}_4) < d(\mathbf{x},\mathbf{v}_3)$ então o caminho para $\mathbf{v}_4$ é selecionado. Finalmente, **x** é comparado com $\mathbf{y}_3$ e $\mathbf{y}_4$ e se $d(\mathbf{x},\mathbf{y}_3) < d(\mathbf{x},\mathbf{y}_4)$, por exemplo, então $\mathbf{y}_3$ é escolhido como o valor quantizado de **x**, ou seja, $q(\mathbf{x}) = \mathbf{y}_3$. Novamente, assumindo um total de K multiplicações/adições para o cálculo da distorção, tem-se um custo

computacional total de

$$C_C = 2\ K\ \log_2 N = 2\ K\ B, \qquad (5.30)$$

que é uma relação linear com o número de bits. Assim uma grande redução computacional pode ser obtida através da busca em árvore. Por outro lado, há um aumento no custo de armazenagem. Da figura 5.4, nota-se que, além de se armazenarem os vetores código $\mathbf{y}_i$, devem-se também armazenar os vetores intermediários $\mathbf{v}_i$. Com isso, o custo total de armazenagem é quase duplicado para

$$C_A = 2\ K\ (N-1). \qquad (5.31)$$

Outra desvantagem adicional é a redução no desempenho dos quantizadores com busca em árvore em relação ao quantizadores com busca plena [25]. Pode-se melhorar o desempenho dos quantizadores com busca em árvore utilizando árvores menos profunda que a binária e com mais desvios em cada nó. Para o exemplo de B = 12 bits, pode-se pensar na busca plena como um nível de desvio com 2^{12} desvios, ao passo que na busca binária tem-se 12 níveis de desvios com dois desvios em cada nó. Pode-se usar uma árvore quaternária com 6 níveis de desvio e 4 desvios em cada nó ($2^{12} = 4^6$) ou 4 níveis de desvio com 8 desvios em cada nó ($2^{12} = 8^4$), etc. A tabela 5.1 ilustra as diversas opções para o exemplo citado:

Número de Níveis de Desvio (p)	Número de Desvios em cada Nó (n)	Número de Cálculos de Distorção (np)
12	2	24 (busca binária)
6	4	24 (busca quaternária)
4	8	32
3	16	48
2	64	128
1	4096	4096 (busca plena)

Tabela 5.1 - Possíveis estruturas de árvores para B = 12 bits variando da busca binária à busca plena.

Se numa árvore há p níveis de desvio e n desvios em cada nível, o número total de níveis será expresso por

$$n^p = 2^B \qquad (5.40)$$

onde B é o número de bits para transmissão. O número total de cálculos de distorção em cada nível de desvio será n e como há p níveis, o total de cálculos de distorção será np.

Nota-se da tabela 5.1 que se pode conseguir melhor desempenho, aumentando o número de desvios em cada nó e sem aumentar o número de cálculos de distorção, através da utilização de uma árvore quaternária.

Existem ainda quantizadores vetoriais com busca em árvores não uniformes, onde em cada estágio o número de divisões em subespaços é diferente, mas estes tipos de quantizadores não serão adordados neste trabalho.

CAPÍTULO 6

SIMULAÇÕES, RESULTADOS E DISCUSSÕES

6.1 - INTRODUÇÃO

Nos Capítulos anteriores foram apresentados a descrição do "vocoder" utilizado, algumas medidas de distorção, os conceitos teóricos e os algoritmos para projetos de quantizadores vetoriais. Este Capítulo apresenta os quantizadores vetoriais implementados, os procedimentos utilizados na geração dos "codebooks" e nas simulações, os resultados objetivos e subjetivos obtidos. Também são apresentados os resultados referentes ao método de eliminação de células vazias.

6.2 - CRITÉRIOS DE DESEMPENHO

Como critérios de avaliação objetiva, utilizaram-se a medida de distorção média do quantizador (DM) e a relação sinal/ruído de quantização (SQNR) [4,23] que é expressa por

$$SQNR_{dB} = 10 \log_{10}\left(\frac{DM_0}{DM_B}\right), \qquad (6.1)$$

onde DM_B é a distorção média total de um quantizador vetorial com uma taxa de codificação de B bits/vetor e DM_0 é a distorção DM_B com B = 0, ou seja, é a distorção média total calculada em relação ao centróide único de toda a seqüência de treinamento. A distorção DM_B é uma função descrescente da taxa de bits B, fazendo com que a SQNR aumente com B.

A avaliação subjetiva consistiu da aplicação do método CJM (Category-Judgment Method) [8]. Neste método, a inteligibilidade e a qualidade do sinal de teste são definidas em termos de um determinado número de categorias que têm um significado intuitivo para os ouvintes. As categorias utilizadas são as seguintes: *excelente*, *boa*, *satisfatória*, *regular* e *péssima*. Os ouvintes utilizam estas categorias para indicarem suas impressões a respeito da qualidade e inteligibilidade do sinal de teste em relação a um sinal de referência.

Com o intuito de estabelecer um ponto de referência para a resposta dos ouvintes, utilizaram-se dois sinais de referência: um definido na categoria *boa* e outro definido na categoria *regular*. A avaliação foi feita utilizando-se o índice MOS (Mean Opnion Score) [8], onde para cada categoria é designada uma nota de ponderação da forma mostrada no anexo 1.

Feitas as avaliações, o número de ouvintes que escolhe cada categoria é multiplicado pelo peso da categoria correspondente. Os resultados são somados e o total é dividido pelo número de ouvintes. O número obtido é então o "score" de opinião para o sinal avaliado.

Utilizaram-se quatro sinais de teste. Os sinais de teste ST1 e ST2 foram obtidos a partir da frase:

"É necessário substituir o espírito literário da educação pelo espírito científico";

pronunciada respectivamente por um locutor feminino e um masculino. Já os sinais de teste ST3 e ST4 foram obtidos a partir da frase:

"Navegar é preciso, viver não. A natureza é a mãe da vida, respeite-a";

também pronunciada pelos mesmos locutores dos sinais ST1 e ST2. A voz do locutor feminino utilizada na geração dos sinais de testes também esteve presente na seqüência de treinamento. Já o locutor masculino não participou da seqüência de treinamento. Vale também observar que na seqüência de treinamento não foram utilizadas as frases dos sinais de testes nem segmentos dos mesmos. Os resultados objetivos e subjetivos são apresentados nos itens a seguir.

6.3 - QUANTIZADORES IMPLEMENTADOS

Basicamente foram implementados cinco quantizadores vetoriais de busca plena, sendo dois com o algoritmo LBG e três com o algoritmo "product code". Dos quantizadores implementados com o algoritmo LBG, um foi realizado com a distorção de erro quadrático e o outro com a versão 1 da distorção de Itakura-Saito. Com o algoritmo "product code", todos foram implementados utilizando-se a versão 2 da distorção de Itakura-Saito, diferenciando-se apenas no método de otimização utilizado. Todas as simulações foram realizadas com programas feitos especialmente para esta dissertação. As primeiras simulações foram realizadas num microcomputador PC/AT 286, mas o tempo de geração de um codebook de 8 bits (256 vetores), por exemplo, excedeu 36 horas. Devido a alta complexidade computacional, portanto, todos os programas foram transportados para uma estação de trabalho SUN SPARC (Apêndice C).

Todos os quantizadores foram implementados com a mesma seqüência de treinamento e simulados numa estação de trabalho SUN SPARC, cujas especificações são apresentadas no Apêndice C. A seqüência de treinamento foi obtida a partir do pronuncionamento de 13 locutores com os textos apresentados no Apêndice B. A seqüência foi então digitalizada no Sistema de Análise e Processamento Digital de Voz [7] e transmitida para a estação SUN SPARC. Nesta estação, foi gerado um único arquivo de coeficientes de autocorrelação, após a eliminação dos quadros de silêncio no início e no final de cada frase (vide Apêndice B), resultando um total de 15700 vetores.

O "vocoder LPC" implementado para os testes da quantização

vetorial apresenta as seguintes características:

Características do Vocoder LPC

- Ordem do filtro preditor: 8
- Janelamento: Janela de Hamming
- Intervalo de análise: 30 ms - 240 amostras
- Duração do quadro: 20 ms - 160 amostras
- Número de níveis de ruído aleatório: 3
- Coeficiente de pré-ênfase: 0.9
- Coeficiente de de-ênfase: 0.9

Em relação ao ganho do sinal de excitação, a geração dos quantizadores com a distorção de erro quadrático e com a distorção de Itakura-Saito 1 independe deste ganho . Já na quantização do sinal original, foi feita uma quantização da raiz quadrada da energia residual (α) e o ganho do sinal de excitação no sintetizador foi obtido através das equações (2.32) para sons sonoros e (2.34) para sons surdos.

Ainda em relação ao ganho, os quantizadores projetados com a distorção de Itakura-Saito 2 (product code) foram obtidos com a utilização direta da energia residual (α). Como nos algoritmos "product code" minimiza-se a energia residual entre o quadro de voz original e o quadro quantizado, e como a versão 2 da distorção de Itakura-Saito foi desenvolvida apenas com a energia residual, qualquer fator de proporcionalidade não constante no ganho pode afetar a medida de distorção e, conseqüentemente, pode afetar também a seleção do vetor ótimo de quantização. Entretanto, este é o procedimento utilizado na maioria dos projetos de "vocoder LPC". Já na síntese, o ganho foi obtido de um "codebook" de ganho com a introdução das constantes das equações (2.32) para sons sonoros e (2.34) para sons surdos (note que as constantes são diferentes).

Também foram implementados quantizadores de busca em árvore para os dois algoritmos (LBG e Product Code) e todas as medidas de distorção apresentadas. Entretanto, durante a geração dos alfabetos em árvore, surgiram várias células vazias, impossibilitando assim a

divisão binária e a continuação do algoritmo. Este problema ocorreu por duas razões: A primeira é que os métodos de divisão binária existentes não garantem células com uma quantidade de vetores mais ou menos igualitária entre si. A segunda foi devido à utilização de uma seqüência de treinamento não muito longa para a taxa de codificação desejada. Para um alfabeto em árvore de 6 ou 7 bits/vetor, os 15700 vetores da seqüência de treinamento foram suficientes, ou seja não surgiram células vazias para essas taxas. Já para uma taxa de codificação maior, condicionado ao método de divisão a ser utilizado, é preciso uma seqüência de treinamento duas ou três vezes maior que a utilizada neste trabalho.

6.3.1 - QUANTIZADOR VETORIAL 1 (QV1)

Este quantizador foi implementado com a distorção de erro quadrático e o algoritmo LBG, utilizando-se uma seqüência de treinamento de 15700 vetores obtida de um arquivo de coeficientes de autocorrelação (vide Apêndice B).

A princípio foram implementados vários quantizadores de 10 bits de codificação para o algoritmo LBG com a distorção de erro quadrático. Alguns dos quais com a quantização direta dos coeficientes LPC e outros com a quantização dos coeficientes REFLEXÃO.

Muito embora seja raro acontecer, a utilização direta dos coeficientes LPC no projeto de "codebooks" pode levar a filtros instáveis. Apesar deste problema, foram gerados "codebooks" através deste método, uma vez que testes subjetivos anteriores revelaram a superioridade da interpolação direta dos coeficientes LPC em relação à interpolação dos coeficientes de REFLEXÃO ou LOG AREA RATIO, a nível de qualidade subjetiva. Esta instabilidade chegou a ocorrer, embora raramente, com alguns dos quantizadores projetados, produzindo *estouros* e *estalos* no sinal de voz quantizado. Como a ocorrência de *estalos* é bastante incômoda, partiu-se para a quantização dos coeficientes de reflexão no projeto final dos quantizadores. A figura 6.1 mostra o comportamento da relação sinal/ruído de quantização durante a geração dos alfabetos de

reprodução para os quantizadores QV1a (limiar de distorção e = 0.5%) e QV1b (e = 0.01%), com o método das divisões sucessivas para o alfabeto inicial.

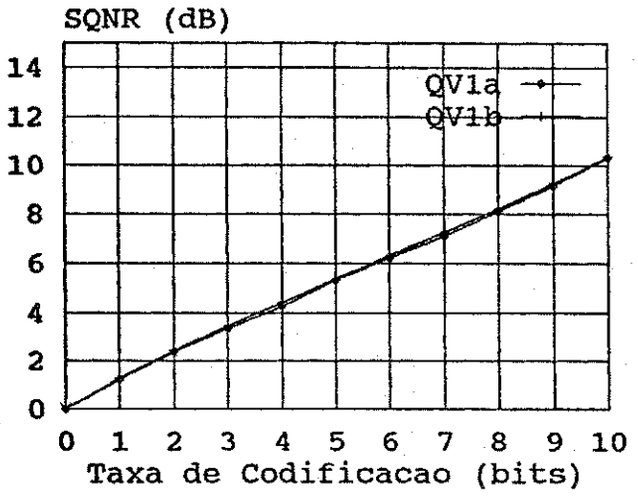


Figura 6.1 - Relação sinal/ruído de quantização do quantizador QV1, durante o projeto do alfabeto de reprodução. Para QV1a, e = 0.5% e para QV1b, e=0.01%.

Foram implementados também dois quantizadores vetoriais com um alfabeto inicial gerado pelo processo aleatório e otimizados pelo algoritmo LBG. Um quantizador (QV1c) foi inicializado selecionando-se aleatoriamente 1024 vetores a partir dos 15700 vetores da seqüência de treinamento. O outro quantizador (QV1d) foi inicializado escolhendo-se 1024 vetores a partir de apenas 10000 vetores da seqüência de treinamento. Para comparações com os outros quantizadores implementados, escolheu-se então o quantizador QV1c, já que ele apresentou uma relação sinal ruído de quantização maior.

Quantizador	SQNR(dB)	e (%)	d (%)	I	T (min)
QV1a	10,33	0,5	1	80	475,0
QV1b	10,40	0,01	1	282	1358,0
QV1c	10,38	0,01	--	20	545,0
QV1d	10,36	0,01	--	19	513,0

Tabela 6.1 - Relação sinal/ruído de quantização para os quantizadores de 10 bits implementados com a distorção de erro quadrático, onde I é o número de iterações, d é a percentagem de pertubação e T é o tempo total de geração dos quantizadores.

Da tabela 6.1 nota-se por comparação dos quantizadores QV1a e QV1b que, em relação ao tempo de execução, não há muita vantagem em se utilizar um limiar de distorção muito pequeno. Isto porque houve um acréscimo de apenas 0.06 dB na relação sinal/ruído de quantização e esse acréscimo foi subjetivamente imperceptível quando da realização de testes informais com os dois quantizadores operando com um mesmo sinal de voz. Analisando agora o desempenho dos quantizadores QV1b e QV1c, constata-se que, apesar de QV1c ter sido projetado com um alfabeto aleatório, não houve muita diferença na SQNR final entre esses quantizadores. Isto mostra que, em termos do desempenho objetivo, o método de geração de alfabetos iniciais aleatórios otimizados pelo algoritmo LBG também fornece bons resultados e um menor tempo de execução.

Ainda da tabela 6.1, comparando-se o tempo de projeto dos quantizadores QV1a e QV1c, pode parecer à primeira vista que se deveria esperar um tempo de projeto do quantizador QV1a bem superior ao do quantizador QV1c, já que o primeiro foi projetado com o método de divisões sucessivas e o segundo foi inicializado com um alfabeto aleatório. Isto entretanto não ocorreu porque o limiar de distorção para QV1a foi bem maior que o limiar para QV1c. Deve-se ressaltar também que, mesmo tendo sido utilizado um limiar de 0.01% no projeto dos quantizadores QV1b, QV1c e QV1d, a utilização de um limiar de 0.1% é considerada razoável.

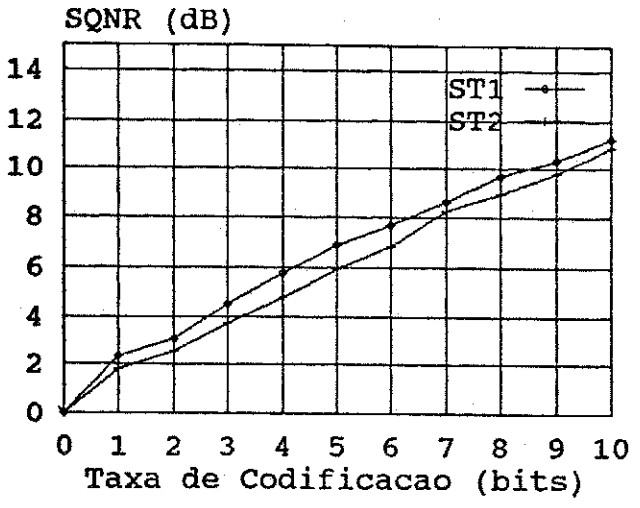


Figura 6.2 - Relação sinal/ruído de quantização dos sinais de seste ST1 e ST2 com o quantizador QV1b.

A figura 6.2 mostra a relação sinal/ruído de quantização para os sinais de teste ST1 e ST2 processados no quantizador QV1b. Deve-se observar que a relação sinal/ruído de quantização para o sinal ST1 supera a do sinal ST2. Este resultado era esperado, já que o locutor do sinal ST1 pertenceu à seqüência de treinamento. A tabela 6.2 apresenta os resultados dos sinais de teste ST1, ST2, ST3 e ST4 processados nos quantizadores QV1a, QV1b e QV1c para uma quantização vetorial com 10 bits.

Sinal	SQNR(dB)			TQ(s)
	QV1a	QV1b	QV1c	QV1
ST1	11.72	11.21	11.69	45
ST2	11.05	10.89	10.97	41
ST3	12.12	12.27	12.38	41
ST4	10.97	11.12	11.12	33

Tabela 6.2 - Relação sinal/ruído de quantização para os sinais de testes ST1, ST2, ST3 e ST4 com os quantizadores QV1a, QV1b e QV1c, onde TQ é o tempo de quantização (em segundos).

O tempo de quantização (TQ) é referente à quantização de todo o sinal de teste e foi obtido incluindo os tempos de processamento da

análise LPC, da quantização vetorial e da síntese. Os tempos de quantização para os sinais ST1 e ST2, que foram gerados com a mesma frase, foram diferentes porque as frases foram pronunciadas com velocidades diferentes. Idem para os sinais ST3 e ST4.

Da tabela 6.2, observa-se que a relação sinal/ruído para os sinais ST1 e ST3 foi superior à relação sinal/ruído dos sinais ST2 e ST4. O que também era de esperar, já que o locutor para os sinais ST2 e ST4 não pertence à seqüência de treinamento.

Um resultado surpreendente é que, pelo menos para o alfabeto aleatório inicial utilizado, o quantizador QV1c apresentou melhor desempenho que os quantizadores QV1a e QV1b para os sinais de teste ST3 e ST4, e que o quantizador QV1b para os sinais ST1 e ST2. Nos demais casos, a SQNR para o quantizador QV1c manteve-se bem próxima da SQNR dos outros quantizadores. Foi realizada uma comparação subjetiva informal para o sinal de teste ST2 processado nos quantizadores QV1b e QV1c, e não notou-se nenhuma diferença sensível nos sinais processados. Isto mostra que se consegue obter bons quantizadores a partir de alfabetos iniciais aleatórios, sem a necessidade da utilização do método das divisões sucessivas.

Foi visto da figura 6.1 que, em termos da relação sinal/ruído de quantização, o quantizador QV1b superou ligeiramente o quantizador QV1a. Com isso, esperava-se que o quantizador QV1b apresentasse melhor desempenho que o quantizador QV1a em todos os sinais de teste, já que ele foi concluído com uma distorção média menor. A tabela 6.2 mostra que isto realmente ocorreu quando os sinais de teste ST3 e ST4 foram processados. Entretanto, com os sinais ST1 e ST2 aconteceu justamente o contrário, ou seja, o quantizador QV1a que deveria apresentar um desempenho inferior, superou o desempenho do quantizador QV1b. Esta mesma análise também é válida para os quantizadore QV1b e QV1c, já que este último foi concluído com uma SQNR menor e quando utilizado com todos os sinais de testes apresentou um desempenho superior ao do quantizador QV1b. Isto não significa que o quantizador QV1b não seja localmente ótimo ou tenha desempenho inferior aos outros quantizadores, e sim que existem casos isolados onde um quantizador pode ser superado por outro que tenha apresentado um desempenho inferior durante o

projeto. E também porque os resultados das SQNR dos quantizadores foram tão próximos que, quando utilizados com os sinais de testes, essa pequena diferença se torna praticamente desprezível. Em testes informais também não notou-se nenhuma diferença na qualidade subjetiva dos sinais quantizados.

6.3.2 - QUANTIZADOR VETORIAL 2 (QV2)

Este quantizador também foi implementado com o algoritmo LBG, mas com a versão 1 da distorção de Itakura-Saito. A partir do arquivo de coeficientes de autocorrelação (vide Apêndice B) foi obtida uma seqüência de treinamento composta por coeficientes LPC, uma vez que a versão 1 da distoção de Itakura-Saito depende tanto dos coeficientes de autocorrelação do sinal de voz como dos coeficientes LPC com e sem quantização (vide equação 3.17). O cálculo dos centróides para esta medida de distorção é realizado a partir da função de autocorrelação média de cada célula.

Para este quantizador, foi explorada a percentagem de perturbação para a divisão das células utilizando o método de divisões sucessivas apresentado no ítem 5.6. A figura 6.3 mostra a relação sinal/ruído de quantização durante o projeto do alfabeto de reprodução para o quantizador QV2a, implementado com uma pertubação $\delta = 10\%$ e para o quantizador QV2b, realizado com uma pertubação de $\delta = 1\%$, ambos com o mesmo limiar de distorção ($\varepsilon = 0.01\%$).

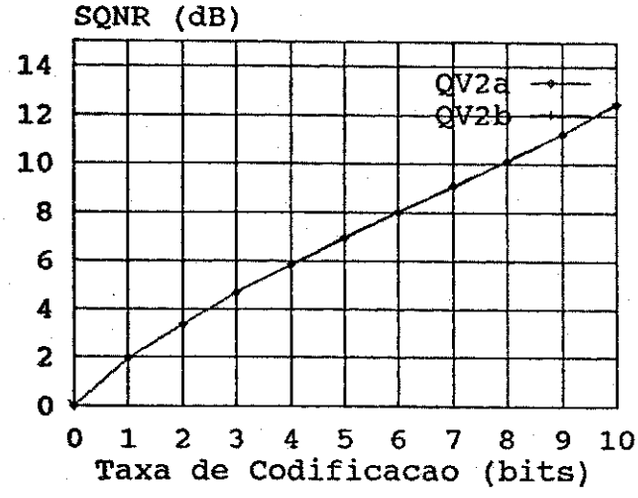


Figura 6.3 - Relação sinal/ruído de quantização durante o projeto dos alfabetos de reprodução dos quantizadores QV2a e QV2b.

A tabela 6.3 resume os parâmetros finais para os quantizadores QV2a e QV2b, juntamente com os parâmetros dos quantizadores QV2c e QV2d, que foram implementados com outros limiares de distorção. Na otimização do alfabeto de 10 bits para o quantizador QV2a, surgiram 3 células vazias que foram eliminadas pelo método apresentado no item 5.5.

Quantizador	SQNR(dB)	ε (%)	δ (%)	I	T (min)
QV2a	12,46	0,01	10	295	1579,4
QV2b	12,51	0,01	1	295	1607,2
QV2c	12,48	0,1	1	143	839,1
QV2d	12,45	0,5	1	84	527,9

Tabela 6.3 - Parâmetros finais dos quantizadores implementados, onde ε é o limiar de distorção, δ é a percentagem de pertubação, I é o número de iterações e T é o tempo total de projeto dos quantizadores.

Novamente, em relação ao limiar de distorção, não houve muita diferença na relação sinal/ruído dos quantizadores obtidos. Oberva-se entretanto que a diferença foi acentuada no tempo de processamento no projeto dos quantizadores.

Como os quantizadores QV2a e QV2b foram gerados com o mesmo limiar de distorção, era de se esperar também que a relação sinal/ruído de quantização para o quantizador QV2b fosse maior que a do QV2a, já que os vetores originados após a pertubação, durante o projeto de QV2b, são bem mais próximos dos vetores ótimos do que os vetores pertubados durante o projeto do quantizador QV2a. Este fato foi verificado, embora a diferença de 0,05 dB seja insignificante.

As tabelas 6.1 e 6.3 também mostram a importância de se utilizarem vários procedimentos ou combinações de inicializações na produção dos alfabetos de reprodução para depois se escolher o alfabeto de menor distorção média ou maior relação sinal/ruído de quantização.

A tabela 6.4 apresenta a relação sinal/ruído de quantização para os sinais de teste ST1, ST2, ST3 e ST4 com todos os quantizadores implementados.

	SQNR (dB)				
Quantizador	QV2a	QV2b	QV2c	QV2d	TQ(s)
ST1	11,85	11,89	11,92	11,89	89,5
ST2	11,17	11,00	11,04	11,22	77,9
ST3	11,74	11,79	11,66	11,84	79,9
ST4	10,54	10,63	10,66	10,64	65,3

Tabela 6.4 - Relação sinal/ruído de todos os sinais de teste com os quantizadores implementados com a versão 1 da distorção de Itakura-Saito, onde TQ é o tempo de quantização (em segundos).

Observa-se da tabela 6.4 que a relação sinal/ruído para os sinais de testes ST1 e ST3 foi sempre superior à dos sinais ST2 e ST4 para cada quantizador implementado. Isto foi devido ao fato de os sinais de teste ST2 e ST4 não terem sido produzidos por um locutor pertencente à seqüência de treinamento. Contudo, vê-se que a relação sinal/ruído para os sinais ST2 e ST4 em cada quantizador não foi muito diferente da relação para os sinais ST1 e ST3, implicando assim que os quantizadores projetados apresentam bom desempenho mesmo para locutores que não pertencem à seqüência de treinamento. Também para estes quantizadores, testes informais não apresentaram muita diferença em termos da qualidade subjetiva dos sinais quantizados. Adicionalmente, em relação ao quantizador QV2a que apresentou 3 células vazias durante a otimização do alfabeto de reprodução e que foram eliminadas pelo método do item 5.5, constata-se também que o método de eliminação de células vazias não afeta o desempenho do quantizador final, e sim que o desempenho melhora ou, no pior caso, permanece o mesmo.

6.3.3 - QUANTIZADORES PRODUCT CODE

Foram implementados três quantizadores filtro-ganho (QV3, QV4 e QV5) com o algoritmo "product code", utilizando a versão 2 da distorção de Itakura-Saito. Todos com uma codificação de 10 bits

para os coeficientes LPC (codebook de filtros) e 5 bits para o ganho (codebook de ganho). A seqüência de treinamento foi a mesma utilizada nos quantizadores anteriores, só que na geração do alfabeto de reprodução são utilizados a função de autocorrelação e os coeficientes LPC associados. O "codebook" transmissor foi constituído pelos parâmetros da autocorrelação de coeficientes, com cada vetor armazenado na forma

$$r_{\hat{a}}(0),\ 2r_{\hat{a}}(1),\ 2r_{\hat{a}}(2),\ \cdots\ ,\ 2r_{\hat{a}}(M),$$

e o "codebook" receptor foi constituído pelos coeficientes LPC correspondestes a cada função de autocorrelação de coeficientes do "codebook" transmissor. Sendo assim, não é necessário calcular os coeficientes LPC do filtro a ser quantizado em nenhuma parte do processamento de análise/síntese de voz, e sim apenas na geração do alfabeto de reprodução (vide equação 3.46).

O quantizador QV3 foi produzido com o processo de otimização separada (algoritmo BGGM), o quantizador QV4 foi obtido com o processo de otimização conjunta e o quantizador QV5 com a otimização individual. Para os quantizadores QV4 e QV5, como alfabeto inicial foi utilizado o alfabeto do quantizador QV3. A tabela 6.5 apresenta os parâmetros e a relação sinal/ruído de quantização ($SQNR_F$) final relativa à distorção de filtros para os quantizadores em questão.

Quantizador	$SQNR_F$ (dB)	ε (%)	δ (%)	I	T (min)
QV3	10,37	0,01	10	424	1760,3
QV4	10,40	0,01	--	15	1045,5
QV5	10,39	0,01	--	13	479,6

Tabela 6.5 - Parâmetros finais dos quantizadores QV3, QV4 e QV5, onde ε é o limiar de distorção, δ é a percentagem de pertubação, I é o número de iterações e T é o tempo total de projeto dos quantizadores.

Observa-se da tabela 6.5, que os quantizadores QV4 e QV5 não têm a constante de pertubação e que o número de iterações foi bem menor que o número de iterações para o quantizador QV3. Isto porque na otimização conjunta (QV4) e na otimização individual (QV5), o processo é inicializado com um alfabeto já com um número exato de níveis enquanto que o quantizador QV3 foi projetado com o método das divisões sucessivas. O quantizador QV4 foi obtido otimizando-se conjuntamente os alfabetos de filtros e de ganho do quantizador QV3. Já o quantizador QV5 foi obtido otimizando-se apenas o alfabeto de filtros do quantizador QV3 e utilizando-se como "codebook" de ganho o alfabeto de ganho do quantizador QV4.

Constata-se também da tabela 6.5 que a $SQNR_F$ para o quantizador QV4 foi superior à $SQNR_F$ para os quantizadores QV3 e QV5. Evidentemente esperava-se que isto acontecesse, já que no projeto do quantizador QV3 o cálculo dos centróides para as células de filtros não é influenciado pelo ganho e o cálculo do ganho ótimo não é influenciado pelos filtros. Já no quantizador QV5, a otimização é feita apenas no alfabeto de filtros com o ganho influenciando no cálculo dos centróides das células. Como para o quantizador QV4 a otimização dos alfabetos é feita conjuntamente, ou seja, filtro dependendo de ganho e vice-versa, realmente esperava-se uma relação sinal/ruído de quantização melhor. Entretanto, em termos de qualidade e inteligibilidade, testes informais apresentaram uma grande dificuldade em se distinguirem os sinais processados nos quantizadores filtro-ganho apresentados.

Todos estes quantizadores foram utilizados para processar os sinais de testes. A tabela 6.6 mostra a $SQNR_F$ e o tempo de quantização para o sinais processados em cada quantizador.

	$SQNR_F$ (dB)			
Quantizador	QV3	QV4	QV5	TQ(s)
ST1	9.54	9.57	9.55	20.0
ST2	7.96	7.97	7.97	21.8
ST3	9.40	9.45	9.44	19.2
ST4	7.20	7.21	7.21	19.8

Tabela 6.6 - Relação sinal/ruído de quantização para os sinais de testes ST1, ST2, ST3 e ST4 processados nos quantizadores filtro-ganho (10 bits para os coeficientes LPC e 5 bits para o ganho), onde TQ é o tempo de quantização dos sinais.

Observa-se da tabela 6.6 que também com os sinais de testes, o quantizador QV4 apresentou melhor desempenho em relação aos outros dois quantizadores, apesar da diferença não ser muito grande. Constata-se também que em todos os quantizadores, os sinais de teste ST1 e ST3 apresentaram uma $SQNR_F$ superior à apresentada com os sinais ST2 e ST4. Esperava-se também que isto acontecesse, já que o locutor para os sinais de teste ST1 e ST3 pertenceu à seqüência de treinamento. A tabela 6.7 apresenta os resultados em relação à distorção média total (DM) e à distorção média de ganho (DG) para todos os sinais de testes processados.

	QV3		QV4		QV5	
Quantizador	DM	DG	DM	DG	DM	DG
ST1	0.15427	0.02160	0.15346	0.02163	0.15406	0.02163
ST2	0.48339	0.27757	0.48287	0.27767	0.48252	0.27750
ST3	0.16453	0.02601	0.16310	0.02594	0.16337	0.02593
ST4	0.43652	0.23408	0.43609	0.23401	0.43585	0.23400

Tabela 6.7 - Resultado da distorção total e da distorção média de ganho para os sinais de testes processados nos quantizadores filtro-ganho.

Os resultados da tabela 6.7 mostram que mesmo em relação à distorção média de ganho, o desempenho dos quantizadores com cada sinal de teste é praticamente o mesmo. Novamente observa-se que as distorções média total e de ganho foram maiores para os sinais ST2 e ST4 que não pertenceram à seqüência de treinamento.

6.4 - RESULTADOS SUBJETIVOS

Nos testes subjetivos realizados tentou-se seguir os procedimentos do método CJM (Category-Judgment Method) [8]. O método sugere que, dependendo da aplicação do sistema a ser testado, seja utilizado um grupo de ouvintes com aproximadamente 50 pessoas. Tendo em vista as dificuldades encontradas para se reunir o número de ouvintes desejado, utilizou-se um grupo com 35 ouvintes.

Como foram produzidos vários quantizadores vetoriais e a diferença na SQNR quando utilizados com os sinais de teste foi praticamente desprezível, escolheu-se para os testes subjetivos apenas três quantizadores. O quantizador QV1b, projetado com o algoritmo LBG e a distorção de erro quadrático; o quantizador QV2b, produzido com o algoritmo LBG e a versão 1 da distorção de Itakura-Saito, e o quantizador QV4 que foi implementado com a otimização conjunta do algoritmo "product code" e a distorção de Itakura-Saito 2.

Testes subjetivos informais indicaram claramente melhor desempenho dos quantizadores QV2b e QV4 em relação ao quantizador QV1. Entre os quantizadores QV2b e QV4, não notou-se diferença perceptível.

Para comparações em relação ao tempo de quantização de todos os quantizadores implementados, considere-se por exemplo o sinal de teste ST1. O tempo de quantização para o quantizador QV1 foi de 45 segundos; para o quantizador QV2, este tempo foi de 49.5 segundos e para os quantizadores QV3, QV4 e QV5 o tempo foi de 20 segundos. Em termos de desempenho da complexidade computacional (tempo de quantização) e para efeito de implementação em tempo real, deve-se então utilizar a versão 2 da distorção de Itakura-Saito.

Os quantizadores QV1 e QV2 quantizam apenas os coeficientes LPC, já os quantizadores QV3, QV4 e QV5 quantizam tanto os coeficientes LPC como o ganho do sinal de excitação. Com isso, levando em consideração o tempo de quantização, a taxa final de codificação e o resultados dos testes subjetivos informais, escolheu-se apenas o quantizador QV4 para a realização dos testes formais.

Foram realizados quatro testes subjetivos. Como sinal de referência para os testes 1 e 2 foram utilizados os sinais de voz processados no "vocoder" sem quantização, sinais estes que correspondem aos sinais de teste ST1 e ST2 respectivamente. Escolheram-se os sinais processados no "vocoder" sem quantização para garantir a avaliação do desempenho apenas dos quantizadores vetoriais, já que o próprio "vocoder" impõe uma certa degradação no sinal processado. Já para os testes 3 e 4 foram utilizados como sinais de referência os sinais de voz processados no quantizador QV1, sinais correspondentes aos sinais de teste ST3 e ST4 respectivamente. Com este procedimento, designa-se a categoria *boa* para os sinais de referência dos teste 1 e 2, e a categoria *regular* para os sinais de referência dos testes 3 e 4, já que o quantizador QV1 apresentou um desempenho inferior ao dos quantizadores QV2 e QV4.

	Mean Opinion Score			
Sinal	Teste 1	Teste 2	Teste 3	Teste 4
Referência	3.94	3.80	2.18	1.54
de Teste	3.40	3.40	2.63	2.40

Tabela 6.8 - Score de opinião para os sinais de referência e para os sinais teste (processados no quantizador QV4), correspondentes aos quatro testes realizados.

Observa-se da tabela 6.8 que, para os testes 1 e 2, o "score" dos sinais de teste ficou bem próximo do "score" dos sinais de referência. Isto indica que os ouvintes assinalaram, em média, a

categoria *boa* para o sinal de referência e julgaram o sinal de teste com uma qualidade e uma inteligibilidade entre as categorias *satisfatória* e *boa*. O que também representa que o quantizador avaliado produz o bom desempenho em termos subjetivos.

Os resultados da tabela 6.8 mostram ainda que para os testes 3 e 4, onde foram utilizados sinais de referência processados no quantizador QV1 (distorção de erro quadrático), os ouvintes indicaram para os sinais de referência uma qualidade e uma inteligibilidade entre as categorias *péssima e regular*. Para os sinais de teste, a impressão dos ouvintes ficou entre as categorias *regular* e *boa*. O que significa que mesmo em relação a um sinal de referência considerado como péssimo ou regular, o quantizador avaliado também apresenta um bom desempenho.

CAPÍTULO 7

CONCLUSÕES

Os quantizadores vetoriais implementados têm apresentados bons resultados em termos da qualidade subjetiva dos sinais sintetizados, tanto para locutores que participaram da seqüência de treinamento quanto para os locutores que não participaram dela. Isto é importante porque os quantizadores devem apresentar um bom desempenho com qualquer locutor.

Pelo menos para os quantizadores implementados com a distorção de erro quadrático, não houve diferença perceptível entre a qualidade de um sinal processado no quantizador implementado com um alfabeto inicial aleatório e a qualidade de um sinal processado no quantizador projetado com o método de divisões sucessivas. Isto indica que pelo menos para este caso, o quantizador projetado com um alfabeto inicial aleatório apresenta bons resultados se o alfabeto inicial for otimizado pelo algoritmo LBG.

A complexidade computacional dos quantizadores vetoriais é um fator importante para processamento em tempo real. Em relação ao tempo de quantização (que incluiu os tempos da análise e da síntese LPC), os quantizadores vetoriais implementados com a versão 2 da distorção de Itakura-Saito (algoritmo product code) possuem um desempenho superior ao desempenho dos quantizadores implementados com a distorção de erro quadrático e a versão 1 da distorção de

Itakura-Saito através do algoritmo LBG. Isto também ficou evidenciado pelos resultados dos testes subjetivos formais.

A diferença na qualidade subjetiva apresentada pelos quantizadores projetados com as distorções de Itakura-Saito (versões 1 e 2) é praticamente imperceptível. Entretanto, é preferível a versão 2 da distorção de Itakura-Saito por ela apresentar um tempo de quantização menor e por possibilitar uma taxa de codificação menor que a taxa de codificação conseguida com a versão 1 da distorção de Itakura-Saito.

Já os sinais processados no quantizador projetado com a distorção de erro quadrático apresentaram uma qualidade de voz razoável, mas inferior à qualidade dos quantizadores projetados com as distorções de Itakura-Saito. Fica evidenciado pois a superioridade das versões da distorção de Itakura-Saito em relação à de erro quadrático.

Uma limitação natural do vocoder LPC é que ele não garante uma transição *suave* entre quadros sucessivos do sinal de voz. O quantizador vetorial pode piorar ainda mais a transição entre esses quadros. Pode-se, entretanto, conseguir melhores resultados, se os efeitos dessas transições forem reduzidos, através de uma interpolação entre quadros adjacentes, por exemplo.

O quantizador vetorial projetado com o algoritmo "product code" apresentou um bom desempenho à taxa de 1100 bits/s, produzindo sinais de voz inteligíveis e de *boa* qualidade subjetiva. Além disso, muitos dos ouvintes utilizados no teste formal não notaram nenhuma diferença entre o sinal quantizado (com o quantizador projetado com o algoritmo product code) e o sinal sem quantização (processado no vocoder LPC).

A quantização vetorial permite uma grande redução na taxa de codificação dos sinais de voz, surgindo como uma boa alternativa para aplicações onde se deseja reduzir a taxa de transmissão ou onde se deseja economizar memória. Além do mais há uma série de aplicações, tais como reconhecimento de voz, onde é necessário efetuar uma quantização vetorial dos parâmetros antes de sua comparação.

Considerando que os algoritmos para geração de codebook já

estão bem definidos e fornecem resultados satisfatórios, e tendo em vista ainda que os quantizadores vetoriais devem ser implementados para processamento em tempo real, sugere-se então que futuras pesquisas na área da quantização vetorial sejam direcionadas para os métodos de busca.

APÊNDICE A

CÁLCULO DOS CENTRÓIDES

No Capítulo 3 foram apresentados alguns tipos de medidas de distorção utilizadas em codificadores de voz com quantização vetorial e no Capítulo 4 foi mostrado que, nos métodos de geração de alfabetos de reprodução, é necessário calcularem-se os centróides de cada célula. Apresentam-se aqui os centróides relacionados a cada medida de distorção.

A.1 - Distorção de Erro Quadrático

Seja $S = \{\mathbf{x}_i;\ i=1,2,\cdots,N\}$ uma célula com uma coleção de N vetores. O centróide ou ponto de distorção mínima para S é o vetor $\hat{\mathbf{x}}(S)$ que minimiza a distorção média total em $\mathbf{u}$, expressa por

$$D\left(S,\hat{\mathbf{x}}(S)\right) = \frac{1}{N}\sum_{\mathbf{x}_i \in S} d(\mathbf{x}_i,\mathbf{u}), \qquad (A.1)$$

onde o vetor $\mathbf{u}$ é o centróide que minimiza a distorção média. Da equação (3.7) com r = 2 e de (A.1), a distorção média total é expressa por

$$D\left(S,\hat{\mathbf{x}}(S)\right) = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{u}\|_2^2 . \qquad (A.2)$$

Derivando ambos os membros da equação (A.2) em relação a **u** e igualando a zero,

$$\frac{1}{N}\frac{\delta}{\delta \mathbf{u}}\left(\sum_{i=1}^{N}\|\mathbf{x}_i - \mathbf{u}\|_2^2\right) = 0 \qquad \text{(A.3)}$$

obtém-se $$\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{u}) = 0.$$

O que implica em $$\hat{\mathbf{x}}(S) = \mathbf{u} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \ . \qquad \text{(A.4)}$$

Assim, para a distorção de erro quadrático, o centróide da célula S é o seu centro euclideano de gravidade ou baricentro.

A.2 - Distorção de Itakura-Saito (Versão 1)

Seja a célula S como no ítem A.1. Então a distorção média total relativa à versão 1 da distorção de Itakura-Saito é expressa por

$$D\left(S,\hat{\mathbf{x}}(S)\right) = \frac{1}{N}\sum_{\mathbf{x}_i \in S} d_R(\mathbf{x}_i,\mathbf{u}), \qquad \text{(A.5)}$$

onde $d_R(\mathbf{x}_i,\mathbf{u}) = (\mathbf{x}_i-\mathbf{u})\mathbf{R}_i(\mathbf{x}_i-\mathbf{u})^t$ e $\mathbf{R}_i$ é a matriz de autocorrelação e $\mathbf{x}_i$ é o vetor de coeficientes LPC do do i-ésimo quadro de voz. Então a equação (A.5) fica

$$D\left(S,\hat{\mathbf{x}}(S)\right) = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{u})\mathbf{R}_i(\mathbf{x}_i - \mathbf{u})^t. \qquad \text{(A.6)}$$

A matriz de autocorrelação R é um produto natural da análise LPC e não precisa ser recalculada. A minimização de (A.6) é um problema de minimização da energia residual mínima da análise LPC e sua solução é dada pela minimização da expressão $\mathbf{u}\,\bar{\mathbf{R}}\,\mathbf{u}^t$, onde

$$\bar{R} = \frac{1}{N}\sum_{i=1}^{N} R_i, \qquad (A.7)$$

encontrando-se o modelo LPC para a autocorrelação média $\bar{R}$. Então o centróide $\hat{x}(S)$ é o vetor formado pelos coeficientes LPC obtidos da matriz $\bar{R}$ através do algoritmo de Levinson-Durbin.

Dos estudos do problema de minimização na teoria da matriz "Toeplitz" [24], o centróide da célula S pode ainda ser expresso por

$$\hat{x}(S) = \left(\sum_{i=1}^{N} R_i \right)^{-1} \sum_{i=1}^{N} R_i x_i^t \qquad (A.8)$$

Distorção de Itakura-Saito (Versão 2)

Seguindo a mesma notação do Capítulo 5, a equação (5.14) pode ser expressa por

$$d_{IS}(x_k;y_{ij}) = \left[\ln\left(\frac{\hat{a}_i R(x_k)\hat{a}_i^t}{\alpha_\infty^k}\right)\right] + \left[\frac{\hat{a}_i R(x_k)\hat{a}_i^t}{\sigma_j^2} - \ln\left(\frac{\hat{a}_i R(x_k)\hat{a}_i^t}{\sigma_j^2}\right) - 1\right] \qquad (A.9)$$

onde $R(x_k)$ é a matriz de autocorrelação do k-ésimo quadro de voz, $x_k = \{a_{kl};\ l=0,\cdots,M\}$ é o vetor de coeficientes LPC obtidos a partis de $R(x_k)$, $\hat{a}_i = \{\hat{a}_{il};\ l=0,\cdots,M\}$ é o vetor quantizado obtido de um alfabeto de filtros, σ_j é o ganho quantizado obtido de um alfabeto de ganho e α_∞^k é a energia residual resultante da representação do k-ésimo quadro de voz pelo seu filtro ótimo de ordem infinita. A distorção de filtros ou distorção de ganho otimizado, que é o primeiro termo do lado direito da equação (A.9), pode ser expressa por

$$d'(x_k;\hat{a}) = \ln(\alpha^k) - \ln(\alpha_\infty^k), \qquad (A.10)$$

onde $\alpha^k = \hat{a}\, R(x_k)\, \hat{a}^t$ com o índice i suprimido por conveniência. Para se calcular o centróide da célula de filtros P com N_p vetores, deve-se minimizar a distorção média de filtros expressa por

$$D'(P;u) = \sum_{k=1}^{N_P} \left(\ln(\alpha^k) - \ln(\alpha_\infty^k) \right). \qquad (A.11)$$

D'(P;u) ainda pode ser expressa por

$$D'(P;u) = \sum_{k=1}^{N_P} \ln\left(\alpha^k / \sigma^2(x) \right) + \sum_{k=1}^{N_P} \ln\left(\sigma^2(x) / \alpha_\infty^k \right), \qquad (A.12)$$

onde $\sigma^2(x)$ é um ganho que dependerá do tipo de otimização (conjunta, separada ou individual) do "codebook". O segundo somatório na equação (A.12) não depende dos parâmetros do filtro A(z) e é apenas uma função da energia dos quadros de voz pertencentes à célula P. O primeiro somatório equivale ao produto de N_P pelo logaritmo da média geométrica da razão $\alpha^k/\sigma^2(x)$ para $k = 1,\cdots,N_P$. Utilizando a média aritmética como uma aproximação para a média geométrica [25], a equação (A.12) pode ser escrita como

$$D'(P;u) \cong N_P \ln\left(\frac{1}{N_P} \sum_{k=1}^{N_P} \left(\alpha^k / \sigma^2(x) \right) \right) + \sum_{k=1}^{N_P} \ln\left(\sigma^2(x) / \alpha_\infty^k \right). \qquad (A.13)$$

Para uma minimização aproximada de (A.13), deve-se minimizar o termo entre parênteses da eq. (A.13) que é a média aritmética da razão $\alpha^k/\sigma^2(x)$. Esta minimização é obtida encontrando-se o modelo LPC para a matriz de autocorrelação média dada por

$$\bar{R} = \frac{1}{N_P} \sum_{k=1}^{N_P} R(x_k)/\sigma^2(x), \qquad (A.14)$$

que difere da equação (A.7) apenas pelo termo $\sigma^2(x)$. Assim cada matriz de autocorrelação associada ao k-ésimo quadro é normalizada por $\sigma^2(x)$ antes de se calcular o modelo LPC para $\bar{R}$.

O cálculo do centróide para o "codebook" de ganho é obtido minimizando-se o segundo termo da equação (A.9), que é a distorção de ganho normalizado. Assim, para uma célula de ganho Q com N_Q

vetores, a distorção média total de ganho é dada por

$$D''(Q;\sigma) = \sum_{k=1}^{N_Q} \left[(\alpha^k/\sigma^2) - \ln(\alpha^k/\sigma^2) - 1 \right], \qquad (A.15)$$

com o índice j também suprimido. Uma vez encontrando a energia residual α^k para cada quadro de voz, a equação (A.15) é minimizada obtendo-se σ^2 como a média aritmética das energias residuais dos vetores pertencentes à célula, ou seja,

$$\sigma^2 = \frac{1}{N_Q} \sum_{k=1}^{N_Q} \alpha^k \qquad (A.16)$$

APÊNDICE B

SEQÜÊNCIA DE TREINAMENTO

A seqüência de treinamento utilizada no projeto dos quantizadores vetoriais foi composta por treze locutores, sendo cinco masculinos e oito femininos. Cada locutor contribuiu com aproximadamente 27 segundos de voz obtida a partir de textos diversos.

Para cada locutor foi gravado um arquivo de voz separado. Os arquivos foram individualmente pré-processados para eliminar as partes de silêncio no início e no final do texto, e produzir um nível sonoro igualitário entre os arquivos. Após o pré-processamento, os arquivos individuais foram unificados em um só arquivo de 6 minutos. A seguir são apresentados os textos utilizados.

Texto 1: "Então qual é a coisa principal para a garantia do sucesso? Não é conhecimento, nem educação, nem preparo, nem experiência, nem dinheiro. É a confiança. Confiança em um projeto e confiança em si mesmo são vitais para o sucesso. Naturalmente outras qualificações são extremamantes importantes, mas o fator principal, o basicamente essencial, é a confiança. A crença de que você pode gerar os resultados poderosos que deseja. Outra maneira de descrever confiança ou crença é o pensamento positivo. É um fato provado que algumas pessoas que pensam positivo obtém resultados

positivos...".

Texto 2: "No princípio era o escambo. O homem dava o que lhe sobrava e recebia o que necessitava. Mas essa troca absoluta, o supérfluo pelo fundamental sem noção de outros valores, começou a ficar difícil quando a ambição perguntou. Espera aí, quantos macacos vale uma canoa? O homem procurou então alguma coisa de utilidade universal que interessasse a todos e pudesse ser trocada por tudo. E o boi virou moeda, uma das mais estáveis e constantes de todos os tempos, pois além do mais é automóvel ...".

Texto 3: "Como o pensamento é algo que acontece em sua mente e que portanto, você pode controlar se assim desejar, e como o desânimo é um acúmulo de pensamentos sombrios, você pode escolher entre acolher ou expulsar esses pensamentos. Esta foi a conclusão sensata a que chegou meu amigo e isso faria sentido, pois quando ele ordenava que os pensamentos sombrios fossem embora eles na verdade não obedeciam. É claro que tentavam resistir, mas ele os enfrentava com poder. Em breve os tinha dominado. Ação era necessária, simplesmente ação. Estava cansado de resmungar e me queixar de auto-piedade que ...".

Texto 4: "É quase irresistível, quando se olha para trás, procurar os fatos mais marcantes em nossas vidas. No jornalismo, tradicionalmente, o evento mais dramático e rico é a guerra, essa sinfonia, completa de possibilidades do homem, na qual medo e bravura, esperança e traição, adquire seu ponto máximo em densidade. Felizmente, de 1968 para cá, aprendeu-se com mais ênfase a conviver com a necessidade de paz e aceitar a exigência da preservação da espécia, com a tomada de conciência que a terra é uma fonte de recursos não renovável, ...".

Texto 5: "Além do mais você aprende a ser filósofo. Freqüentemente, o que parece ser adversidades destrutivas acabam sendo vantagens criativas. E essas vantagens não teriam se tornardo suas se algo, que a princípio parece arruinar tudo para você, não tivesse acontecido. Quando as coisas estão dando errado, pode ao final dar certo. Portanto, quando suas esperanças, metas e sonhos forem destruídos, procure nos destroços. Você pode achar sua oportunidade de ouro onde aparece só haver ruína. Lembra-se da estória de Murd Kybraum, um dos maiores jogadores de baseball do

seu tempo. Seus pais eram muito pobres, mas como bons americanos nem sabiam que o eram ...".

Texto 6: "Desde a queda do regime socialista do leste europeu e da falência da economia soviética, o governo Fidel Castro vem insistindo com maior vigor na integração da América Latina com o objetivo de formar-se um bloco com êxito de decisões comerciais. E dentro da América Latina, é no Brasil que os cubanos depositam maiores esperanças para sair da sinuca conseqüente do desmantelamento do bloco socialista e do bloqueio americano. Este último já dura vinte e oito anos. Na ...".

Texto 7: "O sucesso potencial de determinada apresentação de um assunto, não deve ser avaliado apenas em termos das idéias específicas aprendidas e das técnicas específicas adquiridas. A apreciação final deve também levar em conta quão bem o estudante ficará preparado para continuar seu estudo do assunto. Seja por si mesmo, seja através do trabalho em um curso complementar. Este ...".

Texto 8: "Existiu, nos anos 70, um sonho chamado Brasil Grande. Era um país imaginário, concebido nas planilhas do AI-5. Tinha índices fantásticos de crescimento econômico, a confiança do mundo, investimentos maciços e obras gigantescas. O governo acreditava que desse Grande Brasil sairia uma nova potência mundial. Saiu um grande buraco: a maior dívida externa do mundo e uma conta inadministrável. Hoje o Brasil Grande pode ser visitado nas montanhas de Minas Gerais, nas ruínas da Ferrovia do Aço; nas praias de Angra dos Reis, no esqueleto de usinas nucleares; nas matas da Amazônia, nas estradas ...".

Texto 9: "Fico satisfeito porque nunca fui capaz de dizer ou até mesmo pensar que já tinha alcançado tudo que queria. Estou ainda sonhando, ainda planejando, ainda tentando, ainda trabalhando e tudo é continuamente emocionante. Ouvi pessoas dizerem, finalmente cheguei lá. Sobre pessoas realizadoras nós exclamamos com admiração: Ele chegou ao topo, ele tem tudo que deseja, ou ainda ela tirou a sorte grande, aquela mulher tem tudo e assim por diante. Mas às vezes noto algumas coisas tristes nessas pessoas que chegaram lá. Elas carecem de uma certa qualidade, um incentivo, uma motivação ...".

Texto 10: "O Brasil Grande era uma ficção estatística montada por um governo ditatorial que supunha possível ao estado administrar um país sem sociedade. Os indicadores da distribuição de renda ensinavam, em 1970, que o país estava numa das situações mais perversas do mundo. Eram indicadores pessimistas aos quais a propaganda oficial respondia com outro slogan: Brasil, ame-o ou deixe-o. O ministro Antonio Delfim Neto, pai do que se chamava Milagre Econômico Brasileiro, dizia que era preciso primeiro ...".

Texto 11: "Somente durante o ano de 1987, na Amazônia, uma área de 8 milhões de hectares perdeu a capa de árvore que a recobria, o que equivaleria a deixar sem uma única folha verde a extensão total do Estado de Sergipe. Somente em Rondônia, e também só em 1987, a queima de árvore jogou mais carbono na atmosfera de que toda cidade de São Paulo nos últimos sessenta anos. Essas duas cifras — para ficar só no Brasil e só na Amazônia — ...".

Texto 12: "As manifestações pacifistas dos anos sessenta produziram algo bem maior que o fim da guerra que os Estados Unidos faziam no Vietnam: incorporaram aos direitos dos cidadãos a prerrogativa de fazer a paz. Até então as grandes máquinas dos estados tinham o direito de fazer as guerras. Depois que jovens de todo mundo foram para as ruas cantando e colocando flores nos canos dos fuzis dos soldados que os enxotavam, a guerra foi desmistificada. A manifestação democrática da sociedade americana ...".

Texto 13: "Mesmo que o critério de importância fosse quantitativo — qual o evento que afetou o maior número de pessoas —, a resposta seria múltipla. Como contrapor a pertubadora e cósmica descida do homem na Lua, a revolução provocada na terra pela banalização do microcomputador? Ou a virada do mundo de cabeça para baixo nos anos 70, devido à escassez de um único produto, o petróleo, e a revolução prevista para as próximas décadas com o desenvolvimento de outros materiais como os supercondutores. Não há como comparar ...".

Com o arquivo da seqüência de treinamento, foi gerado um arquivo de coeficientes de autocorrelação obedecendo à seguinte condição:

Se $\{$pitch $\neq 0$ ou (pitch $= 0$ e $\alpha \geq$ limiar)$\}$, armazene os coeficientes de autocorrelação.

Este procedimento teste foi realizado para eliminar os quadros de silêncio da seqüência de treinamento. O valor do limiar para sons surdos foi encontrado gravando-se separadamente vários sons surdos de baixa energia residual. Esses sons surdos foram processados no "vocoder LPC" e a partir dos valores das energias residuais obtidas no processamento, foi escolhido um limiar inferior ao menor valor das energias encontradas. Com a eliminação dos quadros de silêncio, obteve-se uma seqüência de treinamento com 15700 vetores de coeficientes de autocorrelação.

Dependendo da medida de distorção utilizada, os vetores de coeficientes de autocorrelação são transformados. Por exemplo, para a medida de distorção de erro quadrático, os vetores de coeficientes de autocorrelação foram transformados em coeficientes LPC ou em coeficientes de REFLEXÃO e para a distorção de Itakura-Saito 1, os vetores de coeficientes de autocorrelação são transformados em coeficientes LPC. Para a versão 2 da distorção de Itakura-Saito, não é necessário nehuma transformação, já que esta distorção pode ser calculada a partir apenas das energias residuais, que envolvem (vide as equações 3.46 e 5.10) a função de de autocorrelação do quadro de voz e a função de autocorrelação de coeficientes do filtro quantizado $r_{\hat{a}}(\cdot)$.

APÊNDICE C

AMBIENTE DE TRABALHO

A aquisição dos sinais de voz e a reprodução destes sinais para avaliação subjetiva foram realizadas no Sistema de Análise e Processamento Digital de Voz, SAPDV-A [7] do Laboratório de Comunicações Digitais do DECOM/FEE/UNICAMP, descrito no item 2.3.1.

Todas as simulações feitas para este trabalho foram realizadas em tempo não real, numa estação de trabalho SUN SPARCstation da SUN Microsystem Incorporation. Esta estação de trabalho possui as seguintes características [38]:

- Sistema de multiprocessamento e multitarefa;
- Processamento paralelo;
- 12.5 MIPS (milhões de instruções por segundo);
- 1.5 MFLOPS (milhões de operações em ponto flutuante por segundo);
- Otimização assembler global, a nível de compilação.

Todos os programas para a geração dos alfabetos de reprodução foram escritos em linguagem C ANSI, para poderem ser executados na SUN. O programa para simulação do "vocoder LPC", que a *priori* estava escrito em linguagem TURBO C e processava apenas em ambiente DOS, sofreu algumas modificações para ser compatibilizado com a linguagem C ANSI e poder ser executado em ambiente SunOS "Unix Like" na estação SUN. Com isso, todos os arquivos de voz foram

transmitidos do SAPDV-A para a SUN, processados e retransmitidos para o SAPDV-A para reprodução e análise subjetiva.

ANEXO 01

PREPARAÇÃO DO TESTE SUBJETIVO
CATEGORY-JUDGMENT METHOD

Ouvinte:_____

Observações: O ouvinte deve estar motivado e entender o propósito, a pergunta e o procedimento do teste.

O teste consiste em duas fases: A primeira é a familiarização ou treinamento dos ouvintes com um sinal de referência. Nesta fase, o ouvinte escuta o sinal de referência quatro vezes. A segunda fase é a de avaliação do sinal de teste em relação ao sinal de referência, onde o ouvinte escuta aleatoriamente o sinal de teste com o sinal de referência e indica uma categoria para os sinais avaliados.

As categorias abaixo devem ser indicadas em termos da qualidade e inteligibilidade do sinal de teste.

CATEGORIA	PESO
Excelente	5
Boa	4
Satisfatória	3
Regular	2
Péssima	1

TESTE 1	TESTE 2	TESTE 3	TESTE 4

REFERÊNCIAS BIBLIOGRÁFICAS

[1] R. W. Schafer & J. D. Markel, *Speech Analysis*, IEEE Press, Inc., New York, 1979.

[2] R. W. Schafer & L. R. Rabiner, " Digital Representation of Speech Signals", Proceedings of the IEEE, vol. 63, pp 662-677, 1975.

[3] J. Makhoul, S. Roucos & H. Gish, "Vector Quantization in Speech Coding", Proceedings of the IEEE, vol. 73, pp. 1551-1588, 1985.

[4] R. M. Gray, "Vector Quantization", IEEE Acoustic, Speech and Signal Processing Magazine, vol. 01, pp. 4-29, 1984.

[5] D. Y. Wong, B. H. Juang & A. H. Gray, Jr., "An 800 bit/s Vector Quantization LPC Vocoder", IEEE Transactions on Acoustic, Speech and Signal Processing, vol. ASSP-30, pp. 770-780, 1982.

[6] A. Gersho & V. Cuperman, " Vector Quantization: A Pattern-Matching Technique for Speech Coding", IEEE Communications Magazine, vol. 21, pp. 15-21, 1983.

[7] F. Violaro, " Nova Versão do Sistema de Análise e Processamento Digital de Voz - SAPDV-A", Anais do 7º Simpósio Brasileiro de Telecomunicações, pp. 50-53, Florianópolis/SC, 1989.

[8] IEEE, "IEEE Recommended Practice for SPEECH QUALITY MEASUREMENTS", Autor, nº 297, N. York, 1969.

[9] L. R. Rabiner & R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, 1978.

[10] M. R. Schroeder, "Vocoders: Analysis and Synthesis of Speech ", Proceedings of the IEEE, vol. 54, pp. 720-734, 1966.

[11] B. Gold & C. M. Rader,"The Channel Vocoder", IEEE Transaction on Audio and Electroacoustic, vol. AU-15, pp. 148-161, 1967.

[12] J. D. Markel & A. H. Gray, Jr., *Linear Prediction of Speech*, Springer-Verlag, New York, 1976.

[13] J. Makhoul, "Linear Prediction: A Tutorial Review", Proceedings of the IEEE, vol. 63, pp. 561-580, 1975.

[14] J. D. Markel & A. H. Gray, Jr.,"A Linear Prediction Vocoder Simulation Based upon the Autocorrelation Method", IEEE Transaction on Acoustic, Speech and Signal Processing, vol. ASSP-22, pp. 124-134, 1974.

[15] N. S. Jayant & P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Prentice-Hall, Inc., New Jersey, 1984.

[16] G. Bristow, *Electronic Speech Synthesis: Techniques, Technology and Applications*, Granada Publishing Ltd., New York, 1984

[17] B. S. Atal & S. L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave", J. Acoustic Soc. Am., vol. 50, pp. 637-655, 1971.

[18] L. R. Rabiner, M. J. Cheng, A. E. Rosenberg & C. A. McGonegal,"A Comparative Performance Study os Several Pitch Detection Algorithms", IEEE Trans. on Acoustic, Speech end Signal Processing, vol. ASSP-24, pp. 399-417, 1976.

[19] K. Schärer-Vincent,"Pitch Period Detection and Chaining: Method and Evaluation", Phonetica 40, pp. 177-202,1983.

[20] R. M. Gray, J. C. Kiefer & Y. Linde. "Locally Optimal Block Quantizer Design", Information and Control, Academic Press, Inc.,vol. 45, pp. 178-198, 1980.

[21] D. G. Luenberg, *Optimization by Vector Space Methods*, John Wiley & Sons, Inc., N. York, 1969.

[22] D. G. Luemberg, *Linear and Nonlinear Programming*, Addison-Wesley Publishing Company, Inc., 1984.

[23] H. Abut, R. M. Gray & G. Rebolledo, "Vector Quatization of Speech and Speech-Like Waveforms", IEEE Trans. on Acoustic, Speech and Signal Processing, vol. ASSP-30, pp. 423-435, 1982.

[24] Y. Linde, A. Buzo & R. M. Gray,"An Algorithm for Vector Quantizer Design", IEEE Trans. on Communications, vol. COM-28, pp. 84-95, 1980.

[25] A. Buzo, A. H. Gray, Jr., R. M. Gray & J. D. Markel, "Speech Coding Based Upon Vector Quantization, IEEE Trans. on Acoustic, Speech and Signal Processing, vol. ASSP-28, pp. 562-574, 1980.

[26] M. J. Sabin & R. M. Gray, "Product Code Vector Quantizers for Waveform and Voice Coding", IEEE Trans. on Acoustic, Speech and Signal Processing, vol. ASSP-32, pp. 474-488, 1984.

[27] R. M. Gray, A. Buzo, A. H. Gray, Jr., & Y Matsuyama, "Distortion Measures for Speech Processing", IEEE Trans. on Acoustic Speech and Signal Processing, vol. ASSP-28, pp. 367-376, 1980.

[28] B. Chazelle, R. Cole, F. P. Preparata & C. Yap, "New Upper Bounds for Neighbor Searching", Academic Press, Inc., vol. 68, pp. 105-124, 1986.

[29] R. M. Gray, A. H. Gray, Jr., G. Rebolledo & . E. Shore, "Rate Distortion Speech Coding with a Minimum Discrimination Information Distortion Measure", IEEE Trans. on Infomation Theory, vol. 27, pp. 708-721, 1981.

[30] D. Y. Wong, B. H. Juang & A. H. Gray, Jr., "A 800 bits Vector Quantization LPC Vocoder", IEEE Trans. on ASSP, vol. ASSP-30, 1982.

[31] R. M. Gray & Y. Linde, "Vector Quantizers and Predictive Quantizers for Gauss-Markov Sources", IEEE Trans. on Communication, vol. COM-30, pp. 381-389, 1982.

[32] R. B. Ash, *Real Analysis and Probability*, Academic Press, New York, 1972.

[33] L. Breiman, "The Individual Ergodic Theorem of Information Theory", Ann. Math. Statistics, vol. 28, pp. 809-811, 1957.

[34] S. P. Lloyd, "Least Squares Quantization in PCM's", Bell Telephone Labs. Memorandum, Murray Hill, N, Jersey, 1957.

[35] K. R. Parthasarathy, *Probability Measures on Metric Spaces*, Academic Press, New York, 1967.

[36] R. T. Rockafellar, *Convex Analysis*, Princeton Univ. Press, N. Jersey, 1970.

[37] B. H. Juag, D. Y. Wong & A. H. Gray, Jr., "Distortion Performance of Vector Quantization for LPC Voice Coding", IEEE Trans. on ASSP, vol. ASSP-30, pp. 294-303, 1982.

[38] SUN MICROSYSTEM INCORPORATION, "SunOS 4.1 RELEASE MANUALS", O Autor, 1990.