UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE COMUNICAÇÕES

VOCODER LPC

COM QUANTIZAÇÃO VETORIAL

JOSÉ ANTÔNIO MARTINS
Orientador: *Prof. Doutor FÁBIO VIOLARO*

Banca Examinadora: FÁBIO VIOLARO (UNICAMP)
RUI SEARA (UFSC)
AMAURI LOPES (UNICAMP)
REGINALDO PALAZZO JÚNIOR (UNICAMP)

Tese apresentada à Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas - UNICAMP, como parte dos requisitos exigidos para a obtenção do título de MESTRE EM ENGENHARIA ELÉTRICA.

ABRIL - 1991

Resumo

Neste trabalho são descritos os princípios do vocoder LPC, sendo mostrados os métodos para cálculo dos parâmetros do mesmo. Também são apresentados os resultados de simulações de vocoders LPC usando quantização escalar, quantização vetorial e interpolação dos parâmetros quantizados. Inicialmente foi projetado um vocoder LPC não quantizado, o qual serviu de padrão para a avaliação dos vocoders quantizados. Usando a quantização escalar dos coeficientes razão log-área foi obtido um vocoder à taxa de 2200 bit/s, assegurando uma boa qualidade e alta inteligibilidade da voz sintetizada. Com o uso da quantização vetorial obteve-se um bom desempenho em taxas da ordem de 1000 bit/s. Essas taxas foram reduzidas em 50% com o uso da interpolação linear, transmitindo apenas os parâmetros dos quadros ímpares. Assim, conseguiu-se vocoders com taxas ao redor de 500 bit/s, apresentando voz sintetizada com degradação em relação aos sistemas anteriores, mas ainda assegurando uma boa inteligibilidade.

AOS MEUS PAIS
RUBENS E EULÁLIA

Agradecimentos

Ao *Professor Doutor Fábio Violaro*, meu grande agradecimento pela orientação, dedicação e discussões, sem os quais não seria possível a realização deste trabalho. Agradeço ao *Engenheiro José Sindi Yamamoto* e companheiros de trabalho da Área de Processamento Digital de Voz do CPqD/TELEBRÁS, pelo apoio, colaboração e pelas discussões, que em muito contribuíram para o desenvolvimento deste trabalho. Meus agradecimentos também às engenheiras *Flávia Martinho Ferreira Rocha* e *Margarete Mitiko Iramina* e à desenhista *Valéria Timpani Fortunato* pelas figuras. Gostaria de expressar os meus sinceros agradecimentos a todos os meus *amigos* que sempre me incentivaram e colaboraram com suas vozes para a execução das simulações realizadas neste trabalho. A todos o meu *Muito Obrigado*.

Índice

Capítulo 1

INTRODUÇÃO

As técnicas para processamento digital de voz encontraram um grande impulso a partir do avanço da tecnologia de componentes eletrônicos, desenvolvimento de computadores mais rápidos e os crescentes avanços na teoria de processamento digital de sinais.

A representação da voz na forma digital pode ser classificada, de uma forma geral, em dois grandes grupos: codificação de forma de onda e codificação paramétrica. Pertencem ao primeiro grupo, a técnicas que envolvem a reprodução da forma de onda do sinal original com a maior fidelidade possível. Como exemplo, podem ser citados o PCM (Pulse Code Modulation) e o ADPCM (Adaptive Differencial Pulse Code Modulation). A codificação paramétrica envolve a utilização de um modelo para a produção de voz. O modelo de produção de voz mais usado é baseado na estrutura dos órgãos vocais humanos e nas características do sinal de voz. Os parâmetros desse modelo são obtidos através do processamento do sinal de voz, sendo os mesmos utilizados para a obtenção do sinal de voz sintetizado. A combinação desses dois tipos de codificação é denominada codificação híbrida.

A escolha do tipo de codificação de voz a ser utilizado envolve vários fatores como: custos de transmissão, qualidade da voz sintetizada, flexibilidade da representação, complexidade do codificador, taxas de transmissão, entre outros.

As técnicas incluídas na classe de codificação de forma de onda, apresentam bom desempenho quando trabalham em taxas maiores que 15000 bit/s, enquanto as técnicas de codificação paramétrica conseguem um bom desempenho em taxas muito menores.

Como a largura de faixa de um canal de transmissão limita o número de sinais que podem ser transmitidos pelo mesmo, e com o crescente número de aplicações que necessitam de armazenamento de voz, a taxa de bits tornou-se um fator importante na escolha do tipo de codificação. Assim, devido ao fato de alcançar taxas muito baixas, a codificação paramétrica tem evoluído bastante, destacando-se a utilização da técnica de predição linear. Essa técnica em muito contribuiu para o bom desempenho dos codificadores paramétricos com baixas taxas.

Um dos mais importantes codificadores paramétricos é o **vocoder LPC**, com o qual pode-se conseguir voz sintetizada com alta inteligibilidade a uma taxa muito baixa. Como aplicação desse sistema pode-se citar o armazenamento de voz em sistemas de resposta automática por voz. O mesmo também está relacionado com desenvolvimento de terminais telefônicos de baixo custo.

Devido a essas razões, foi realizado neste trabalho um estudo sobre o **vocoder LPC**, com o objetivo de obter-se voz sintetizada com boa inteligibilidade em taxas menores que 1000 bit/s através do uso de técnicas como a quantização vetorial e a interpolação linear.

Dessa forma, no capítulo 2 é apresentado o sistema de produção de voz humano e um modelo para representá-lo. Esse modelo utiliza um filtro digital para representar o trato vocal e classifica os sons em sonoros e não sonoros. O sinal de excitação do trato vocal para os sons sonoros é representado por uma seqüência de impulsos periódicos e, para os sons não sonoros, o sinal de excitação é representado por uma fonte de ruído de faixa larga.

No capítulo 3 são descritos os princípios da técnica de predição linear, os parâmetros do vocoder LPC e os métodos usados para obtê-los.

No capítulo 4 são discutidos diferentes algoritmos para a detecção do pitch e para a decisão sonoro/não sonoro. O detector de pitch é o componente mais crítico do vocoder LPC. São apresentadas técnicas para a detecção do pitch no domínio do tempo e técnicas utilizando a função de autocorrelação.

No capítulo 5 é mostrado como podem ser obtidos vocoders em taxas que variam de 2200 bit/s a 1000 bit/s utilizando-se a quantização escalar e a interpolação dos parâmetros do vocoder.

No capítulo 6 são descritos os tipos de quantização vetorial. Também são apresentados resultados de simulações usando a quantização vetorial juntamente com a interpolação para obter-se vocoders com taxas em torno de 500 bit/s, assegurando voz sintetizada com boa inteligibilidade.

No capítulo 7 são discutidos outros modelos para a excitação do trato vocal, eliminando a necessidade do detector de pitch. Também é apresentado um RELP (' residual excited LPC vocoder ') a uma taxa de 2500 bit/s , usando quantização vetorial.

No capítulo 8 são apresentadas as conclusões deste trabalho.

As medidas usadas na avaliação objetiva dos resultados e o ambiente de trabalho para a simulação dos vocoders, são descritos nos apêndices I e II respectivamente.

Capítulo 2

MODELO PARA PRODUÇÃO DE VOZ

2.1 INTRODUÇÃO

O sinal de voz, o qual é usado pelas pessoas para a comunicação entre si, é formado por ondas acústicas emitidas pelo sistema vocal. Essas ondas originam-se devido às variações de pressão no sistema vocal. Esse sinal é variante no tempo, mudando suas caracteríticas quando a forma e dimensão do sistema vocal são alteradas.

2.2 PRODUÇÃO DO SINAL DE VOZ

O sistema que produz a voz é formado pelos pulmões, laringe, traquéia, faringe e as cavidades oral e nasal [1]. A figura 2.1 mostra os órgãos que compõem o sistema vocal [1].

Os pulmões constituem a fonte do fluxo de ar que dará origem ao sinal de voz. A traquéia é o canal que conduz o fluxo de ar à laringe.

A laringe é uma estrutura formada por quatro cartilagens. Na parte superior da laringe estão as cordas vocais, um par de estruturas elásticas de tendão e músculos. Através de contrações de vários músculos, as cordas vocais podem ser variadas em comprimento e espessura, e podem ser posicionadas em várias configurações. A abertura entre as cordas vocais é denominada glote, a qual está normalmente aberta durante a respiração.

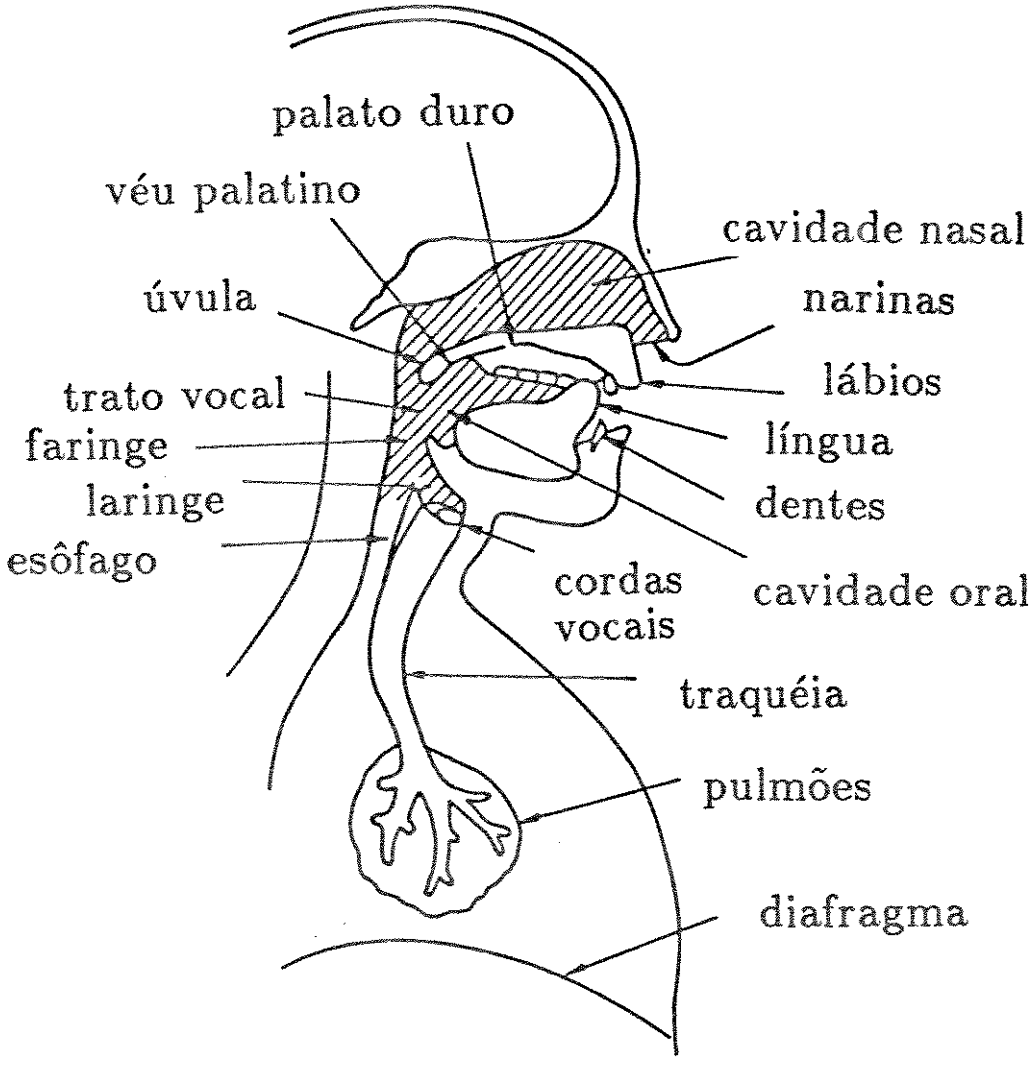


Figura 2.1: Órgãos que compõem o sistema vocal [1].

O fechamento da glote pode obstruir parcial ou totalmente o fluxo de ar que vem dos pulmões. Quando a glote está fechada, o ar ao forçar a passagem, coloca as cordas vocais em vibração. Quando a glote está aberta, o ar passa sem dificuldade e as cordas vocais não vibram.

O trato vocal, formado pela faringe e pela cavidade oral, é o mais importante componente no processo de produção da voz. O trato vocal começa na glote e vai até os lábios. Em homens adultos, o seu comprimento médio é de aproximadamente 17 cm, e a área da sua seção transversal varia de zero a 20 cm^2 [2]. Uma cavidade auxiliar, a cavidade nasal, é acoplada ao trato vocal para a produção dos sons nasais. Esse acoplamento é realizado abaixando-se o véu palatino. Nos homens adultos, a cavidade nasal atinge aproximadamente 12 cm de comprimento.

O trato vocal é um tubo acústico com área de seção transversal não uniforme e variável com o tempo. As freqüências de ressonância do tubo trato vocal são denominadas formantes. O movimento dos órgãos articuladores, mandíbula, lábios, dentes, língua, cordas vocais e véu palatino altera a forma do tubo acústico e assim sua resposta em freqüência. Cada forma do trato vocal é caracterizada por um conjunto de formantes. Diferentes sons são formados variando-se a forma do trato vocal e conseqüentemente suas freqüências de ressonância. A variação dos formantes no tempo em um segmento sonoro de voz pode ser observada na figura 2.2.

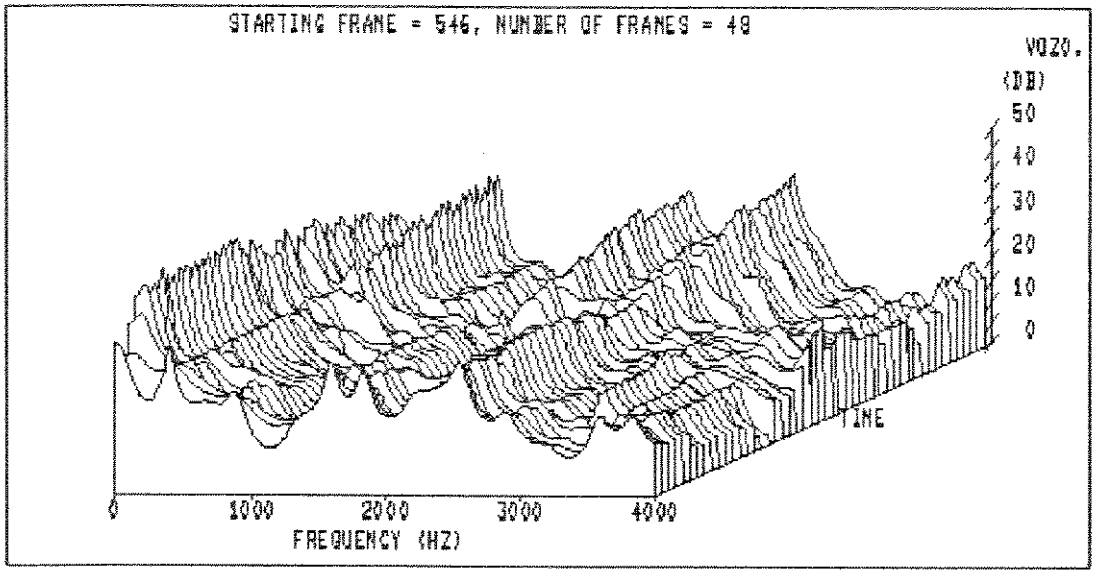


Figura 2.2: Variação dos formantes no tempo em um segmento sonoro de voz.

Na produção dos sons nasais, o trato vocal apresenta freqüências de anti-ressonância, além das freqüências de ressonância.

Os sons gerados pelo sistema vocal são classificados em três categorias, de acordo com o modo de excitação do trato vocal. Assim, os sons podem ser: sonoros, fricativos e explosivos.

Sons Sonoros

Esses sons são produzidos elevando-se a pressão do ar nos pulmões e forçando-se a passagem do ar através da glote, com a tensão das cordas vocais ajustada para que elas vibrem, interrompendo o fluxo de ar através de fechamentos quase periódicos. O fluxo de ar interrompido produz pulsos quase periódicos de faixa larga, os quais irão excitar o trato vocal. Eles são denominados pulsos glotais, e podem ser simulados por ondas triangulares assimétricas [1]. A taxa de vibração das cordas vocais é chamada freqüência fundamental ou pitch, e é dependente da pressão do ar na traquéia e das variações no comprimento, espessura e tensão das cordas vocais. A freqüência fundamental varia entre 50 Hz e 250 Hz para uma voz masculina e pode atingir até 500 Hz para uma voz feminina. A figura 2.3 mostra a forma de onda do som sonoro /a/.

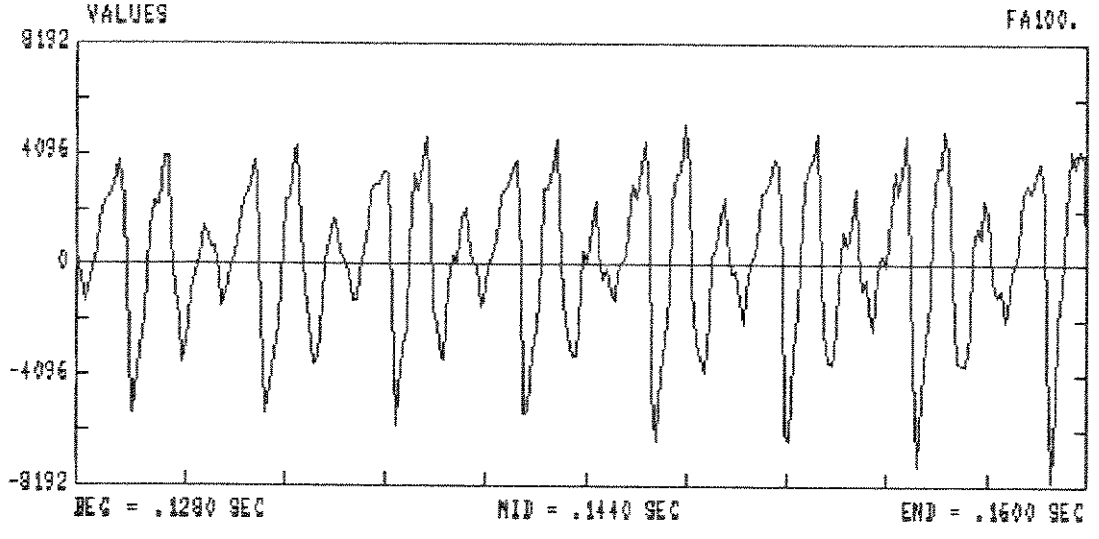


Figura 2.3: Forma de onda do som sonoro /a/.

Sons Fricativos

Esses sons são gerados forçando-se a passagem do ar vindo dos pulmões através de um estreitamento, criado pelos órgãos articuladores, em algum ponto do trato vocal e assim criando-se uma turbulência. A localização desta constrição no trato vocal determina qual som fricativo é produzido. Quando, além da constrição, ocorre a vibração das cordas vocais, tem-se os sons fricativos sonoros. O sinal de excitação do trato vocal na produção dos sons fricativos pode ser modelado por um ruído de faixa larga. A figura 2.4 mostra a forma de onda do som fricativo /f/ em /fa/.

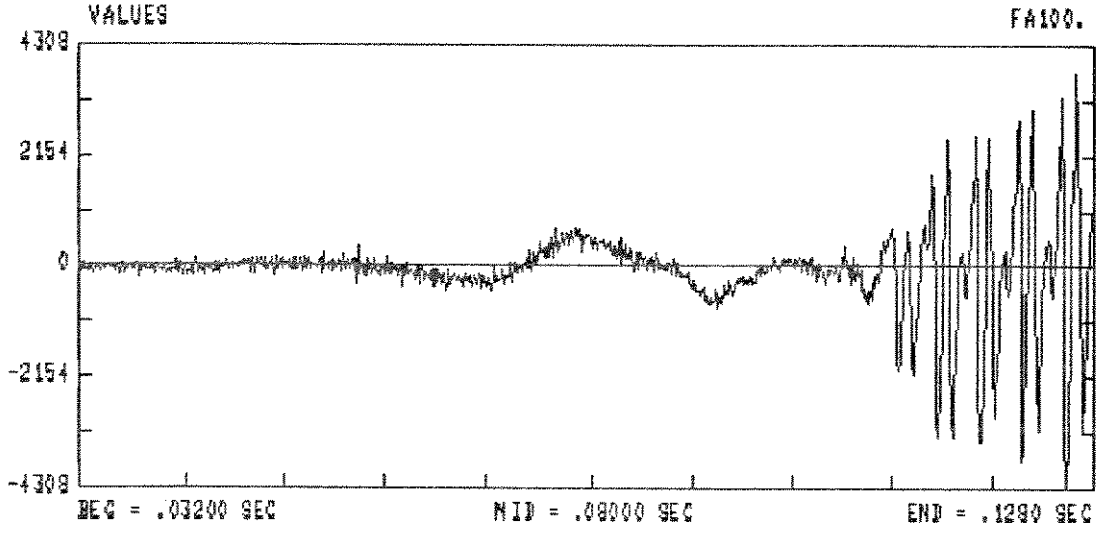


Figura 2.4: Forma de onda do som fricativo /f/ em /fa/.

Sons Explosivos

Resultam de um completo fechamento de algum ponto do trato vocal. O fluxo de ar dos pulmões é interrompido, e passa a exercer uma pressão atrás da obstrução. Com a remoção da mesma ocorre um abrupto relaxamento da pressão, gerando um transitório, o qual é seguido por ruído. A figura 2.5 mostra a forma de onda do som explosivo /p/ em /pa/.

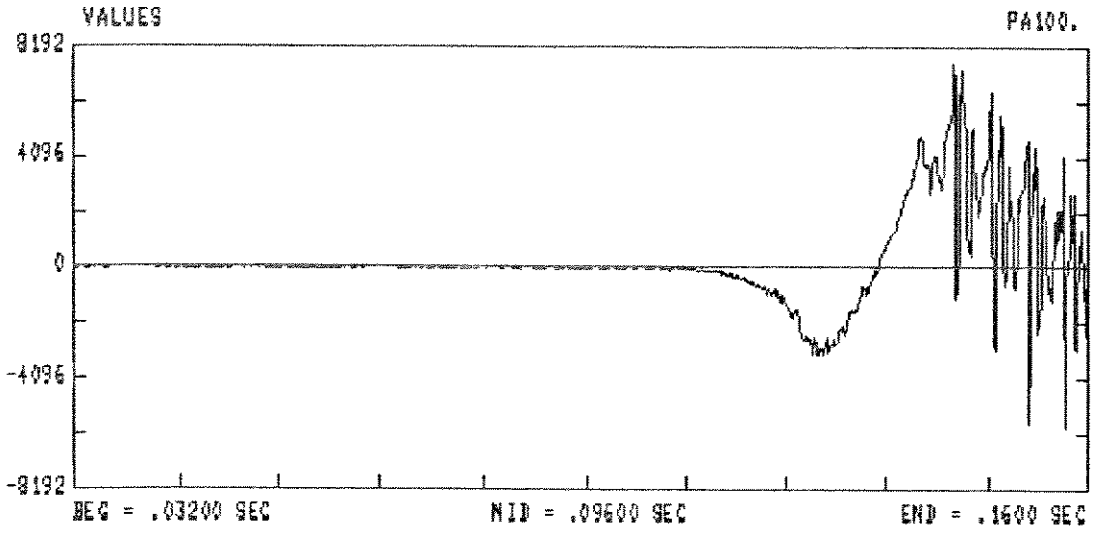


Figura 2.5: Forma de onda do som explosivo /p/ em /pa/.

Em resumo, o sistema de produção de voz é caracterizado por um conjunto de freqüências de ressonância e anti-ressonância, as quais dependem do formato do trato vocal. Além disso, três diferentes formas de excitação do trato vocal são usadas para produzir os diferentes sons.

2.3 CLASSIFICAÇÃO DOS FONEMAS NA LÍNGUA PORTUGUESA

Fonemas são as menores unidades sonoras da fala [3]. Funcionam como elementos diferenciadores das palavras, isto é, são capazes de diferenciar uma palavra das outras, de forma que não podem substituir-se mutuamente sem alterar o sentido das palavras.

Os 33 fonemas da língua portuguesa são classificados em vogais, semi-vogais e consoantes [3].

2.3.1 Vogais

São fonemas sonoros. A corrente de ar sonorizada que sai da laringe encontra na faringe, fossas nasais e boca, uma caixa de ressonância de dimensões e formas variáveis.

Classificam-se de acordo com quatro critérios [4]:

Zona de articulação:

Zona de articulação é o ponto em que se dá o contato ou a aproximação dos órgãos que cooperam para a produção dos fonemas. Na produção das vogais, esses órgãos são: a língua e o palato (palato duro e véu palatino). Quanto à zona de articulação, as vogais classificam-se em:
anteriores:

/é/ em fé /i/ em ri

posteriores:

/ó/ em nó /u/ em tatu

média:

/a/ em ave

Timbre:

O timbre das vogais resulta da maior ou menor abertura da boca. Essa abertura é máxima na produção das vogais abertas. Quanto ao timbre, as vogais podem ser:
abertas:

/é/ em pé /ó/ em cipó

fechadas:

/ê/ em vê /u/ em cru

Ressonância nas cavidades oral ou nasal:

Quanto à ressonância nas cavidades oral ou nasal, as vogais são classificadas em:
orais:

/a/ em ato /o/ em fogo

nasais:

/ã/ em lã /u/ em mundo

Intensidade:

Quanto à intensidade, as vogais podem ser:
tônicas: apresentam a maior intensidade.

/á/ em pá /u/ em luz

átonas: apresentam intensidade mínima.

/i/ em lição /e/ em mole

2.3.2 Consoantes

Resultam de um fechamento ou de um estreitamento do canal bucal, oferecendo obstáculos à saída do ar.

Quanto ao tipo de obstáculo oposto à corrente de ar, as consoantes podem ser classificadas em [4]:

Oclusivas:

São caracterizadas pela aproximação completa de dois órgãos da boca. Quando ocorre o afastamento desses órgãos, o ar acumulado sai rapidamente, ocasionando um ruído seco.

/p/ em pala	/b/ em bala
/t/ em tão	/d/ em dão
/k/ em cola	/g/ em gola

Podem ser sonoras ou não sonoras.

Fricativas:

Resultam da aproximação incompleta de dois órgãos. Devido à essa obstrução o ar comprime-se e produz um ruído comparável a uma fricção.

/f/ em fada	/s/ em sela
/x/ em xá	/j/ em já
/v/ em vala	/z/ em zela

Podem ser sonoras ou não sonoras.

Laterais:

Embora exista obstrução à passagem da corrente de ar, esta escoa pelos lados do canal bucal.

/l/ em lua	/lh/ em ilha

Vibrantes:

Acarretam vibrações da língua.

/r/ em reza	/r/ em era
/rr/ em carro	

2.3.3 Semi-vogais

São os fonemas /i/ e /u/ quando formam uma sílaba com uma vogal .

pai mau

2.4 MODELO PARA PRODUÇÃO DE VOZ

Devido à relativa independência entre o trato vocal e as fontes de excitação, os mesmos podem ser representados separadamente em um modelo para a produção de voz. Dessa forma, o sistema de produção de voz pode ser representado pelo seguinte modelo [2]: um sistema linear variante no tempo e um gerador de excitação. A figura 2.6 mostra o diagrama em blocos desse modelo.

O gerador de excitação gera um sinal periódico para representar os sons sonoros e um ruído branco para representar os sons não sonoros (fricativos e explosivos).

O sistema linear variante no tempo modela as ressonâncias do trato vocal e os efeitos da radiação nos lábios. Pode ser modelado usando tubos acústicos ou filtros digitais.

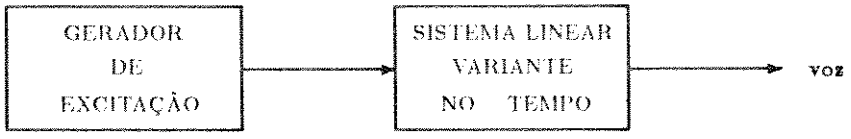


Figura 2.6: Diagrama em blocos do modelo para produção de voz.

2.4.1 Gerador de Excitação

Os sons sonoros são produzidos excitando-se o trato vocal com um sinal quase periódico. Um modelo para a geração desse sinal está representado na figura 2.7.

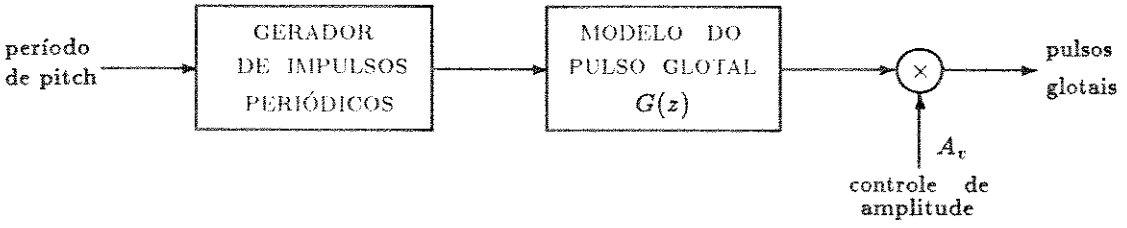


Figura 2.7: Representação da geração do sinal de excitação para os sons sonoros.

Nesse modelo, um gerador de impulsos produz um trem de impulsos unitários na freqüência fundamental do sinal. Esses impulsos vão excitar um sistema linear cuja resposta impulsiva é igual à forma de onda dos pulsos glotais. A forma de onda dos pulsos glotais [2] pode se representada por:

$$g(n) = \begin{cases} 1/2[1 - \cos(\pi n/N_1)] & \text{se } 0 \leq n \leq N_1 \\ \cos(\pi(n - N_1)/2N_2) & \text{se } N_1 \leq n \leq N_1 + N_2 \\ 0 & \text{caso contrário} \end{cases} \tag{2.1}$$

Como $g(n)$ tem comprimento finito, sua transformada Z apresenta apenas zeros.

Uma outra representação para $g(n)$ pode ser obtida utilizando-se um modelo com dois pólos para representar $G(z)$ [6]. Assim, tem-se:

$$G(z) = \frac{1}{(1 - e^{-cT} z^{-1})^2} \tag{2.2}$$

No domínio da freqüência, o pulso glotal introduz um efeito passa-baixas, apresentando um espectro cuja envoltória decresce 12 dB por oitava. A intensidade do pulso glotal é controlada por um controle de ganho.

Os sons não sonoros são produzidos obstruindo-se total ou parcialmente o fluxo de ar em algum ponto do trato vocal. A geração desses sons pode ser modelada utilizando-se uma fonte de ruído com espectro plano e um controle de ganho para controlar a intensidade do mesmo. Esse ruído irá excitar o trato vocal para a produção dos sons não sonoros. Essa fonte pode ser representada por um gerador de números aleatórios com variância unitária e média zero.

2.4.2 Trato Vocal

O trato vocal pode ser modelado como uma associação em cascata de tubos com área de seção transversal variável. A freqüência de ressonância de cada tubo corresponde a um formante. Desprezando-se os efeitos da radiação nos lábios, a função de tranferência do trato vocal pode ser representada pela equação [2]:

$$V(z) = \frac{G}{\prod_{i=1}^{N}(1 - p_i z^{-1})} \tag{2.3}$$

onde o ganho G está associado à amplitude do sinal de voz e p_i, $i = 1, ..., N$ são os pólos de $V(z)$. Esses pólos modelam as freqüências de ressonância do trato vocal.

Esse modelo com apenas pólos é uma boa representação para a maioria dos sons. Isso é devido ao fato de que na produção da maioria dos sons, o trato vocal apresenta apenas ressonâncias. Na produção dos sons nasais e fricativos, além de ressonâncias, o trato vocal apresenta anti-ressonâncias. Para uma representação precisa desses sons, a função de transferência $V(z)$ deve apresentar pólos e zeros. Uma boa aproximação, é aumentar o número de pólos, pois o efeito dos zeros na função de transferência pode ser conseguido, de uma maneira aproximada, aumentando-se o número de pólos [5].

Assim, o trato vocal pode ser representado por um sistema linear, cuja função de transferência apresenta apenas pólos.

Como o trato vocal é um sistema estável, todos os pólos de $V(z)$ estão dentro do círculo unitário.

2.4.3 Radiação

O efeito de radiação nos lábios pode ser modelado por um filtro passa-altas. Nas baixas freqüências, a abertura entre os lábios funciona como uma superfície irradiadora, ocorrendo uma difração das ondas sonoras. Nessas freqüências a impedância de radiação aproxima-se de um curto-circuito ideal.

Uma boa aproximação para a resposta em freqüência associada a essa radiação é dada pela equação [2]:

$$R(z) = R_0(1 - z^{-1}) \tag{2.4}$$

a qual apresenta um ganho de 6 dB por oitava.

2.4.4 Modelo completo

O modelo completo que representa o sistema vocal para a produção da voz está representado na figura 2.8.

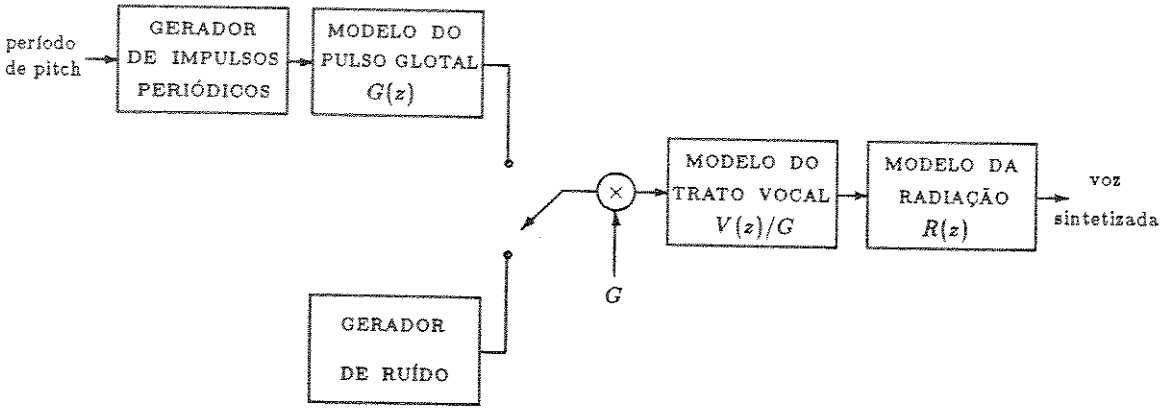


Figura 2.8: Modelo completo para a produção de voz.

Combinando as funções de transferência do pulso glotal, trato vocal e radiação em uma única função de transferência, o modelo pode ser simplificado. A função de transferência obtida é dada pela equação:

$$H(z) = G(z).V(z).R(z) \tag{2.5}$$

Dessa forma, obtém-se um modelo simples para a produção de voz. Este modelo é constituído por um sistema linear variante no tempo, e dois tipos de fontes para gerar o sinal de excitação desse sistema: um gerador de impulsos unitários periódicos com período variável para simular os sons sonoros e um gerador de ruído de faixa larga para simular os sons não sonoros.

Utilizando um modelo com apenas pólos para representar o sistema linear, a função de transferência $H(z)$ do mesmo pode ser escrita como:

$$H(z) = \frac{G}{1 - \sum_{k=1}^{P} a_k z^{-k}} \tag{2.6}$$

Para representar a natureza variante no tempo do sinal de voz, os coeficientes de $H(z)$ e o sinal de excitação são atualizados em intervalos de tempo regulares. Na saída de $H(z)$ obtém-se a voz sintetizada. O modelo simplificado está esquematizado na figura 2.9.

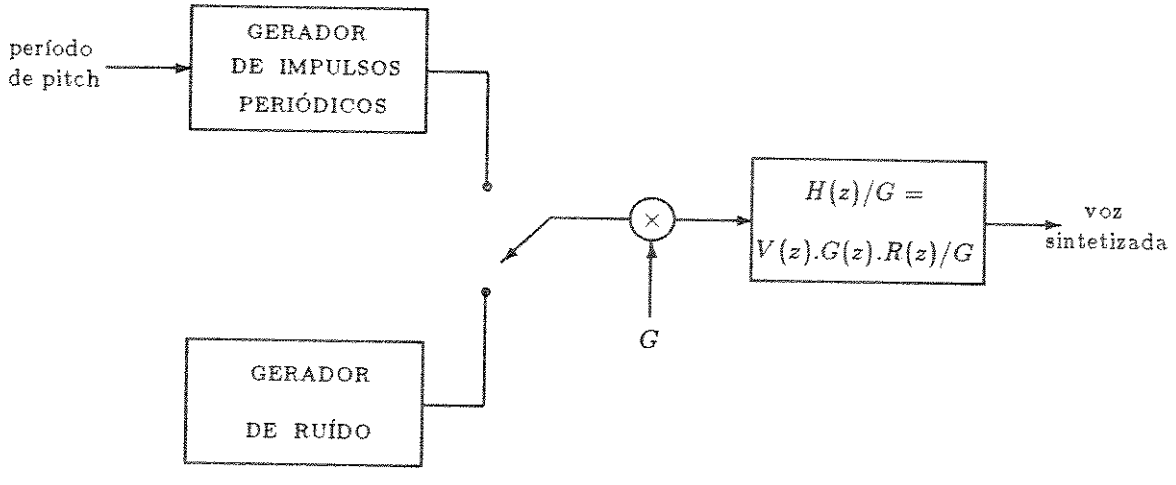


Figura 2.9: Modelo simplificado para a produção de voz.

2.4.5 Imperfeições do modelo

Esse modelo representa bem os sons que variam lentamente, como as vogais, mas não é bom para representar sons transitórios, como os explosivos. Outra falha do modelo está no fato de $H(z)$ não possuir zeros e assim não ser possível uma boa simulação dos sons nasais. Além disso, na geração dos sons fricativos sonoros e oclusivos sonoros, o sinal de excitação é constituído por uma combinação entre pulsos periódicos e ruído, e isto não ocorre neste modelo. A ocorrência de uma forma de excitação exclui a outra.

Mesmo com essas imperfeições pode-se obter com esse modelo, voz sintetizada com boa qualidade e alta inteligibilidade, embora ocorra perda da naturalidade da voz.

2.5 VOCODERS

Os codificadores de voz que utilizam características do espectro de voz como o período de pitch, formantes e outros, para sintetizarem voz, são denominados vocoders. O termo vocoder é uma abreviação de ' voice coder ' [6].

Esses codificadores conseguem reduzir as taxas de bits para a síntese da voz, e embora ocorra uma diminuição na qualidade da voz sintetizada, a inteligibilidade da mesma ainda é mantida.

Um dos principais problemas dos vocoders é a dificuldade em obter uma correta separação entre o comportamento dos formantes e o comportamento do pitch. Com a utilização da técnica de predição linear, a separação das características do trato vocal e da fonte glotal pode ser equacionada, produzindo um resultado eficaz. O **Vocoder LPC**, o qual utiliza essa técnica, será apresentado no próximo capítulo.

Além do vocoder LPC, outros vocoders foram propostos utilizando diferentes técnicas. Como exemplo pode-se citar [1]:

Vocoder de Canal

Foi proposto por H. Dudley em 1939. A análise espectral é realizada usando-se um banco de filtros passa-faixa.

Vocoder de Fase

Foi proposto por J. L. Flanagan e a análise espectral utiliza um banco de filtros passa-faixa.

Vocoder Homomórfico

Realiza a análise do cepstrum e foi proposto por A. V. Oppenheim em 1969.

O principal responsável pela degradação (perda da naturalidade) da voz sintetizada pelos vocoders é o modelo de excitação utilizado pelos mesmos, o qual apresenta um detector sonoro/não sonoro. Para conseguir-se sintetizar voz com melhor qualidade e evitar-se a utilização desse detector, foram propostos sistemas que transmitem os coeficientes do filtro do trato vocal e as componentes de baixa freqüência do sinal de voz original, as quais serão usadas para gerar o sinal de excitação no sintetizador. Esses sistemas são denominados ' **Voice Excited Coders** '. Também foram propostos sistemas que transmitem o erro de predição ou variações do mesmo, para gerar o sinal de excitação do filtro de síntese. Esses sistemas são chamados ' **Residual Excited Coders** '. Os sistemas mais complexos de codificação residual utilizam o método análise por síntese.

Capítulo 3

VOCODER LPC

3.1 INTRODUÇÃO

O vocoder LPC (Linear Predictive Coding) utiliza o modelo de produção de voz apresentado no capítulo anterior. Esse modelo [2] é composto por um filtro digital variante com o tempo, uma chave de seleção sonoro/não sonoro, um detector de pitch e sinais de excitação sonoros/não sonoros. O filtro digital é constituído apenas por pólos (modelo AR), representando as características do trato vocal, a forma do pulso glotal e os efeitos da radiação nos lábios. A excitação dos sons sonoros é representada por um trem de impulsos periódicos e a excitação dos sons não sonoros por ruído branco. Os coeficientes do filtro digital são calculados usando-se a técnica de predição linear, a qual é um método preciso e confiável para a estimação desses parâmetros. Os coeficientes do filtro calculados por essa técnica são denominados coeficientes LPC.

3.2 PRINCÍPIOS DA TÉCNICA DE PREDIÇÃO LINEAR

Com relação ao modelo de produção de voz, seja $H(z)$ a transformada Z da resposta impulsiva do filtro digital, $U(z)$ a transformada Z da excitação $u(n)$, $S_o(z)$ a transformada Z do sinal de voz sintetizado $s_o(n)$ e G o ganho da excitação. Pode-se escrever:

$$S_o(z) = U(z).H(z) \tag{3.1}$$

Como $H(z)$ apresenta apenas pólos , tem-se :

$$H(z) = \frac{G}{1 - \sum_{k=1}^{P} a_k z^{-k}} \tag{3.2}$$

onde a_k são os coeficientes do filtro .

Dessa forma , o sinal de voz sintetizado é dado pela equação :

$$s_o(n) = \sum_{k=1}^{P} a_k s_o(n-k) + Gu(n) \tag{3.3}$$

O sinal de voz sintetizado é um processo autorregressivo de ordem P pois a amostra atual é obtida pela combinação do sinal de excitação com P amostras anteriores do sinal de voz sintetizado.

Um preditor linear [2] é um sistema que estima o valor da amostra atual a partir de uma combinação linear dos valores das amostras anteriores. Definindo-se $s(n)$ como o sinal de voz original e $\hat{s}(n)$ como uma estimativa desse sinal, obtida com um preditor linear de ordem P, na saída do preditor linear tem-se :

$$\hat{s}(n) = \sum_{k=1}^{P} a_k s(n-k) \tag{3.4}$$

O erro de predição é definido como a diferença entre o sinal $s(n)$ e o seu valor estimado $\hat{s}(n)$.

$$e(n) = s(n) - \hat{s}(n) \tag{3.5}$$

$$e(n) = s(n) - \sum_{k=1}^{P} a_k s(n-k) \tag{3.6}$$

Seja $E(z)$ a transformada Z de $e(n)$ e $S(z)$ a transformada Z de $s(n)$. A partir da equação 3.6 pode-se escrever:

$$A(z) = \frac{E(z)}{S(z)} \tag{3.7}$$

$$A(z) = 1 - \sum_{k=1}^{P} a_k z^{-k} \tag{3.8}$$

$A(z)$ é a função de transferência do filtro do erro de predição, o qual é denominado filtro inverso. O erro de predição é também chamado de resíduo [2].

Caso o modelo de produção de voz apresentado fosse exato, $s_o(n)$ seria igual a $s(n)$.

$$s_o(n) = s(n) \tag{3.9}$$

Assim, o erro de predição seria igual ao sinal de excitação.

$$e(n) = Gu(n) \tag{3.10}$$

Porém, como esse modelo para a produção de voz não é perfeito , existe uma diferença entre o erro $e(n)$ e a excitação $Gu(n)$. Em um modelo que utilize o sinal de resíduo como excitação do filtro $H(z)/G$, o sinal de voz sintetizado $s_o(n)$ será igual ao sinal original $s(n)$.

A obtenção de um preditor para um sinal de voz envolve a determinação dos coeficientes a_k do mesmo, os quais são obtidos a partir do sinal de voz. O cálculo desses parâmetros pode ser feito utilizando-se o método dos mínimos quadrados. Esse método minimiza o erro quadrático médio do sinal de voz em relação a cada um do coeficientes a_k. Utilizando-se esse método tem-se duas principais técnicas: o método da autocorrelação, no qual é feito um janelamento do sinal de voz e o erro é minimizado no intervalo $0 \leq n \leq (N+P-1)$, e o método da covariância, no qual é feito um janelamento do erro, sendo este calculado no intervalo $0 \leq n \leq N-1$, onde N é o comprimento da janela e P é a ordem do preditor. Os coeficientes LPC também podem ser obtidos pelo método treliça, o qual permite a atualização instantânea dos coeficientes. Esse método é o menos eficiente dos três em termos computacionais. Computacionalmente, o método da autocorrelação é mais eficiente que o método da covariância, devido ao fato deste último requerer um maior número de multiplicações para a resolução das equações matriciais. No método de covariância, a estabilidade do preditor não é garantida, enquanto nos outros métodos, teoricamente a estabilidade é garantida, e pode ser verificada analisando-se os coeficientes parcor [2].

A figura 3.1 mostra a forma de onda do sinal de voz original, do sinal estimado pelo preditor e do erro de predição para um segmento sonoro de voz. Esses sinais foram obtidos utilizando-se um preditor de ordem 12.

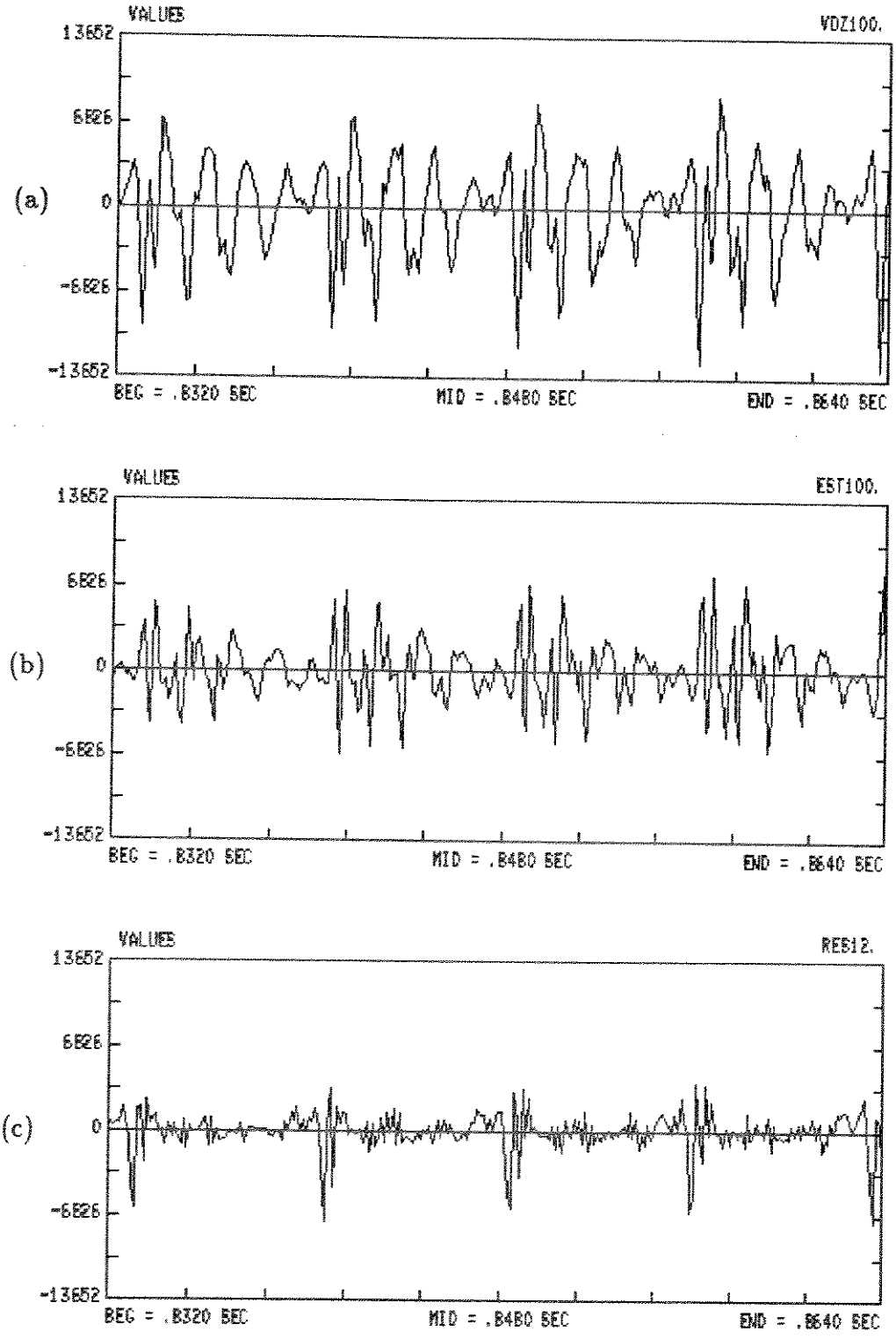


Figura 3.1: (a) sinal de voz original. (b) sinal estimado. (c) erro de predição. O sinal estimado e o erro de predição foram obtidos usando-se um preditor de ordem 12.

3.3 MÉTODO DA AUTOCORRELAÇÃO

Seja $s(n)$ o sinal de voz e $w(n)$ uma janela de comprimento finito, a qual assume o valor zero fora do intervalo $0 \leq n \leq N-1$. O sinal para o qual serão calculados os coeficientes LPC é o resultado da multiplicação de $s(n)$ por $w(n)$.

$$x(n) = s(n).w(n) \tag{3.11}$$

O sinal $x(n)$ será igual a zero fora do intervalo $0 \leq n \leq N-1$.

O erro de predição de $x(n)$ pode ser escrito como [2]:

$$e(n) = x(n) - \sum_{k=1}^{P} a_k x(n-k) \tag{3.12}$$

A energia do erro de predição de $x(n)$ é definida como [2]:

$$\alpha_r^2 = \sum_{n=-\infty}^{+\infty} e_n^2(n) \tag{3.13}$$

$$\alpha_r^2 = \sum_{n=-\infty}^{+\infty} [x(n) - \sum_{k=1}^{P} a_k x(n-k)]^2 \tag{3.14}$$

Deseja-se encontrar os valores dos coeficientes a_k que minimizem α_r^2. Isso pode se feito, tomando-se as derivadas parciais de α_r^2 em relação a cada um dos coeficientes a_k e igualando-as a zero.

$$\frac{\partial \alpha_r^2}{\partial a_i} = 0 \quad \text{para} \quad i = 1, \ldots, P \tag{3.15}$$

Isso resulta em P equações lineares:

$$\sum_{n=-\infty}^{+\infty} x(n-i)x(n) = \sum_{k=1}^{P} a_k \sum_{n=-\infty}^{+\infty} x(n-i)x(n-k) \quad \text{para} \quad i = 1, \ldots, P \tag{3.16}$$

Como $x(n)$ tem duração finita pode-se escrever:

$$\sum_{n=-\infty}^{+\infty} x(n)x(n-i) = \sum_{n=0}^{N-1-i} x(n)x(n+i) \quad \text{para } i = 1, \ldots, P \tag{3.17}$$

$$\sum_{n=-\infty}^{+\infty} x(n-i)x(n-k) = \sum_{n=0}^{N-1-(k-i)} x(n)x(n+(k-i)) \quad \text{para} \quad (k-i) > 0 \tag{3.18}$$

$$\sum_{n=-\infty}^{+\infty} x(n-i)x(n-k) = \sum_{n=0}^{N-1-(i-k)} x(n)x(n+(i-k)) \quad \text{para} \quad (i-k) > 0 \tag{3.19}$$

A função de autocorrelação a curto prazo de $x(n)$ [2] é definida por:

$$R(i) = \sum_{n=0}^{N-1-i} x(n)x(n+i) \tag{3.20}$$

Avaliando-se a função de autocorrelação para $(i-k)$ tem-se:

$$R(i-k) = \sum_{n=0}^{N-1-(i-k)} x(n)x(n+(i-k)) \tag{3.21}$$

A partir das equações 3.18, 3.19 e 3.21 e como a função de autocorrelação $R(i)$ é uma função par, pode-se escrever:

$$\sum_{n=-\infty}^{+\infty} x(n-i)x(n-k) = R(|i-k|) \tag{3.22}$$

Usando-se as equações 3.17, 3.20 e 3.22 as equações 3.16 podem ser escritas da seguinte forma:

$$\sum_{k=1}^{P} a_k R(|i-k|) = R(i) \quad \text{para } i = 1, \ldots, P \tag{3.23}$$

Essas equações podem também serem escritas na forma matricial:

$$Ra = r \tag{3.24}$$

onde: R é uma matriz $P \times P$ de elementos $R(i,k) = R(|i-k|)$
r é o vetor coluna $(R(1), R(2), \ldots, R(P))$
a é o vetor coluna dos coeficientes LPC $(a_1, a_2, \ldots, a_p)$.

$$\begin{pmatrix} R(0) & R(1) & \ldots & R(P-1) \\ R(1) & R(0) & \ldots & R(P-2) \\ \vdots & & & \vdots \\ R(P-1) & R(P-2) & \ldots & R(0) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_P \end{pmatrix} = \begin{pmatrix} R(1) \\ R(2) \\ \vdots \\ R(P) \end{pmatrix} \tag{3.25}$$

Essas equações são conhecidas como equações de Yule-Walker ou equações Normais.

Os coeficientes a_k obtidos resolvendo-se as equações 3.23 são os coeficientes do preditor ótimo, para o qual a energia residual é mínima. A energia residual mínima é obtida substituindo-se a equação 3.23 em 3.14, resultando em:

$$\alpha^2 = R(0) - \sum_{k=1}^{P} a_k R(k) \tag{3.26}$$

3.4 SOLUÇÃO DAS EQUAÇÕES LPC

Um dos mais eficientes métodos para a resolução dessas equações é o procedimento recursivo de Durbin [2], o qual utiliza a natureza Toeplitz da matriz R.

Esse método consiste em resolver recursivamente para $i = 1, \ldots, P$ as seguintes equações:

$$k_i = [R(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j)]/E^{(i-1)} \tag{3.27}$$

$$a_i^{(i)} = k_i \tag{3.28}$$

$$a_j^{(i)} = a_j^{(i-1)} - k_i a_{(i-j)}^{(i-1)} \qquad \text{para} \quad j = 1, \ldots, (i-1) \tag{3.29}$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)} \tag{3.30}$$

A solução dessas equações é dada por:

$$a_j = a_j^P \qquad \text{para } 1 \leq j \leq P \tag{3.31}$$

A condição inicial para esse procedimento é $E^{(0)} = R(0)$.

A cada iteração i, os coeficientes a_k $(k = 1, \ldots, i)$ obtidos representam o preditor linear ótimo de ordem i.

Os parâmetros k_i são denominados coeficientes de correlação parcial (parcor) e apresentam magnitude menor ou igual a 1.

$$-1 \leq k_i \leq 1$$

Para valores dos coeficientes parcor dentro desse intervalo é garantida a estabilidade do filtro ótimo de predição [2]. Os coeficientes parcor correspondem ao negativo dos coeficientes de reflexão.

3.5 ORDEM DO PREDITOR

A ordem do filtro do modelo LPC depende da precisão espectral desejada, levando-se em conta a largura de faixa do sinal de voz. Em geral, a ordem do filtro deve ser escolhida de forma a tornar possível a representação de todos os formantes presentes na largura de faixa do sinal, e também de modo que se consiga uma boa representação para os sons nasais utilizando-se um filtro com apenas pólos. O espectro do sinal de voz pode ser representado como tendo um par de pólos complexos por kHz [2]. Assim, amostrando-se o sinal com uma freqüência F_s kHz, necessita-se de F_s polos para representar o espectro de voz. A esse número deve-se somar de 2 a 4 pólos, os quais são necessários para simular os zeros dos sons nasais [2].

Utilizando-se um modelo com maior número de pólos ocorrerá uma melhor representação dos formantes e o sinal estimado estará mais próximo do original.

A figura 3.2 mostra um sinal de voz sonoro e seu espectro.

As figuras 3.3, 3.4, 3.5, 3.6 e 3.7 mostram o espectro do sinal de voz original e o espectro LPC para ordens iguais a 4, 8, 12 , 20 e 60 respectivamente.

As figuras 3.8, 3.9, 3.10, 3.11 e 3.12 mostram o sinal erro de predição para ordens do filtro preditor iguais a 4, 8, 12, 20 e 60 respectivamente.

Aumentando-se a ordem do preditor, o espectro deste torna-se mais próximo do espectro do sinal original. Para preditores com ordens próximas a F_s, o espectro do preditor é aproximadamente a envoltória do espectro do sinal de entrada. Para preditores com ordem muito elevada, o espectro do mesmo aproxima-se do espectro do sinal, passando a incorporar também as características da excitação, o que não é desejado nos vocoders.

Utilizando-se uma freqüência de amostragem de 8 kHz, preditores com ordem entre 8 e 12 conseguem apresentar um espectro razoavelmente próximo da envoltória do espectro do sinal original. Neste trabalho, após uma série de testes, resolveu-se utilizar um preditor de ordem 8.

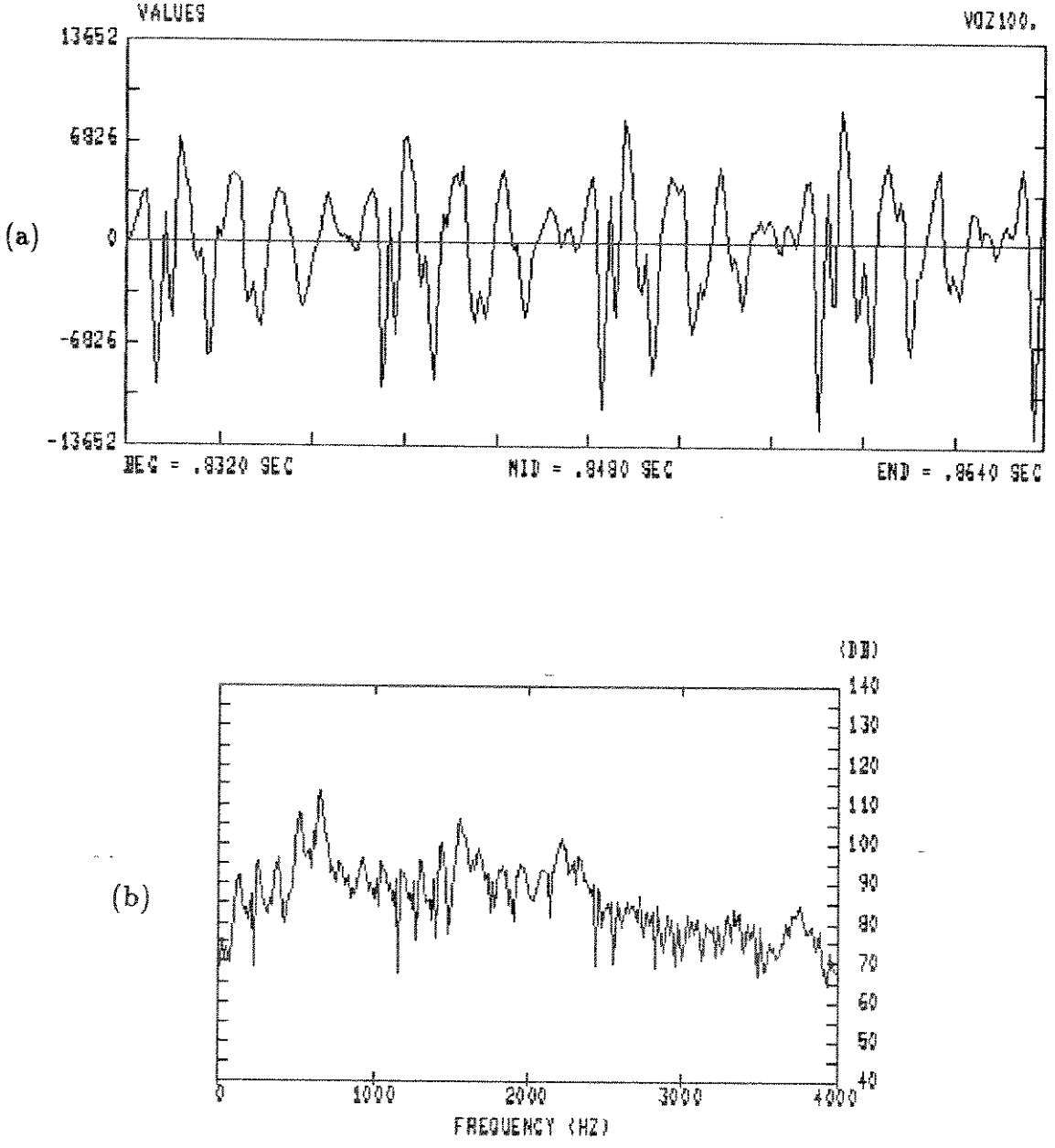


Figura 3.2: (a) sinal de voz sonoro. (b) espectro do sinal de voz.

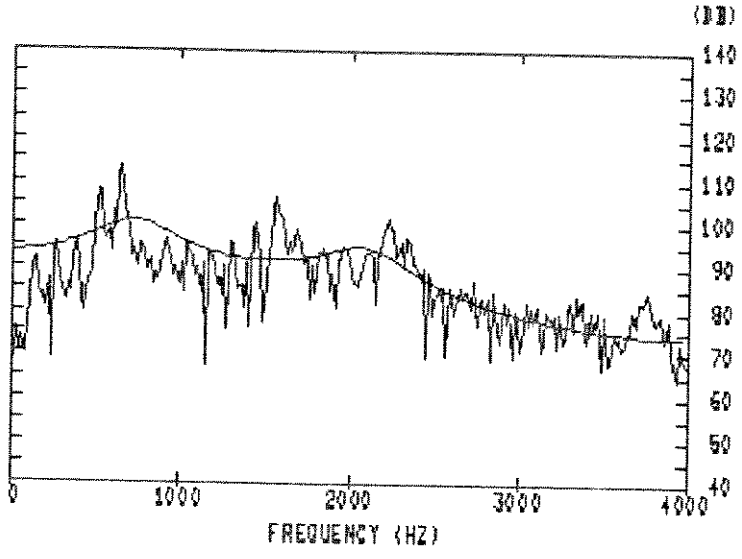


Figura 3.3: espectro do sinal de voz original e espectro LPC para P=4.

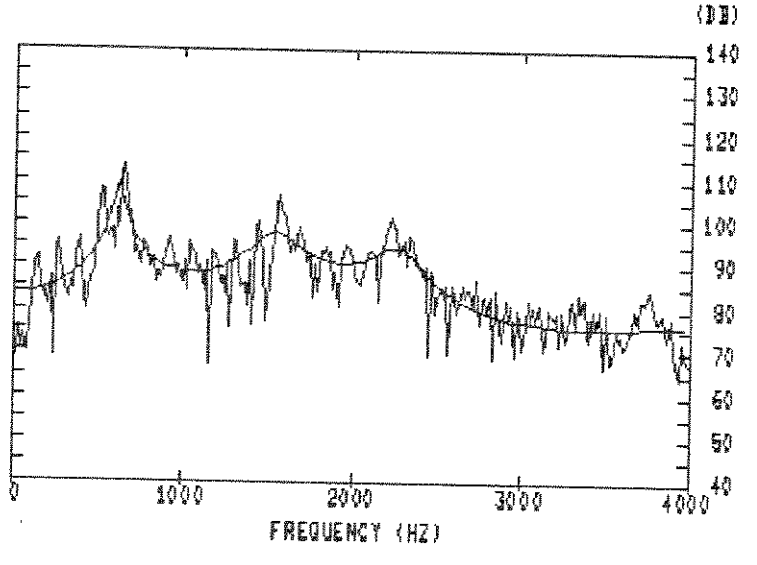


Figura 3.4: espectro do sinal de voz original e espectro LPC para P=8.

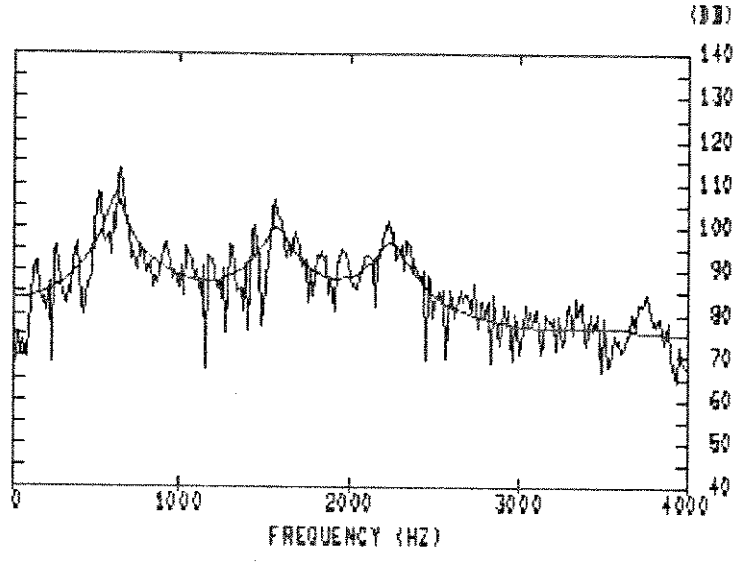


Figura 3.5: espectro do sinal de voz original e espectro LPC para P=12.

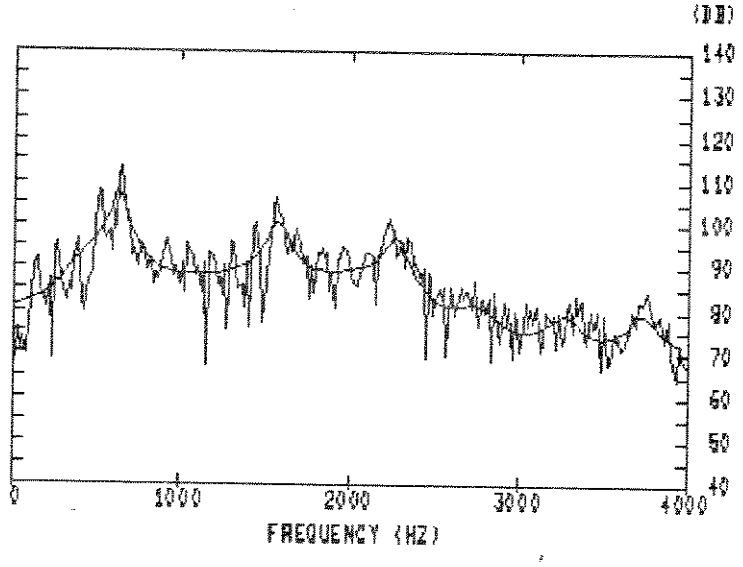


Figura 3.6: espectro do sinal de voz original e espectro LPC para P=20.

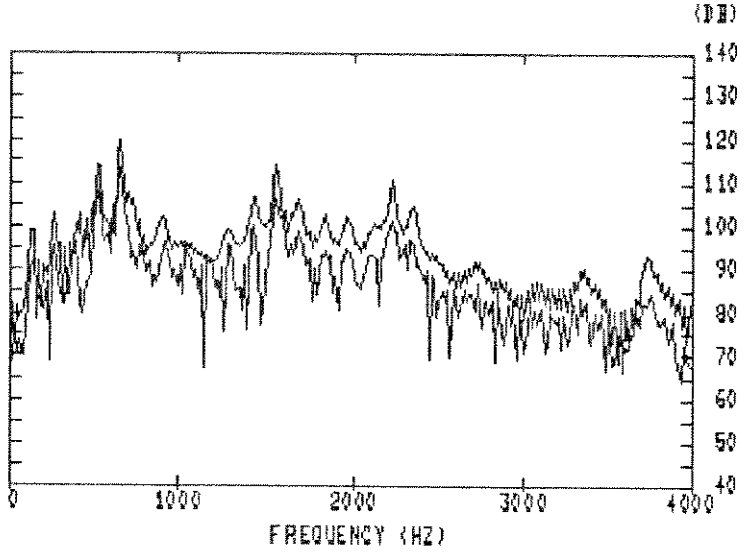


Figura 3.7: espectro do sinal de voz original e espectro LPC para P=60.

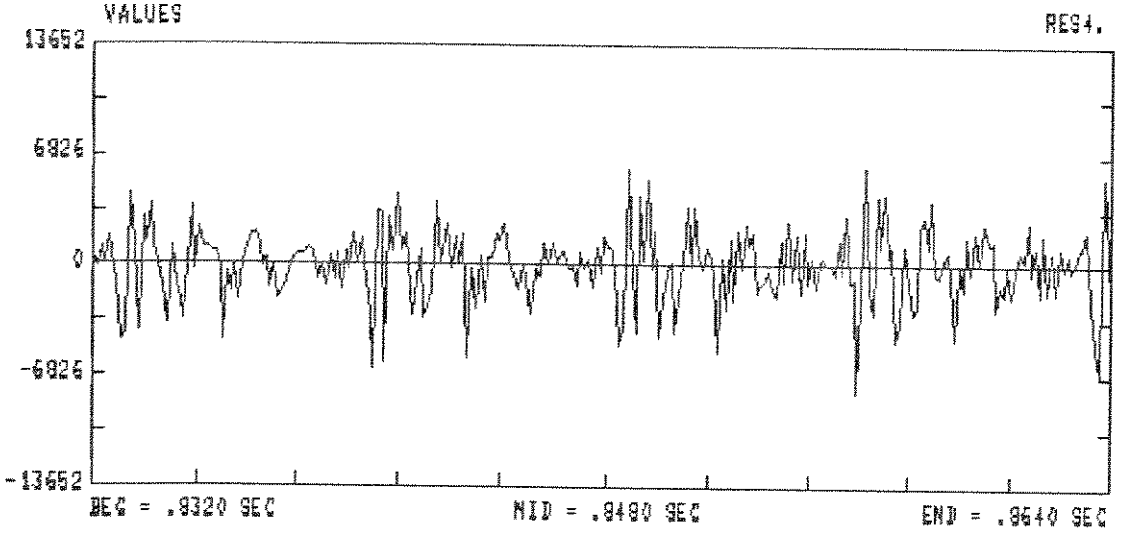


Figura 3.8: erro de predição para preditor de ordem 4.

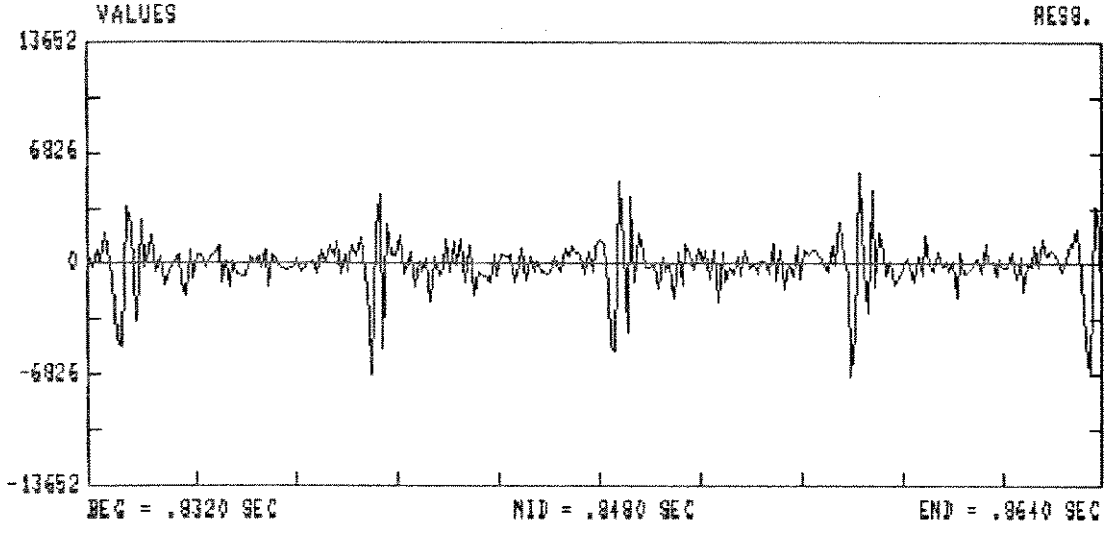


Figura 3.9: erro de predição para preditor de ordem 8.

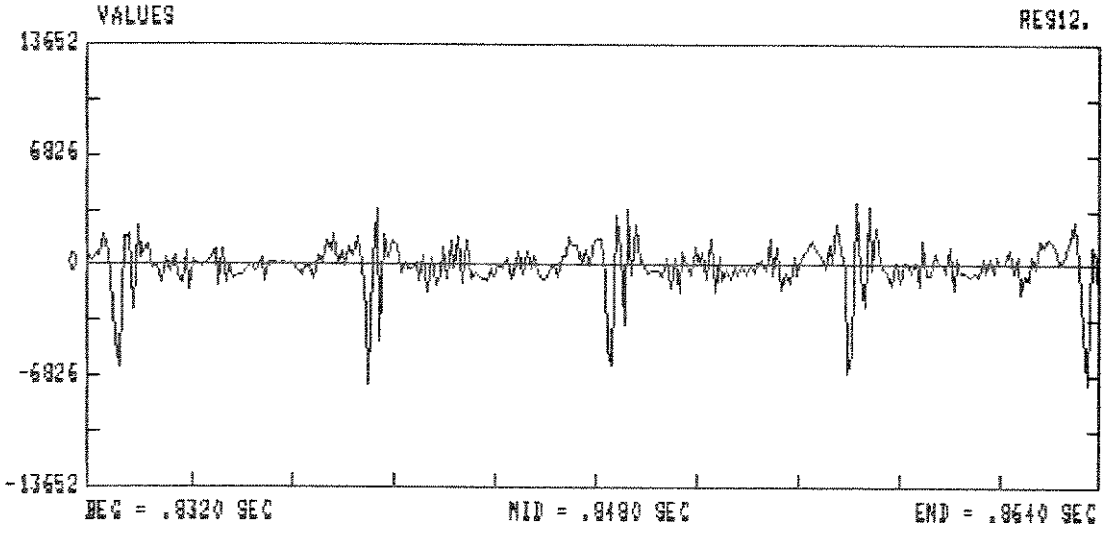


Figura 3.10: erro de predição para preditor de ordem 12.

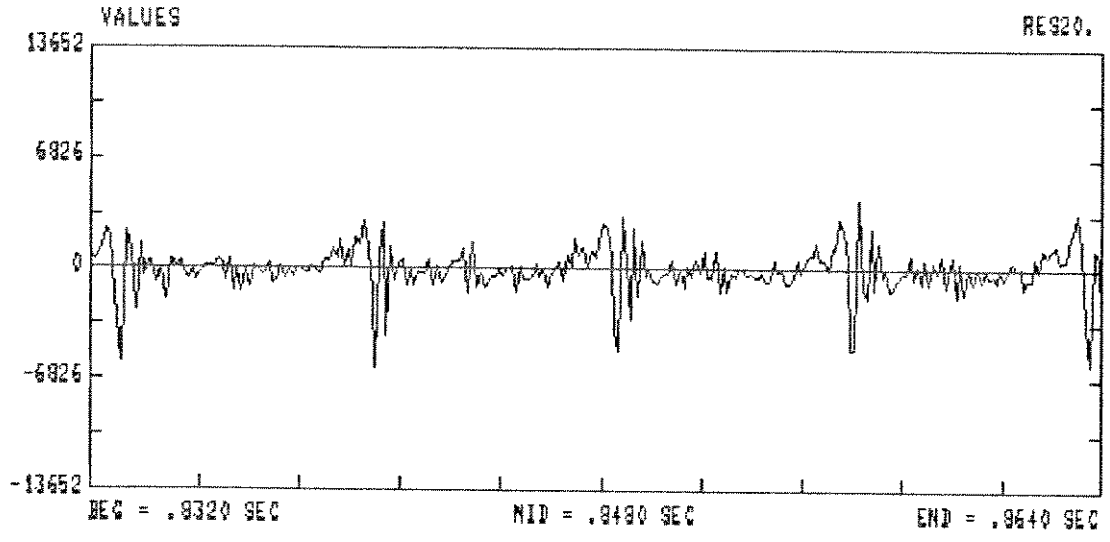


Figura 3.11: erro de predição para preditor de ordem 20.

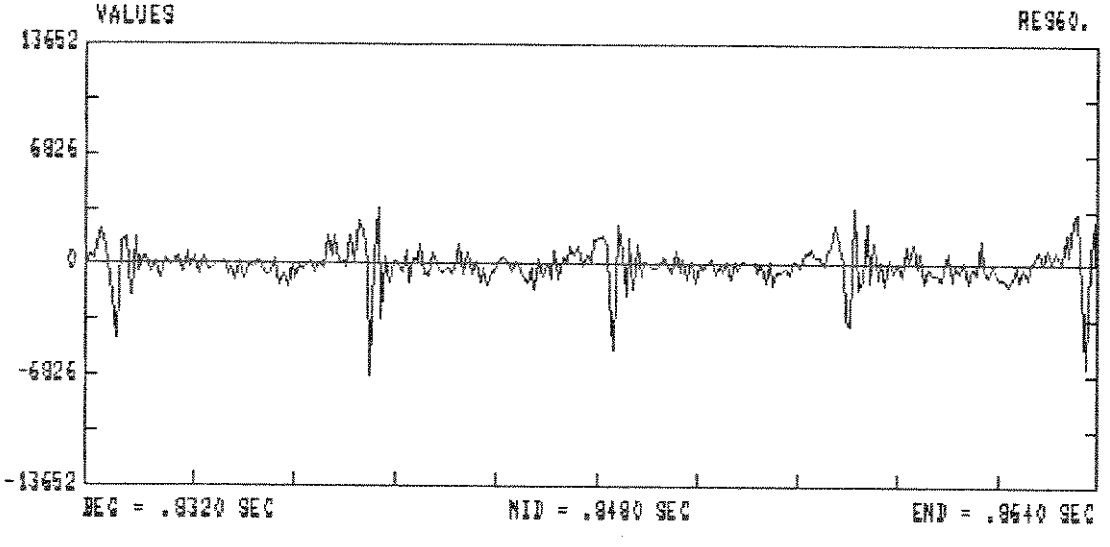


Figura 3.12: erro de predição para preditor de ordem 60.

3.6 INTERVALO DE ANÁLISE E TIPO DE JANELA

A duração do intervalo de análise ou quadro de análise, deve ser escolhida de forma que o sinal de voz permaneça estacionário dentro do mesmo. Para que seja obtida uma boa estimativa espectral dos sons sonoros, o intervalo deve incluir aproximadamente dois períodos de pitch. Intervalos longos devem ser evitados devido à não estacionaridade do sinal de voz. Tipicamente tem-se usado intervalos de análise com duração de 20 a 30 ms [5].

No método da autocorrelação para cálculo dos coeficientes LPC, ocorre um janelamento do sinal, sendo essencial o uso de uma janela com transição gradual para evitar um elevado erro de predição nos extremos da janela. Esse erro é devido ao fato que fora da janela o sinal assume o valor zero. Assim, no início da janela tenta-se estimar um valor diferente de zero a partir de zeros. Analogamente, no final da janela, tenta-se estimar zeros a partir de valores diferentes de zero. Isso ocorre com o uso da janela retangular [2], a qual é representada pela equação :

$$r(n) = \begin{cases} 1 & \text{se } 0 \leq n \leq N-1 \\ 0 & \text{caso contrário} \end{cases} \tag{3.32}$$

Uma forma de minimizar o erro de predição nas extremidades é a utilização da janela de Hamming. Essa janela dá maior ênfase às amostras localizadas no centro da janela. A janela de Hamming [2] é representada pela equação :

$$h(n) = \begin{cases} 0.54 - 0.46\cos(2\pi n/(N-1)) & \text{se } 0 \leq n \leq N-1 \\ 0 & \text{caso contrário} \end{cases} \tag{3.33}$$

Como as amostras localizadas nas extremidades da janela de Hamming são atenuadas, o erro de predição nas extremidades é menor.

Uma forma de se evitar grandes flutuações dos parâmetros calculados, é o uso da superposição de intervalos de análise adjacentes. Nesse caso, uma parte das amostras finais do intervalo anterior e uma parte das amostras iniciais do intervalo posterior fazem parte do intervalo atual de análise. Assim, os parâmetros calculados em cada intervalo de análise são influenciados pelas amostras dos intervalos adjacentes.

Devido à forma da janela de Hamming e ao uso da superposição de intervalos, pode-se usar intervalos de análise maiores que os usados levando-se em consideração apenas a estacionaridade do sinal de voz.

Testes subjetivos mostraram que usando-se a janela retangular a degradação da voz sintetizada é bastante elevada e que a superposição dos intervalos no cálculo dos coeficientes LPC melhora a qualidade da voz sintetizada.

A figura 3.13 mostra a forma das janelas retangular e Hamming, e a figura 3.14 mostra o formato das janelas com superposição para cálculo dos coeficientes LPC.

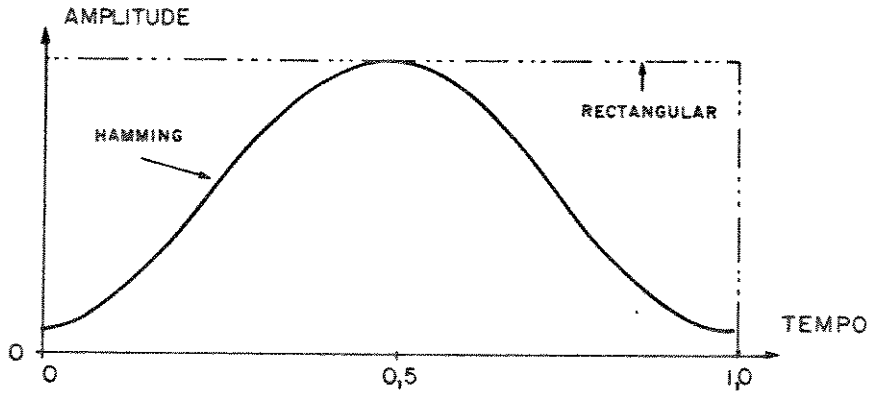


Figura 3.13: forma das janelas retangular e Hamming.

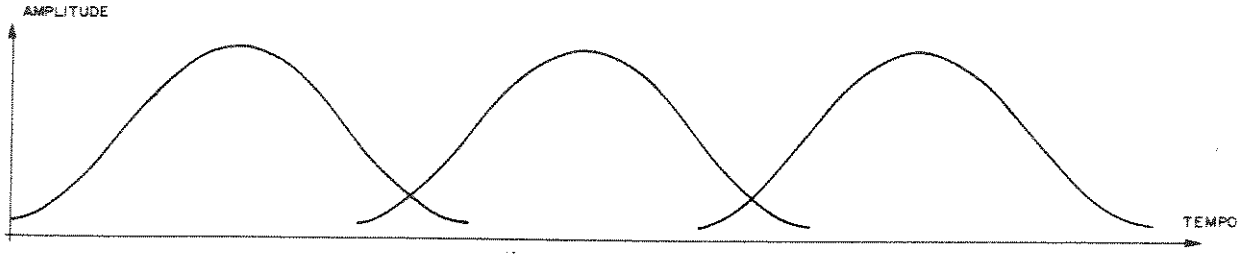


Figura 3.14: superposição das janelas para cálculo dos coeficientes LPC.

3.7 PRÉ-ÊNFASE

No modelo de produção da voz tem-se: pulso glotal (para sons sonoros), trato vocal e radiação nos lábios. O espectro do pulso glotal apresenta uma queda de 12 dB por oitava, e o espectro da radiação um ganho de 6 dB por oitava. A finalidade da pré-ênfase é dar um ganho de 6 dB por oitava para compensar o efeito combinado do pulso glotal e da radiação. Normalmente a pré-ênfase é realizada antes do janelamento do sinal [6], e consiste em passar o sinal de voz por um filtro FIR de primeira ordem com transformada Z dada por $1 - \mu z^{-1}$. Valores de μ maiores ou iguais a 0.9 e memores que 1.0 apresentam bons resultados [6]. Com a utilização da pré-ênfase o sinal para a análise LPC passa a ser:

$$y(n) = s(n) - \mu s(n-1) \tag{3.34}$$

Embora os sons não sonoros não apresentem a componente do pulso glotal, o efeito da pré-ênfase para esses sons, é uma ênfase nas altas freqüências dos mesmos, não resultando degradação. Uma solução para esse caso, seria o uso de um filtro de pré-ênfase com coeficiente adaptativo dado por [6]:

$$\mu = \frac{R(1)}{R(0)} \tag{3.35}$$

onde $R(n)$ é a função de autocorrelação do sinal de voz $s(n)$.

No presente trabalho utilizou-se pré-ênfase fixa com coeficiente $\mu = 0.9$, não sendo observadas degradações do sinal de voz, mesmo para os sons não sonoros.

Na recepção, após a reconstituição do sinal, este é submetido a uma de-ênfase, através de um filtro IIR dado por $1/(1 - \mu z^{-1})$.

A figura 3.15 mostra o espectro de um segmento sonoro de voz antes da pré-ênfase e a figura 3.16 mostra o espectro do mesmo depois da pré-ênfase.

Devido às considerações acima, a pré-ênfase realiza o ganho de 6 dB por oitava, restando para o preditor modelar apenas o trato vocal. Caso contrário, um pólo real do preditor deveria realizar essa tarefa. Assim, um sistema com pré-ênfase e preditor de ordem P é aproximadamente equivalente a um sistema com preditor de ordem $P + 1$ sem pré-ênfase.

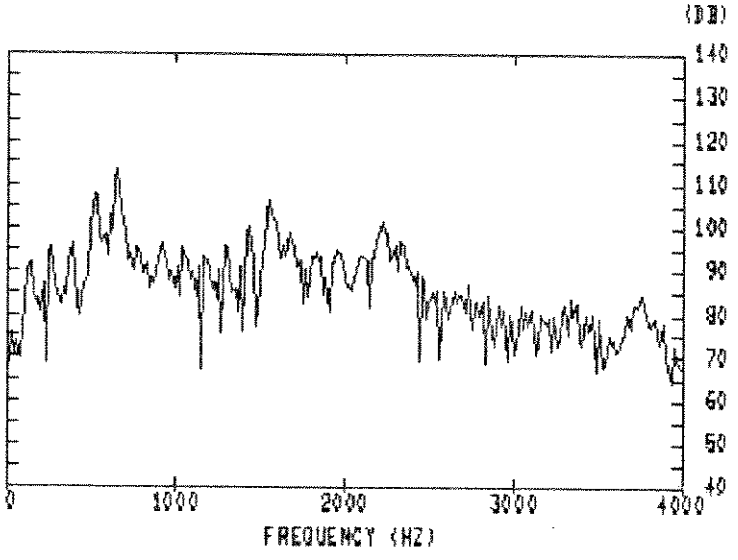


Figura 3.15: espectro do sinal antes da pré-ênfase.

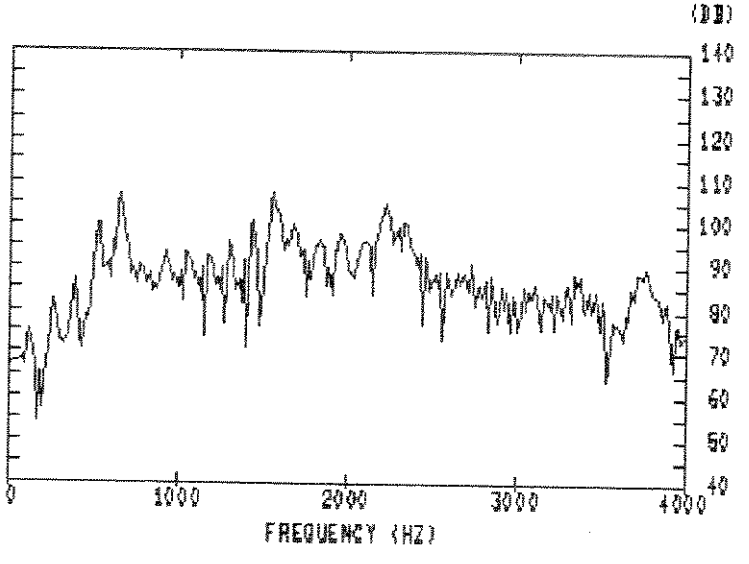


Figura 3.16: espectro do sinal depois da pré-ênfase.

3.8 EXCITAÇÃO DO VOCODER LPC

Em um vocoder LPC o sinal de voz sintetizado é dado pela equação:

$$s_o(n) = \sum_{k=1}^{P} a_k s_o(n-k) + Gu(n) \tag{3.36}$$

O sinal de excitação $u(n)$, o qual excita o filtro $H(z)$ é o seguinte:

- Sons sonoros

  Para sons sonoros é usual se utilizar como excitação uma seqüência de impulsos unitários periódicos, com período igual ao período de pitch. Esse sinal de excitação não apresenta média zero. Quando utiliza-se pré-ênfase, a voz sintetizada pode apresentar um ruído de baixa freqüência, devido ao fato do sinal de excitação não apresentar média zero e essa média ser amplificada pelo alto ganho do filtro de de-ênfase na freqüência zero. Esse ruído torna-se maior quando a freqüência fundamental (pitch) aumenta. Um sinal de excitação com média zero pode ser obtido usando-se o seguinte sinal para excitar o filtro $H(z)$ para os sons sonoros [6]:

$$u(n) = \begin{cases} 1 & \text{para n=0, T, 2T, ...} \\ -1/(T-1) & \text{caso contrário} \end{cases} \tag{3.37}$$

  onde T é o período de pitch.

- Sons não sonoros

  Utiliza-se um sinal de ruído com espectro plano, variância unitária e média zero para excitar o filtro $H(z)$.

3.9 CÁLCULO DO GANHO DA EXCITAÇÃO

Para o cálculo do ganho da excitação é razoável assumir que a energia mínima do erro de predição é igual à energia do sinal de excitação. A energia residual mínima [2] pode ser escrita como:

$$\alpha^2 = R(0) - \sum_{k=1}^{P} a_k R(k) \tag{3.38}$$

e a energia da excitação é dada por:

$$E_n = G^2 \sum_{m=0}^{N-1} u^2(m) \tag{3.39}$$

onde N é o número de amostras do sinal de excitação.

Assumindo-se que para sons sonoros $u(n) = \delta_T(n)$, onde $\delta_T(n)$ é um trem de impulsos periódicos de amplitude unitária e período T, sendo T o período de pitch, tem-se:

$$E_n = G^2 \sum_{m=0}^{N-1} \delta_T(m) \tag{3.40}$$

$$E_n = \frac{G^2 N}{T} \tag{3.41}$$

$$E_n = \alpha^2 \tag{3.42}$$

$$G^2 = \frac{\alpha^2 T}{N} \tag{3.43}$$

Embora a equação 3.43 tenha sido deduzida para uma excitação periódica com média não nula, o seu valor pode ser mantido para a exitação dada pela equação 3.37 com um pequeno erro tolerável.

Para sons não sonoros assume-se que a excitação é ruído branco com média zero e variância unitária. Para um sinal aleatório, tem-se:

$$E[u(n)u(n)] = \sigma_v^2 \tag{3.44}$$

onde: σ_v^2 é a variância do sinal.

Para o ruído branco usado como excitação para os sons não sonoros tem-se $\sigma_v^2 = 1$. Assim, o ganho da excitação é dado por:

$$G^2 = \frac{\alpha^2}{N} \tag{3.45}$$

Portanto, o ganho da excitação pode ser calculado no sintetizador a partir da energia residual mínima (α^2), pois o mesmo é igual à raiz quadrada da energia residual mínima (α) multiplicada por uma constante. Devido à essa razão, a raiz quadrada da energia residual mínima é transmitida juntamente com os coeficientes LPC , ao invés do ganho da excitação. Assim, α é o parâmetro que será quantizado, sendo que as medidas de distorção serão calculadas considerando-se esse parâmetro. As constantes $\sqrt{T/N}$ para sons sonoros e $1/\sqrt{N}$ para sons não sonoros serão introduzidas na recepção quando da síntese do sinal de voz.

3.10 VOCODER LPC IMPLEMENTADO

Para implementar o vocoder LPC calculou-se os parâmetros do filtro $H(z)$, o qual modela o trato vocal, e também foi obtido o sinal de excitação do mesmo. Considerando-se as características do sinal de voz, esses parâmetros foram calculados a cada 20 ms. Como o sinal de voz original foi amostrado em 8 kHz, os parâmetros foram atualizados a cada 160 amostras.

3.10.1 Características do filtro $H(z)$

A análise LPC foi realizada utilizando-se superposição entre intervalos adjacentes em 5 ms (40 amostras).

Para a escolha da ordem do preditor foram feitos testes subjetivos utilizando preditores de ordens entre 8 e 12. Como as diferenças entre os sinais sintetizados eram mínimas optou-se pelo preditor de ordem 8.

Levando-se em conta as considerações feitas nas seções anteriores, o vocoder LPC implementado apresenta as seguintes características:

- Análise LPC: utilização do método de autocorrelação e algoritmo de Durbin para cálculo dos coeficientes LPC
- Janela: Janela de Hamming
- Intervalo de análise: 30 ms - 240 amostras
- Ordem do preditor: 8
- Coeficiente de pré-ênfase: 0.9
- Coeficiente de de-ênfase: 0.9

Quanto ao ganho do sinal de excitação, transmitiu-se a raiz quadrada da energia residual (α) e a partir dela, foi obtido no sintetizador o ganho do sinal de excitação. Para isso, utilizou-se a equação 3.43 para sons sonoros e a equação 3.45 para sons não sonoros. Nos capítulos seguintes a raiz quadrada de energia residual mínima será referenciada como ganho do modelo LPC.

3.10.2 Características do sinal de excitação

O sinal de excitação do filtro $H(z)$ foi atualizado a cada 20 ms (160 amostras), não sendo utilizada a pré-ênfase e nem a superposição de intervalos para a obtenção do mesmo. Esse sinal foi gerado da seguinte forma:

- Sons não sonoros

  Utilizou-se um gerador de números aleatórios, com função densidade de probabilidade uniforme, média zero e variância unitária.

- Sons sonoros

  Utilizou-se a seqüência de impulsos com média zero, dada pela equação 3.37.

O algoritmo utilizado para detectar o período de pitch será apresentado no próximo capítulo. Para quadros sonoros, seguintes a quadros sonoros, o período de pitch é contado a partir do último impulso unitário colocado no quadro anterior.

Capítulo 4

DETECTOR DE PITCH

4.1 INTRODUÇÃO

O detector de pitch é o componente do vocoder LPC que tem como função determinar se o sinal de voz é sonoro ou não sonoro, e encontrar a freqüência fundamental ou período de pitch dos sons sonoros. O período de pitch mede o intervalo entre sucessivos ciclos de abertura e fechamento das cordas vocais.

O pitch determina a entonação da voz sintetizada e para que esta apresente uma boa qualidade é necessário uma detecção precisa do pitch. Mantendo-se fixo o pitch, a voz sintetizada será monotônica e metálica, e dependendo do valor do mesmo, a voz será grave ou aguda.

A obtenção de um detector de pitch confiável é uma tarefa difícil, pois os sons não podem ser classificados apenas como sons sonoros ou não sonoros. Alguns sons como os oclusivos sonoros e fricativos sonoros, apresentam características dos sons sonoros e sons não sonoros, e assim não são classificados corretamente. Além disso, é difícil separar a componente do sinal de voz devido às vibrações das cordas vocais dos efeitos do trato vocal. Outro ponto que deve ser considerado é a grande quantidade de valores que o pitch pode assumir, variando de 50 Hz a 500 Hz.

Diferentes técnicas para a detecção do período de pitch tem sido apresentadas, algumas explorando as características do sinal de voz no domínio do tempo, e outras analisando o sinal no domínio da freqüência. Em muitos algoritmos a análise para a detecção do pitch é realizada no próprio sinal de voz, enquanto em outros a análise é feita no resíduo do sinal, isto é, passando-se o sinal de voz pelo filtro inverso e analisando-se a saída deste.

Os algoritmos para detecção de pitch podem ser classificados em 3 grupos [1]:

Processamento de formas de onda:

É composto pelos métodos que detectam os picos periódicos no sinal de voz, realizando uma análise do sinal no domínio do tempo.

Processamento de correlação:

Nesse grupo estão os métodos que utilizam a função de autocorrelação e variações da mesma, como a função AMDF (Average Magnitude Difference Function).

Processamento espectral:

Utilizam a análise cepstral para extração do pitch.

Para remover os efeitos do trato vocal do sinal de voz e assim evitar erros na detecção do pitch, tem-se usado técnicas para suavizar o espectro do sinal de voz. Uma das técnicas utilizadas [2] consiste em passar o sinal de voz por um ' center clipping '. Em algoritmos que utilizam o sinal de resíduo, o mesmo é filtrado por um filtro passa-baixas com freqüência de corte de aproximadamente 900 Hz para eliminar os efeitos do trato vocal (influência dos formantes).

O tamanho de cada intervalo de análise para a detecção do pitch deve envolver no mínimo um período de pitch. Para o janelamento do sinal de voz deve-se usar a janela retangular.

Os seguintes algoritmos foram estudados e implementados:

- Estimação do período de pitch utilizando a função de autocorrelação [2]
- Estimaçao do pitch baseada na filtragem LPC inversa e função AMDF [7]
- Detecção e encadeamento do período de pitch [8]

Os dois primeiros algoritmos pertencem ao grupo de processamento de correlação enquanto o último está no grupo de processamento de formas de onda. Os dois últimos algoritmos apresentaram melhores resultados e serão descritos a seguir.

4.2 DETECÇÃO E ENCADEAMENTO DO PERÍODO DE PITCH

Esse algoritmo, proposto por Schäfer Vincent [8], faz uma análise do sinal de voz no domínio do tempo, determinando se o mesmo é sonoro ou não sonoro, e o período de pitch para o caso do sinal ser sonoro. Para isso, analisa-se um segmento de voz de 100 ms. Inicialmente são determinados os pontos de máximos e mínimos locais, obtendo-se assim um ' esqueleto ' do sinal de voz que está sendo analisado. Os pontos pertencentes ao esqueleto serão chamados pontos significativos. Na figura 4.1 tem-se um segmento de um sinal de voz sonoro e o seu correspondente ' esqueleto '.

Uma amostra $y(i)$ do sinal de voz é considerada um ponto de máximo local se:

$$y(i) > y(i-1), y(i-2), ..., y(i-N) \quad \text{e}$$
$$y(i) > y(i+1), y(i+2), ..., y(i+N)$$

A definição para ponto de mínimo local é:

$$y(i) < y(i-1), y(i-2), ..., y(i-N) \quad \text{e}$$
$$y(i) < y(i+1), y(i+2), ..., y(i+N)$$

O parâmetro N é o espaçamento mínimo entre dois pontos significativos. Considerando-se que os pontos significativos são delimitadores do período de pitch, adotou-se N como o número de amostras correspondentes ao intervalo de 2 ms (para uma freqüência de amostragem igual a 8 kHz, tem-se $N = 16$). Dessa forma, o valor máximo de pitch que pode ser encontrado é 500 Hz.

Os pontos significativos são armazenados em uma memória de 20 posições. Quando todas as posições da memória estão ocupadas, para que um novo ponto possa ser armazenado na mesma, o ponto mais antigo é desprezado. Cada ponto significativo é caracterizado por três parâmetros:

- Posição no eixo temporal
- Valor da amplitude
- Tipo de ponto significativo (máximo ou mínimo)

O valor da posição no eixo temporal de cada ponto significativo é determinado com relação a um referencial temporal, o qual é atualizado com o tempo. Os dados na memória são constantemente atualizados em relação ao referencial.

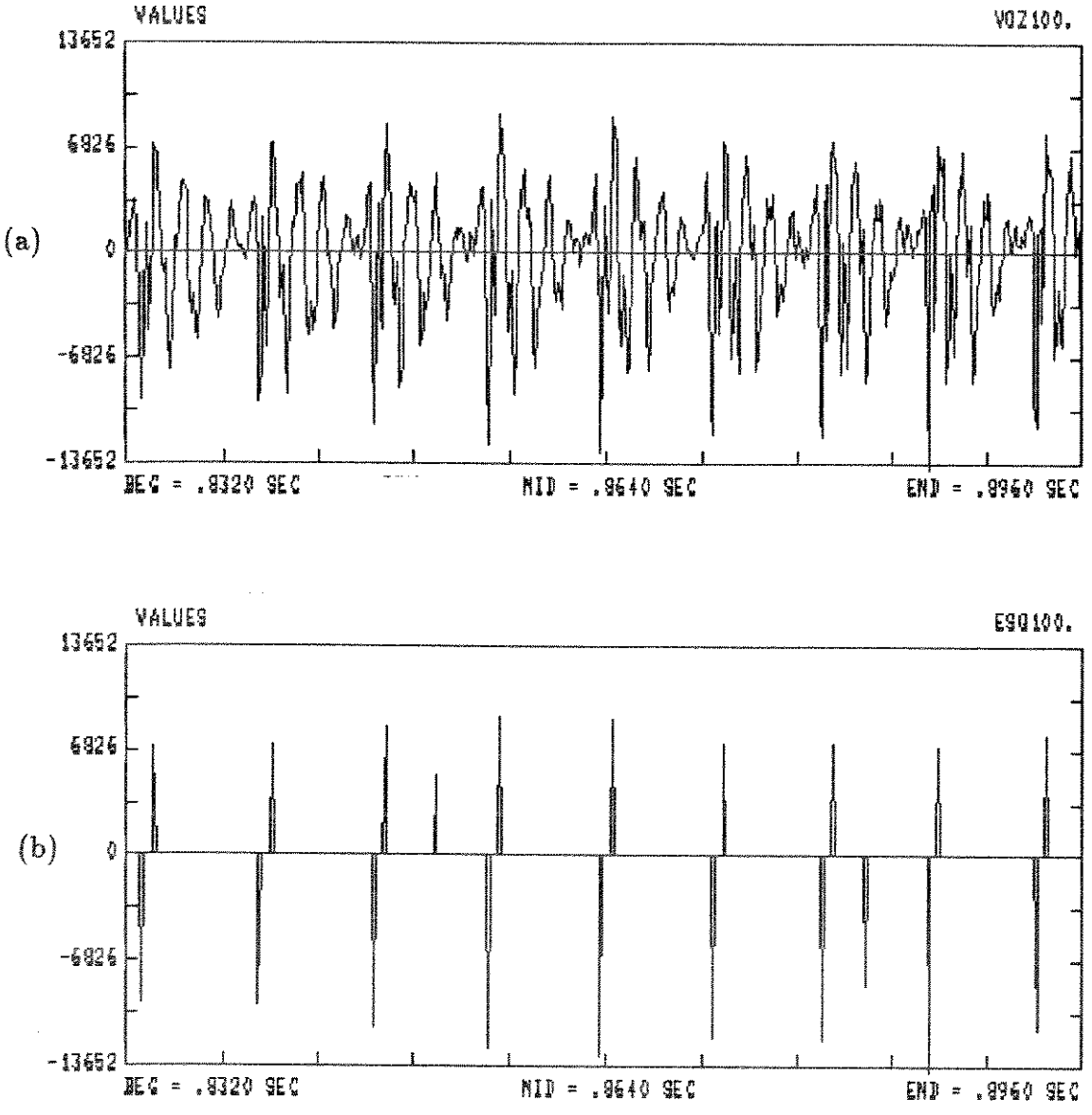


Figura 4.1: (a) sinal de voz sonoro. (b) ' esqueleto ' do sinal.

Após ter-se encontrado um ponto significativo, o próximo passo é verificar a formação de períodos gêmeos. Um período gêmeo é formado por três pontos significativos de mesmo tipo, sendo que a combinação desses pontos passou em seis testes que serão descritos a seguir. Os pontos que formam um período gêmeo são limitantes de dois possíveis períodos de pitch do sinal de voz. Para cada ponto significativo determinado, toda combinação deste ponto com os pontos armazenados na memória de 20 posições é testada para verificar se constitui um período gêmeo.

Os testes para verificar a formação de um período gêmeo são os seguintes:

Teste 1

A duração de um período de pitch, isto é, a distância entre dois pontos significativos consecutivos deve ser menor que 20 ms. Assim o menor valor de pitch que pode ser detectado para uma freqüência de amotragem de 8 kHz é 50 Hz.

Teste 2

A diferença entre a duração dos dois períodos dentro do período gêmeo deve ser inferior a 10% . As rápidas variações no pitch que podem ser voluntariamente efetuadas estão dentro desse limite. Assim, a razão entre os dois períodos deve ser menor que 1.1 e maior que 0.9.

Teste 3

Esse teste verifica se a amplitude de cada ponto significativo é maior que 0.25% do máximo valor fornecido pelo conversor A/D. Para um conversor de 16 bits (-32768 a 32767), cada ponto significativo deve apresentar amplitude superior a 80 unidades. Esse teste tem por objetivo excluir da análise os intervalos de silêncio.

Teste 4

Os três pontos significativos que formam um período gêmeo devem apresentar amplitudes maiores que as amplitudes dos pontos significativos localizados entre eles. Assim, linhas retas traçadas unindo os pontos significativos de um período gêmeo devem envolver qualquer ponto localizado entre eles.

Teste 5

A variação da envoltória dos pontos significativos que limitam um período gêmeo deve ser inferior a 50%. Para que isso ocorra, a razão entre o ponto significativo central do período gêmeo e a média aritmética dos outros dois pontos componentes do período gêmeo deve ser maior que 0.5 e menor que 2. O objetivo deste teste é evitar que o primeiro formante seja confundido com um pulso glotal, quando a freqüência do primeiro for o dobro da freqüência fundamental (pitch). Nesse caso eles podem ser distinguidos apenas pela amplitude.

Teste 6

Nesse teste as variações da magnitude média a curto prazo do sinal de voz nos dois períodos componentes do período gêmeo são comparadas para a verificação da similaridade das mesmas. Para o cálculo da variação da magnitude média do sinal de voz em cada um dos dois períodos componentes do período gêmeo, divide-se cada período em 8 segmentos. Para cada um desses segmentos a variação da magnitude média é calculada por:

$$AMV = \frac{1}{N_1} \sum_{i=1}^{N_1} |x(i)| - AM \tag{4.1}$$

$$AM = \frac{1}{N} \sum_{i=1}^{N} |x(i)| \tag{4.2}$$

onde: N é o comprimento é o comprimento do período
N_1 é o comprimento de cada um dos oito segmentos
$x(i)$ são amostras do sinal de entrada

A variação da magnitude média do sinal de voz em cada período do período gêmeo é igual à soma dos valores absolutos das variações da magnitude média dos oito segmentos que compõem cada período. A medida de similaridade utilizada é a soma das diferenças absolutas entre as variações da magnitude média de cada segmento de um período e o seu correspondente no outro período. O limiar de comparação usado neste teste é a média aritmética das variações de magnitude média nos dois períodos. Esse limiar não é fixo, pois depende da média da variação da magnitude nos dois períodos. Nesse teste é verificada a semelhança da envoltória dos dois períodos. A combinação dos pontos significativos será rejeitada

se a soma das diferenças absolutas for maior que a média aritmética das variações da magnitude média dos dois períodos.

Toda combinação de três pontos significativos que passar por todos esses testes é considerada um período gêmeo, o qual é mostrado na figura 4.2.

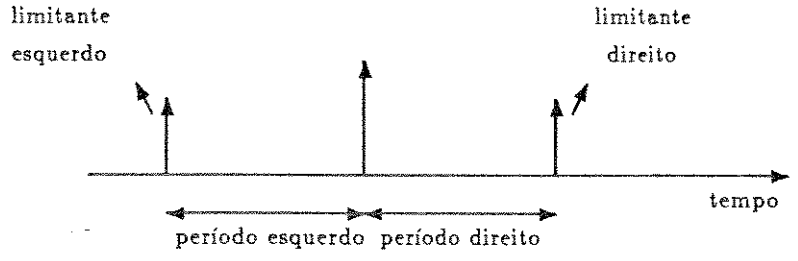


Figura 4.2: Esquema de um período gêmeo.

Após um período gêmeo ter sido determinado, o próximo passo é tentar unir os períodos gêmeos para a formação de cadeias de períodos, as quais passam por alguns testes para que os componentes das mesmas possam ser exteriorizados como períodos de pitch. Após isso, o período gêmeo é armazenado em uma memória de 100 posições. Nessa memória são armazenados vários dados sobre o período gêmeo como: se ele pertence a uma cadeia de períodos gêmeos, número de períodos gêmeos ligados a ele, duração da cadeia na qual ele está contido, se o período gêmeo é o elemento final de uma cadeia e outros. Essa memória possui um referencial temporal, sendo que os dados nela armazenados estão relacionados com esse referencial. Quando um novo período gêmeo é encontrado, o referencial temporal e os dados da memória são atualizados. Nesse procedimento são feitos alguns testes para apagar da memória os períodos gêmeos considerados antigos e sem condição de serem ligados em alguma cadeia.

Um período gêmeo será apagado se: a distância do limitante direito do período gêmeo ao referencial temporal da memória for maior que 5 vezes o período direito do último período gêmeo armazenado ou se essa distância for maior que 3 vezes o período direito do último período gêmeo e a cadeia, da qual o período gêmeo faz parte, estiver morta. Uma cadeia é considerada viva, se os períodos gêmeos componentes dessa cadeia são exteriorizados como períodos de pitch.

Para verificar se o novo período gêmeo pode ser ligado em alguma cadeia , ele é comparado com o último período gêmeo de cada cadeia. Caso exista alguma cadeia viva, esta será a primeira a ser testada. São realizados os seguintes testes:

- A razão entre o período direito do último período gêmeo da cadeia e o período esquerdo do novo período gêmeo deve ser maior que 0.86 e menor que 1.16.
- A distância entre o limitante direito do último período gêmeo da cadeia e o limitante esquerdo do novo período gêmeo deve ser menor que 1.1 vezes o período esquerdo deste.

Caso o período gêmeo não seja ligado na cadeia viva, as outras cadeias são testadas arbitrariamente, e ele será colocado na primeira cadeia em que ele passar nos testes. Nesta fase, os testes são mais rígidos, pois além da semelhança entre os períodos, no mínimo um limitante do novo período deve ser idêntico a um limitante do último período da cadeia.

Para que os períodos gêmeos componentes de uma cadeia sejam exteriorizados como períodos de pitch, é necessário que a cadeia apresente as seguintes características:

- Comprimento total maior que 30 ms.
- Comprimento total maior que 2.8 vezes o período direito do último período gêmeo da cadeia.
- Ser constituída por no mínimo 3 períodos gêmeos

Para evitar que erros sejam cometidos, adotando-se como período de pitch o dobro do valor correto, a cadeia cujos componentes serão exteriorizados deve satisfazer a seguinte condição: a duração do período direito do último período gêmeo da cadeia deve ser menor que 1.5 vezes a duração do período direito do último período gêmeo das outras cadeias.

Quando são exteriorizados, os valores dos períodos de pitch são colocados em uma memória com comprimento correspondente a 100 ms. À medida que os períodos gêmeos passam pelos testes, os valores desses períodos vão sendo armazenados nessa memória em posições correspondentes ao quadro de análise que eles pertencem. O comprimento de cada quadro de análise é igual ao intervalo de atualização do período de pitch. O valor do pitch detectado para cada quadro é a média aritmética dos valores armazenados para esse quadro. Quadros não sonoros são detectados quando o período de pitch apresenta valor zero.

Esse algoritmo apresenta as seguintes características:

- Faixa de valores de pitch: 50 a 500 Hz.
- Mínimo comprimento dos segmentos sonoros: 30 ms.
- Variação na duração de períodos de pitch adjacentes: 10%

A figura 4.3 mostra um segmento de voz feminina e o gráfico do período de pitch obtido com este algoritmo, para esse segmento de voz.

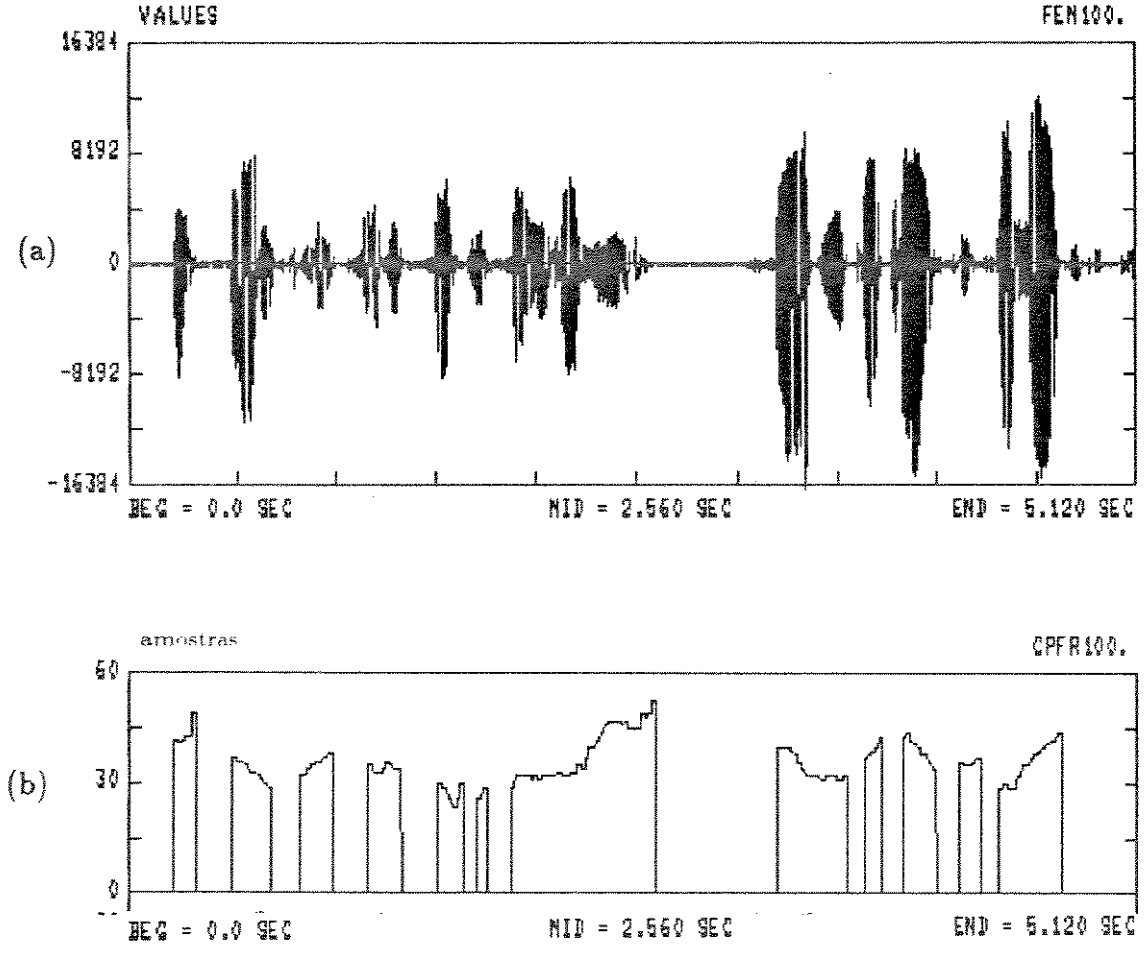


Figura 4.3: (a) Sinal de voz. (b)Período de pitch obtido com o algoritmo de detecção e encadeamento do período de pitch.

4.3 DETECTOR DE PITCH BASEADO NA FILTRAGEM LPC INVERSA E FUNÇÃO AMDF [7]

Esse algoritmo, proposto por Un e Yang [7], realiza uma análise da função AMDF e da energia do sinal de resíduo para detectar o pitch. O sinal de resíduo é obtido passando- se o sinal de voz pelo filtro inverso $A(z)$. Caso tenha-se utilizado a pré-ênfase para calcular os coeficientes LPC, deve-se passar o sinal de resíduo por um filtro de de-ênfase para atenuar as altas freqüências. O sinal de resíduo de-enfatizado é usado para o cálculo da função AMDF.

A função AMDF (Average Magnitude Difference Function) [2] baseia-se no fato de que para uma sequência verdadeiramente periódica de período P, a função:

$$d(n) = x(n) - x(n+k) \tag{4.3}$$

assume valor zero para $k = 0, \pm P, \pm 2P, ...$

Define-se a função AMDF como [2]:

$$AMDF(k) = \sum_{i=-\infty}^{\infty} |x(i) - x(i+k)| \tag{4.4}$$

O cálculo da função AMDF é mais rápido que o cálculo da função de autocorrelação, pois a subtração e a retificação são operações mais simples que a multiplicação. Para os sons sonoros, os quais são periódicos, a função AMDF apresenta valores mínimos nos pontos múltiplos do período de pitch. Esses pontos de mínimo são bastante nítidos. Para os sons não sonoros, a função AMDF não apresenta pontos de mínimos nítidos. Isso é devido ao fato dos sons não sonoros não serem periódicos. A forma de onda da função AMDF pode ser classificada em três classes [7]:

Classe 1

Nesta classe estão as funções AMDF com nítidos pontos de mínimo. O sinal de voz que produz essa função é caracterizado por ser periódico durante todo o intervalo de análise.

Classe 2

Os pontos de mínimo de uma função AMDF pertencente à essa classe, não são tão nítidos quanto os das funções da classe 1. As funções AMDF dessa classe são produzidas por segmentos de voz correspondentes ao início ou fim de sons sonoros.

Classe 3

As funções AMDF desta classe são caracterizadas por não apresentarem pontos de mínimos distintos. Elas resultam de sons não sonoros.

4.3.1 Descrição do Algoritmo

A figura 4.4 mostra o diagrama em blocos do algoritmo.

Esse algoritmo pode ser dividido em três partes: cálculo da função AMDF, procura dos pontos de mínimo e decisão sonoro/não sonoro.

Cálculo da função AMDF

A função AMDF foi calculada usando a equação:

$$AMDF(k) = \frac{1}{E}\sum_{i=1}^{N} |e(i) - e(i+k)| \quad k = 0, ..., (N_1 - N) \tag{4.5}$$

onde: $E = \sum_{i=1}^{N} |e(i)|$,

$e(i)$ são amostras do sinal de resíduo

N_1 é o comprimento do intervalo para cálculo da função AMDF. Fora desse intervalo considera-se $e(i) = 0$. Tem-se $N < N_1$.

$(N_1 - N)$ é o deslocamento máximo para cálculo da função AMDF.

N é número de amostras correspondentes ao intervalo de atualização do período de pitch

Como a função AMDF foi normalizada, os seus valores mínimos dependem muito pouco das amplitudes das amostras do sinal de resíduo.

Determinação dos pontos de mínimo da função AMDF

O primeiro passo nesse procedimento é a determinação do valor mínimo da função AMDF. O ponto no qual a função AMDF assume esse valor é considerado o primeiro candidato ao período de pitch. Outros pontos de mínimo são localizados usando-se limiares TH1 e TH2. Esses pontos de mínimo devem estar distantes no mínimo TH1 amostras uns dos outros e os valores da função AMDF nesses pontos devem estar dentro da região entre o valor mínimo e a soma desse valor com TH2.

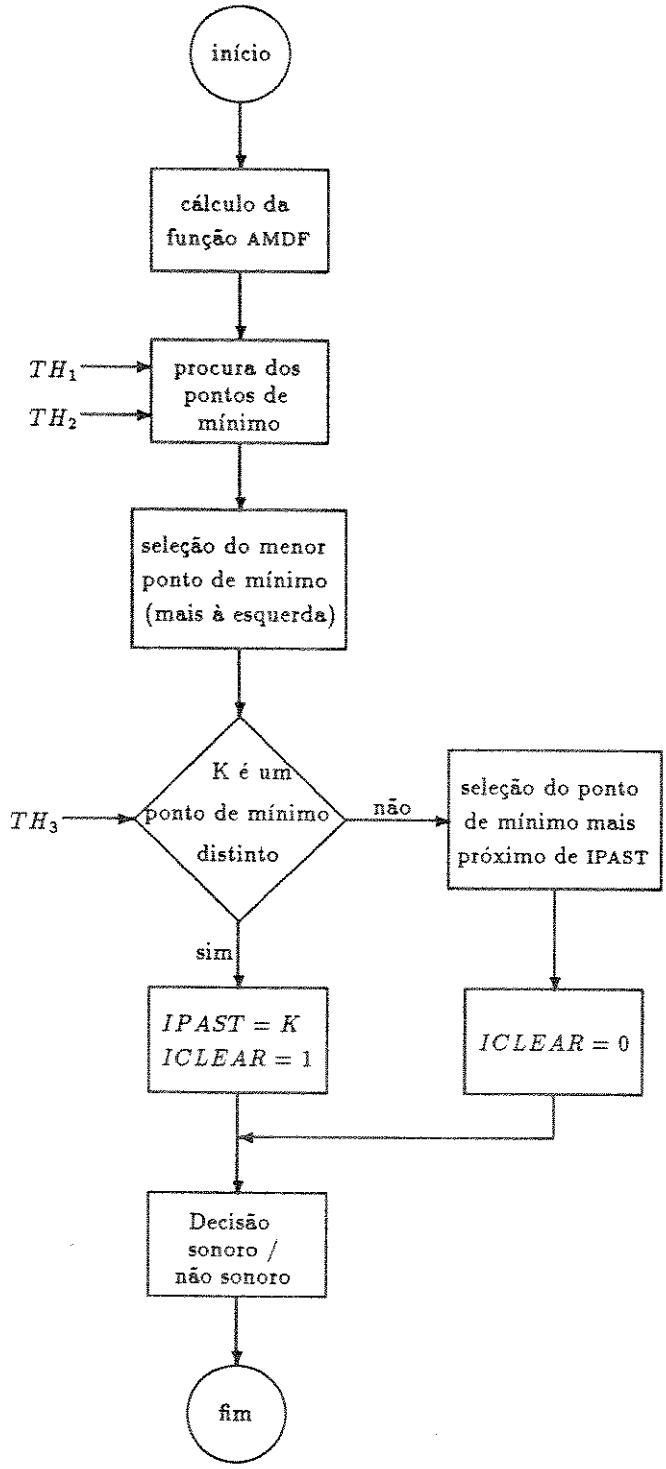


Figura 4.4: Diagrama em blocos do algoritmo

Como a função AMDF apresenta valores mínimos em pontos iguais ao período de pitch e seus múltiplos, o ponto de mínimo mais a esquerda (menor ponto de mínimo) é escolhido como provável período de pitch. A seguir é realizado um teste para ver se esse ponto é um ponto de mínimo distinto. Um ponto de mínimo é considerado distinto se a diferença entre o valor da função AMDF nesse ponto e o valor máximo da função AMDF na região compreendida por 16 pontos de cada lado do ponto de mínimo for maior que o limiar TH3.

Se o ponto de mínimo for considerado distinto a variável ICLEAR recebe o valor 1. Caso contrário, essa variável recebe o valor zero e o ponto de mínimo mais próximo do último ponto de mínimo distinto é escolhido como provável período de pitch. A variável IPAST armazena o valor do último ponto de mínimo considerado distinto. A figura 4.5 mostra os limiares TH1, TH2, TH3.

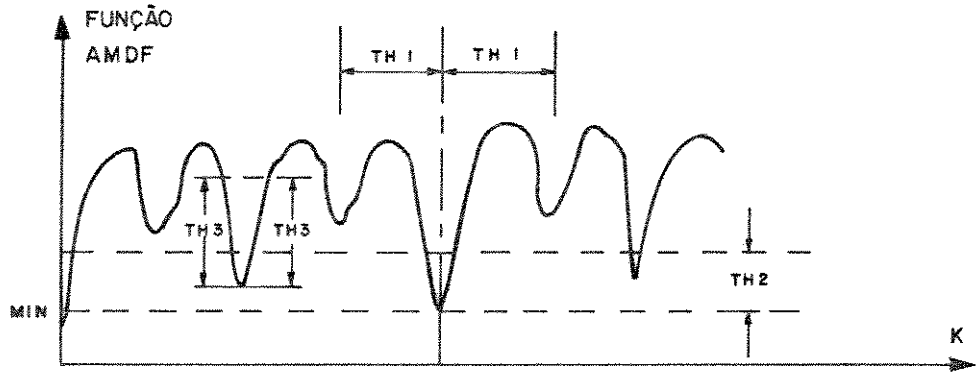


Figura 4.5: Limiares TH1, TH2, TH3.

Decisão Sonora/Não Sonora

A decisão sonora/não sonora é baseada na energia do sinal de resíduo e no parâmetro ICLEAR. A figura 4.6 mostra o fluxograma da decisão sonoro/não sonoro.

A variável ICLEAR igual a 1 indica forte periodicidade do segmento em análise que, portanto, deve ser sonoro.

O limiar E1 é usado para classificar os segmentos que contêm o final de uma região não sonora e o início de uma região sonora. Com esse limiar determina-se qual das duas regiões é maior. Para os segmentos que contém o final de uma região sonora e o início de uma região não sonora, usa-se o limiar E2 para classificá-los.

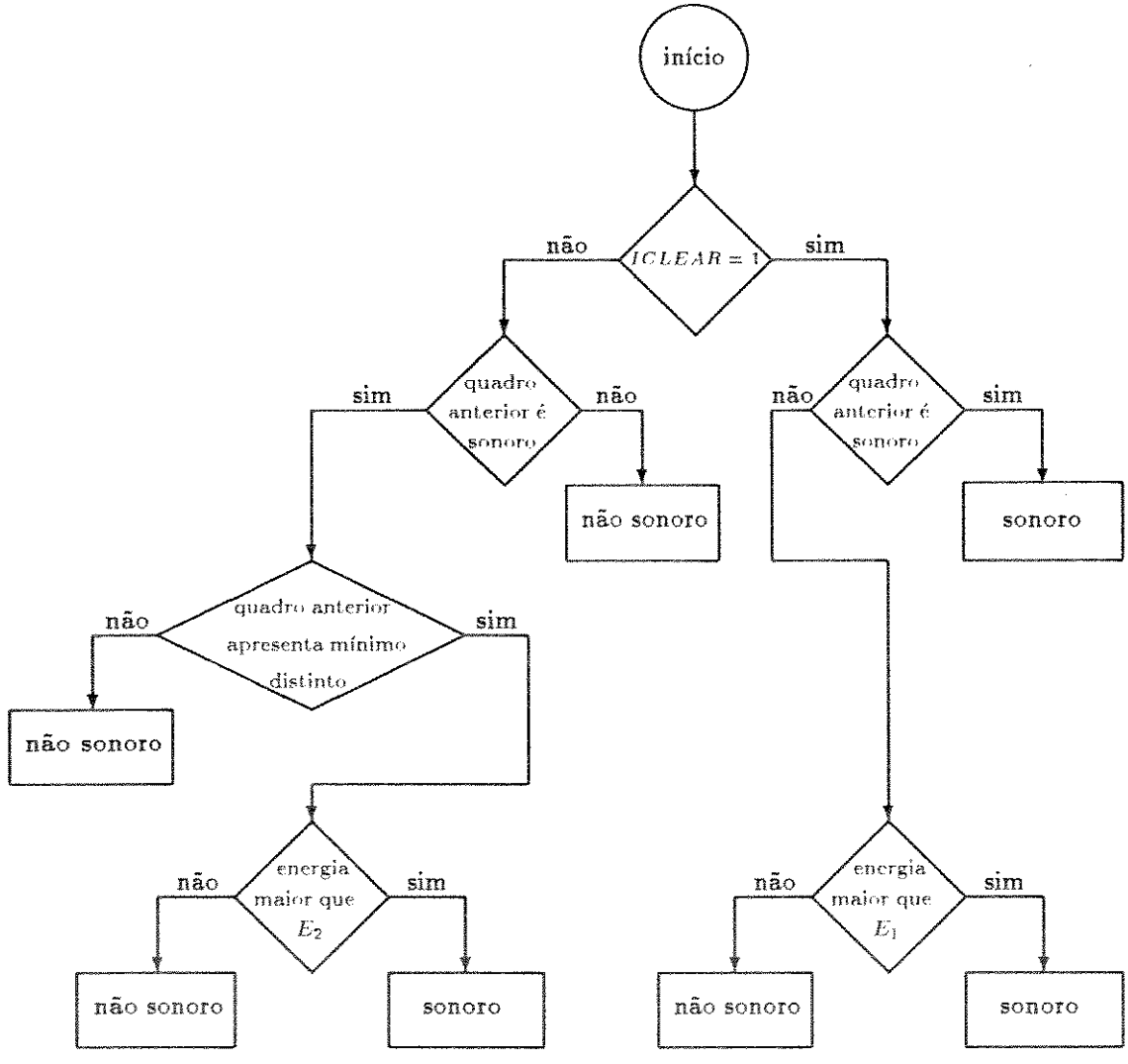


Figura 4.6: Fluxograma da decisão sonoro/não sonoro.

Quando o segmento em análise apresentar fraca periodicidade e o segmento anterior tiver sido classificado como não sonoro, este segmento também é classificado como não sonoro. Para segmentos não sonoros, a saída do detector de pitch é igual a zero.

Na figura 4.7 é apresentado um segmento de voz feminima e o gráfico do período de pitch obtido com esse algoritmo, para esse segmento de voz.

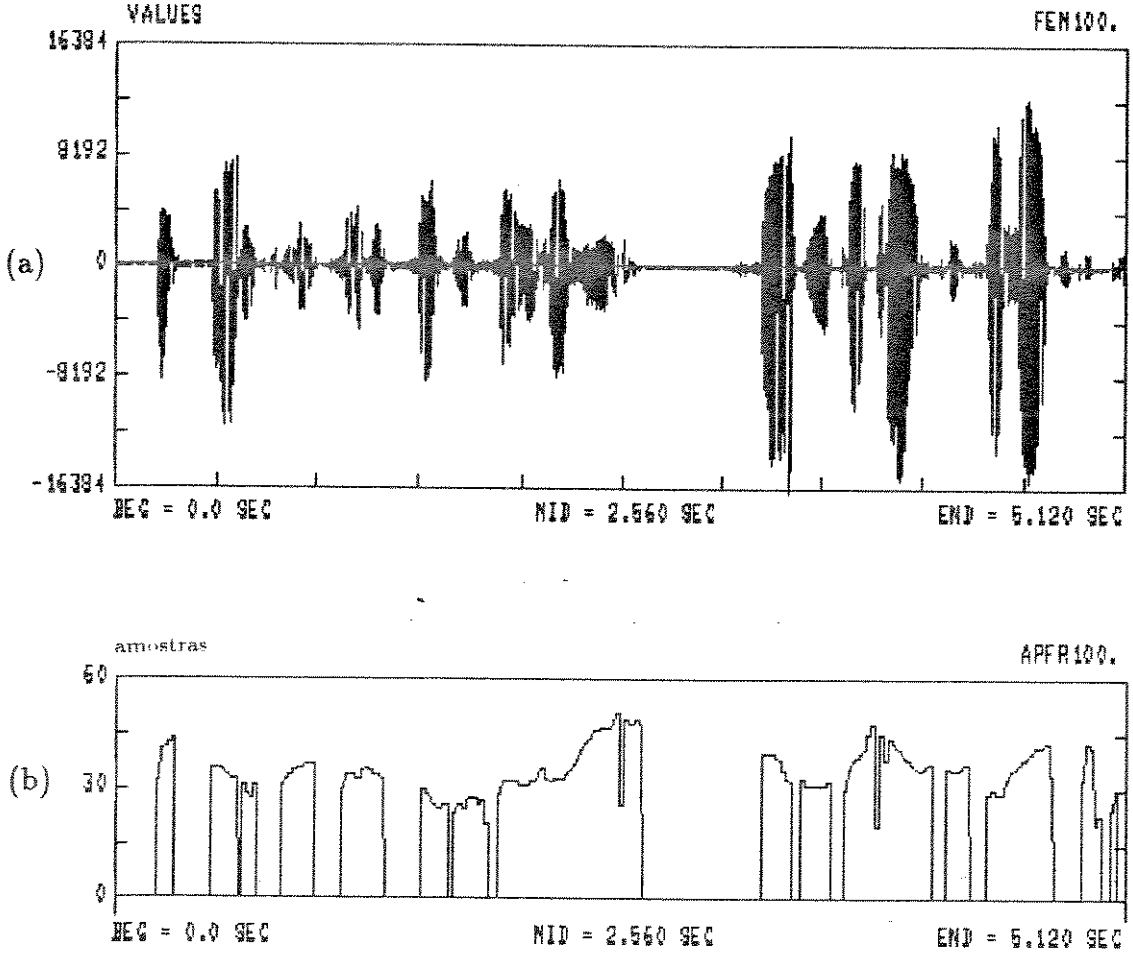


Figura 4.7: (a) Sinal de voz. (b) Período de pitch obtido com o algoritmo para detecção de pitch baseado na filtragem LPC inversa e função AMDF, para um segmento de voz feminina.

4.4 SUAVIZADORES

Para corrigir erros do detector de pitch, como erros na decisão sonoro/não sonoro ou a obtenção de valores múltiplos do período de pitch, são usados algoritmos suavizadores. Os suavizadores devem eliminar as irregularidades, mas devem preservar as descontinuidades dos períodos de pitch detectados. Dois algoritmos suavizadores foram implementados.

Suavizador 1 [5]

Denominado ' median smoothing ', esse suavizador realiza um janelamento do sinal de entrada, utilizando uma janela retangular de comprimento N (Nímpar). Após o janelamento, as amostras são ordenadas em amplitude. A saída do suavizador é a amostra que, após a ordenação, ocupar a posição $(N+1)/2$. Esse algoritmo preserva as descontinuidades em intervalos maiores ou iguais a $(N+1)/2$ e apresenta um atraso igual a $(N+1)/2$.

Na implementação desse algoritmo utilizou-se uma janela com comprimento igual a 5 amostras do sinal de entrada. Na figura 4.8 são apresentados gráficos do período de pitch original e do período de pitch suavizado, utilizando-se o suavizador 1.

Suavizador 2 [7]

Nesse suavizador, o qual foi baseado no algoritmo proposto por Un e Yang [7], o período de pitch do quadro atual é corrigido baseado nos períodos de pitch do quadro anterior e de dois quadros posteriores. Devido a isso, esse suavizador requer um atraso de dois quadros. O fluxograma desse suavizador é apresentado na figura 4.9.

Esse suavizador corrige o valor do pitch detectado da seguinte forma:

- Se o quadro atual é sonoro e o quadro anterior e um dos quadros posteriores são não sonoros, o quadro atual é mudado para não sonoro.
- Se o quadro atual é não sonoro e os quadros anterior e posterior são sonoros, o quadro atual é mudado para sonoro e adota-se como valor do pitch o pitch do quadro anterior.

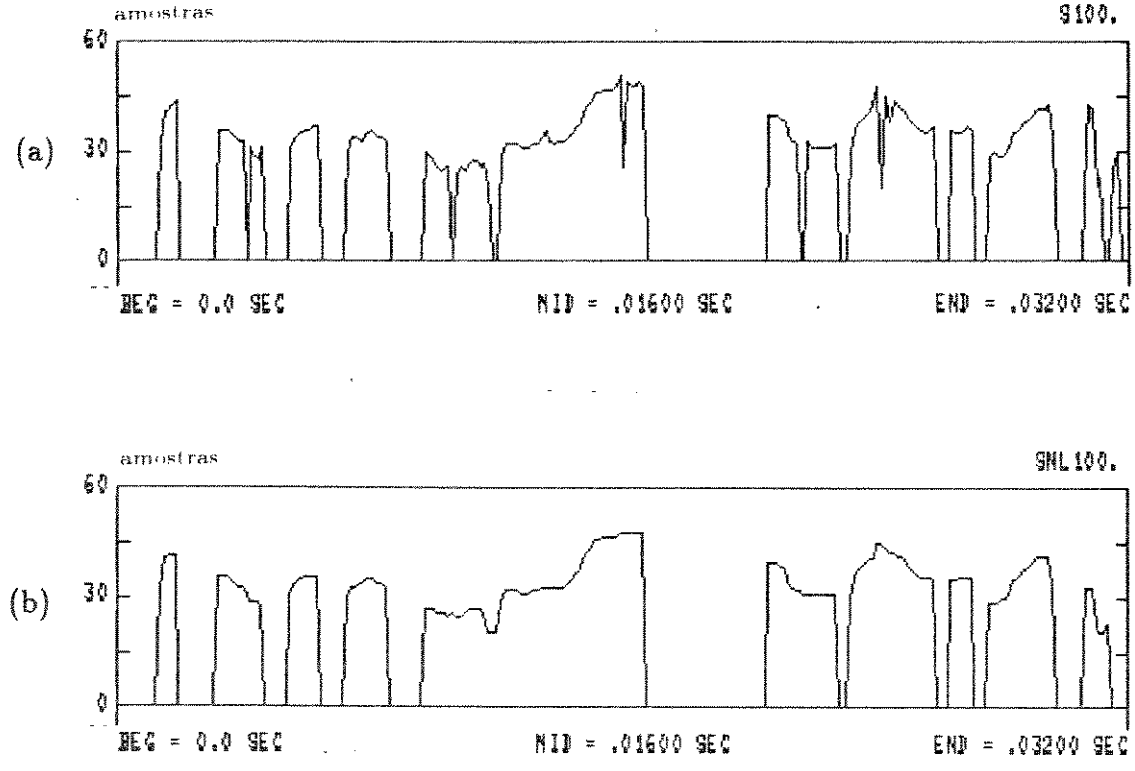


Figura 4.8: (a) Período de pitch original. (b) Período de pitch suavizado, usando-se o suavizador 1.

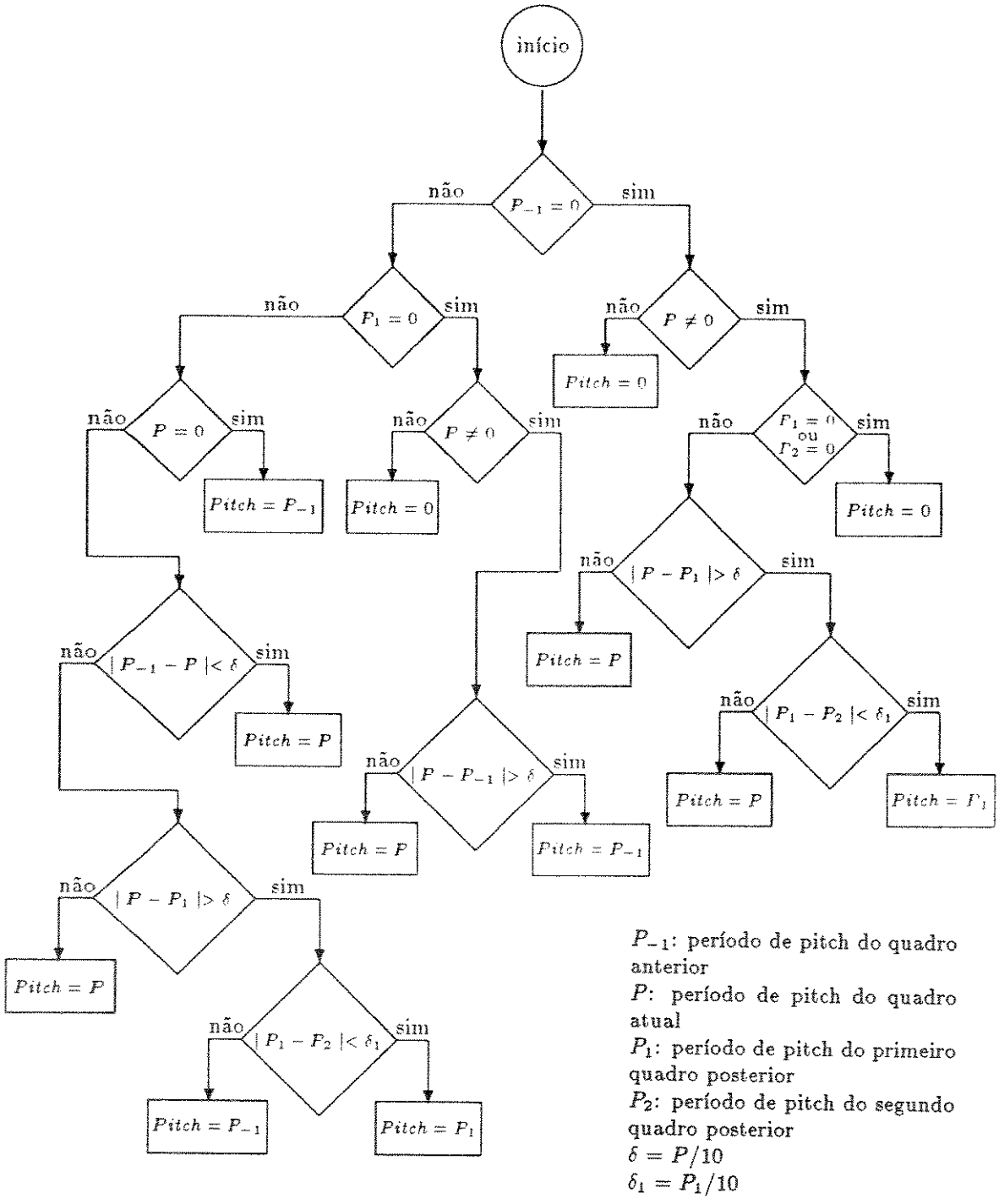


Figura 4.9: Fluxograma do suavizador 2.

- Para quadros sonoros seguintes a um quadro não sonoro, o valor de pitch do quadro atual é comparado com os valores de pitch dos quadros posteriores. Se a diferença entre eles for maior que um limiar, o valor do pitch do quadro atual é substituído pelo valor do pitch do quadro seguinte.
- Para quadros sonoros seguintes a um quadro sonoro e anteriores a um quadro não sonoro, é encontrada a diferença entre os valores do pitch do quadro atual e do quadro anterior. Caso essa diferença seja maior que um limiar, o valor do período de pitch do quadro atual é substituído pelo período de pitch do quadro anterior.
- Para quadros sonoros seguintes a um quadro sonoro e anteriores a um quadro sonoro, o valor do pitch do quadro atual é comparado com o valor do pitch do quadro anterior. Se a diferença entre eles for maior que um limiar, compara-se o valor do pitch do quadro atual com valor do pitch do quadro posterior. Se a diferença exceder ao limiar, compara-se os períodos de pitch dos quadros posteriores. Caso os mesmos sejam semelhantes, assume-se como período de pitch do quadro atual o período de pitch do quadro posterior. Caso contrário, toma-se o período de pitch do quadro anterior como o valor correto para o quadro atual.

  Na implementação desse algoritmo utilizou-se os seguintes limiares:
  δ = 10% do valor do período de pitch do quadro atual
  δ_1 = 10% do valor do período de pitch do quadro posterior

A figura 4.10 mostra o período de pitch original e o período de pitch suavizado, usando-se o suavizador 2.

4.5 DISCUSSÃO DOS ALGORITMOS

Testes com esses algoritmos mostraram que pequenos erros na detecção do pitch podem causar um ruído ' metálico ' similar a um ' eco ' na voz sintetizada. Esse problema foi encontrado principalmente em vozes femininas. Como essas vozes apresentam períodos de pitch menores, os erros tornam-se mais significativos. Outro problema observado foi a ocorrência de ' soluços ', os quais são causados quando detecta-se como período de pitch a metade do valor correto.

Em todos os algoritmos implementados o período de pitch é atualizado a cada 20ms (160 amostras para freqüência de amostragem de 8 kHz).

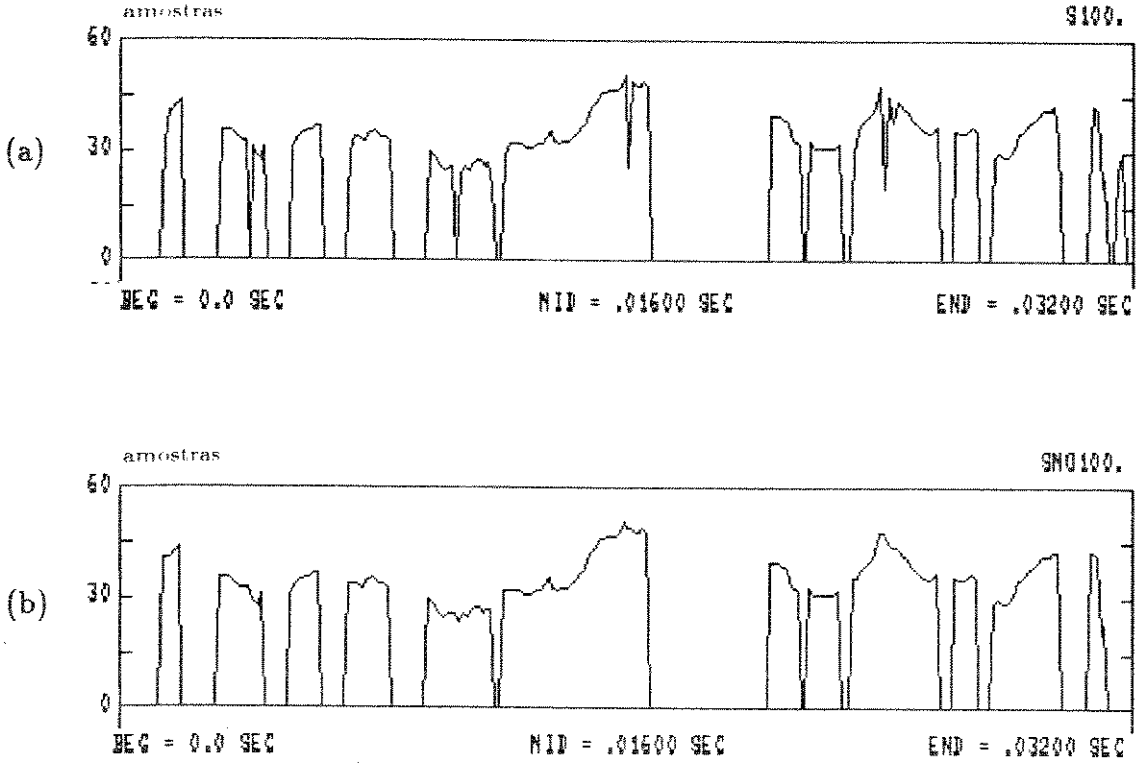


Figura 4.10: (a) Período de pitch original. (b) Período de pitch suavizado, usando-se o suavizador 2.

Na implementação do algoritmo baseado na filtragem inversa, os seguintes limiares foram usados:

- N: 160 amostras
- N_1: 320 amostras
- TH1: 16
- TH2: 0.25
- TH3: 0.75
- E1: 500000
- E2: 500000

Esse valores foram obtidos experimentalmente, analisando-se vozes masculinas e femininas.

Para eliminar os efeitos do trato vocal, o sinal de resíduo foi filtrado por um filtro IIR passa-baixas de primeira ordem [6]. A transformada Z do filtro utilizado é dada por:

$$F(z) = \frac{1}{1 - 0.99z^{-1}} \tag{4.6}$$

O principal erro apresentado pelo algoritmo baseado na filtragem inversa foi a obtenção de valores múltiplos de pitch, tornando-se necessária a utilização de um suavizador. Assim, conseguiu-se bons resultados para vozes masculinas e femininas.

O algoritmo de detecção e encadeamento do período de pitch apresentou alguns poucos erros na detecção sonoro/não sonoro, apresentando bons resultados tanto para vozes masculinas como para vozes femininas e assim, dispensando o uso de suavizadores. Devido ao fato de que os limiares utilizados não precisam ser ajustados, os resultados obtidos com esse algoritmo independem do locutor. Além disso, os custos computacionais são menores , pois são realizadas apenas comparações. O único incoveniente desse algoritmo é o atraso de 100 ms. Assim, decidiu-se usar esse algoritmo para a detecção de pitch nos vocoders implementados.

Capítulo 5

QUANTIZAÇÃO ESCALAR E INTERPOLAÇÃO

5.1 INTRODUÇÃO

No processo de síntese de voz, os parâmetros do vocoder, os quais são os coeficientes do preditor, o ganho da excitação e o período de pitch, são calculados e transmitidos ao receptor, onde a voz é reconstruída usando-se o modelo de produção de voz descrito nos capítulos anteriores. Para seguir a natureza variante no tempo do sinal de voz, esses parâmetros são normalmente atualizados a uma taxa de 50 a 100 Hz.

Para a transmissão desses parâmetros em um canal de comunicação digital ou para armazená-los em uma memória digital, é necessário que eles sejam quantizados em um número finito de valores para que possam ser representados por um número finito de bits. Quantização é o processo de aproximação de sinais de amplitudes contínuas por sinais de amplitudes discretas. A quantização independente de cada parâmetro é denominada quantização escalar.

A qualidade da voz sintetizada depende do tipo de quantização e dos parâmetros que são transmitidos. Para a representação do filtro de síntese $H(z)$, vários conjuntos de parâmetros podem ser transmitidos.

5.2 QUANTIZAÇÃO ESCALAR

5.2.1 Parâmetros para a representação do filtro $H(z)$ [9]

O modelo com apenas pólos usado em um sistema de predição linear tem como função de transferência:

$$H(z) = \frac{G}{A(z)} \tag{5.1}$$

onde o filtro inverso $A(z)$ é dado por:

$$A(z) = 1 - \sum_{n=1}^{P} a_n z^{-n} \tag{5.2}$$

Os seguintes parâmetros podem ser transmitidos para representar $H(z)$:

- Coeficientes do preditor a_k, $1 \leq n \leq P$
- Resposta impulsiva do modelo com apenas pólos

  A resposta impulsiva $h(n)$ do filtro $H(z)$ pode ser obtida a partir dos coeficientes LPC por:

$$h(n) = \sum_{k=1}^{P} a_k h(n-k) + G\delta(n) \ , \quad 0 \leq n \leq P \tag{5.3}$$

  As primeiras $P+1$ amostras da resposta impulsiva especificam o filtro $H(z)$.

  Os coeficientes a_k podem ser obtidos a partir das primeiras $P+1$ amostras da resposta impulsiva, através da equação 5.3, tendo-se os a_k como incógnitas e conhecendo-se os $h(n)$.

- Autocorrelação da resposta impulsiva de $H(z)$

  A função de autocorrelação da resposta impulsiva [9] é definida por:

$$R_h(i) = \sum_{n=0}^{\infty} h(n)h(n+i) \ , \quad 0 \leq i \leq P \tag{5.4}$$

  Os coeficientes $R_h(i)$ são iguais aos coeficientes $R(i)$ para $0 \leq i \leq P$ [9]. Assim, os coeficientes a_k podem ser obtidos resolvendo-se a equação normal (eq. 3.24).

- Autocorrelação dos coeficientes do preditor

  A função de autocorrelação dos coeficientes do preditor [9] é dada por:

$$R_a(i) = \sum_{k=0}^{P-i} a_k a_{k+i} \ , \quad 0 \leq i \leq P \tag{5.5}$$

Para a obtenção dos coeficientes a_k, utiliza-se a FFT para obter-se o espectro de potência de $A(z)$ a partir da autocorrelação dos coeficientes do preditor. O espectro de potência de $H(z)$ é obtido invertendo-se as amostras do espectro de potência de $A(z)$ e multiplicando-as por G^2. Usando-se a transformada de Fourier inversa, obtêm-se $R_h(i)$ a partir do espectro de potência de $H(z)$. Usa-se a função de autocorrelação $R_h(i)$ para calcular os coeficientes a_k através da equação normal.

- Coeficientes Cepstrais

  Os coeficientes cepstrais [2] podem ser obtidos por um método iterativo diretamente dos coeficientes do preditor, usando-se as equações:

$$c_1 = a_1 \tag{5.6}$$

$$c_n = a_n + \sum_{m=1}^{n-1} \frac{m}{n} c_m a_{n-m} \ , \quad 2 \leq n \leq P \tag{5.7}$$

  Os coeficientes LPC são obtidos a partir dos coeficientes cepstrais, resolvendo-se iterativamente as equações:

$$a_1 = c_1 \tag{5.8}$$

$$a_n = c_n - \sum_{m=1}^{n-1} \frac{m}{n} c_m a_{n-m} \ , \quad 2 \leq n \leq P \tag{5.9}$$

- Pólos de $H(z)$ ou zeros de $A(z)$, z_k, $1 \leq k \leq P$

- Coeficientes Parcor

  Os coeficientes parcor k_i são obtidos como um subproduto do cálculo dos coeficientes do preditor pelo método de autocorrelação. Esses coeficientes podem ser obtidos diretamente dos coeficientes do preditor usando-se a seguinte forma recursiva:

$$k_i = a_i^{(i)} \tag{5.10}$$

$$a_j^{(i-1)} = \frac{a_j^{(i)} + a_i^{(i)} a_{i-j}^{(i)}}{1 - k_i^2} \ , \quad 1 \leq j \leq i-1 \tag{5.11}$$

  onde i varia de P a 1 e inicialmente tem-se:

$$a_j^{(P)} = a_j \ , \quad 1 \leq j \leq P \tag{5.12}$$

  Os coeficientes parcor apresentam magnitude menor ou igual a 1.

Os coeficientes LPC são obtidos a partir dos coeficientes parcor usando-se as seguintes equações:

$$a_i^{(i)} = k_i \tag{5.13}$$

$$a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)} \ , \ 1 \leq j \leq i-1 \tag{5.14}$$

A solução dessas equações é dada por:

$$a_j = a_j^{(P)} \quad \text{para} \quad 1 \leq j \leq P \tag{5.15}$$

5.2.2 Propriedades da Quantização [10]

Duas propriedades desejáveis que o conjunto de parâmetros a ser quantizado deve apresentar são:

- Estabilidade do filtro $H(z)$ para os parâmetros quantizados

  Depois da quantização dos parâmetros, o novo filtro $H(z)$ obtido com os coeficientes quantizados deve apresentar pólos dentro do círculo unitário.

- Os parâmetros devem apresentar uma ordenação natural

  Como exemplo dessa propriedade tem-se: os coeficientes do preditor são ordenados como a_1, a_2, ..., a_P. Se os coeficientes a_1 e a_2 são trocados, tem-se um preditor diferente do preditor $H(z)$ original. Os pólos de $H(z)$ não apresentam essa propriedade, pois a troca de dois pólos quaisquer não altera $H(z)$. Para parâmetros que apresentam essa propriedade estudos estatísticos da distribuição de cada parâmetro podem ser usados para a quantização dos mesmos.

Dos parâmetros apresentados na seção anterior, apenas os coeficientes parcor apresentam as duas propriedades. A quantização dos coeficientes do preditor, resposta impulsiva e coeficientes de autocorrelação pode causar instabilidade do filtro $H(z)$ [9]. A quantização desses parâmetros necessita de um elevado número de níveis, resultando em uma alta taxa de transmissão. A quantização dos coeficientes cepstrais também pode causar instabilidade do filtro $H(z)$ [9].

Dessa forma, os melhores parâmetros para serem quantizados são os coeficientes parcor, pois além de garantirem a estabilidade do filtro $H(z)$, apresentam uma ordem natural.

5.2.3 Quantização Ótima dos Coeficientes Parcor [10]

Para a escolha de um esquema ótimo para a quantização dos coeficientes parcor, é necessário estudar a sensibilidade do espectro do filtro $H(z)$ para variações dos coeficientes parcor.

A sensibilidade espectral do coeficiente parcor k_i é definida por:

$$\frac{\partial S}{\partial k_i} = \lim_{\Delta k_i \to 0} \frac{|\Delta S|}{|\Delta k_i|} \tag{5.16}$$

onde ΔS é o desvio no espectro de $H(z)$ devido a uma variação Δk_i no coeficiente k_i.

A figura 5.1 mostra as curvas de sensibilidade espectral para os coeficientes de reflexão (negativo dos coeficientes parcor) de um filtro $H(z)$ com 12 coeficientes, calculados em segmentos de 20 ms de voz [10]. Cada curva foi obtida variando-se o valor de cada coeficiente, enquanto os outros 11 coeficientes foram mantidos constantes.

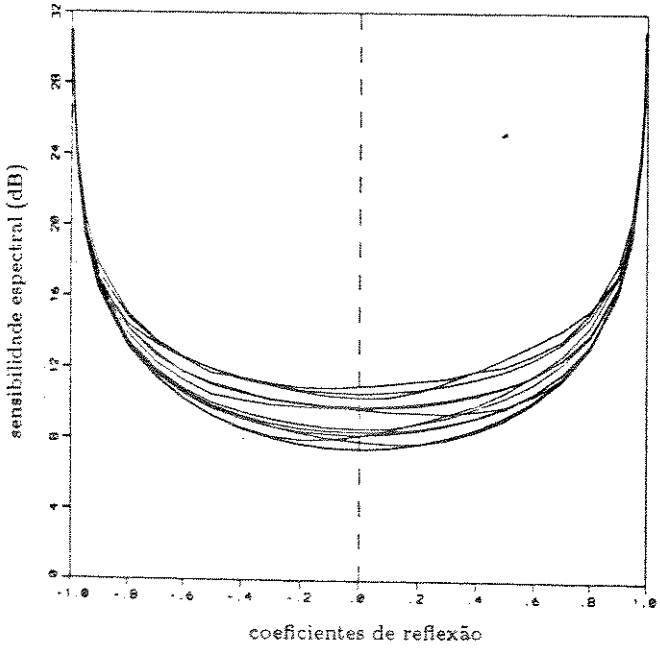


Figura 5.1: Curvas de sensibilidade dos coeficientes de reflexão [10].

As curvas de sensibilidade apresentam as seguintes características:

- Apresentam a mesma forma, independente do índice do coeficiente.

- Apresentam formato em U, são simétricas em relação a $k_i = 0$ e assumem grandes valores para valores de k_i próximos a ± 1. Além disso, para valores de k_i próximos a zero, a sensibilidade é pequena.

As curvas de sensibilidade dos coeficientes parcor mostram que uma quantização linear dos mesmos não deve apresentar bons resultados, especialmente quando eles assumem valores próximos a ± 1. Assim, deve-se usar um quantizador não linear que apresente um maior número de níveis de quantização próximo a ± 1.

Uma quantização não linear dos coeficientes parcor é equivalente a uma quantização linear de um parâmetro obtido através de uma transformação não linear dos coeficientes parcor. Essa transformação deve resultar em parâmetros que apresentem sensitividade espectral plana.

Uma transformação com essas características é dada por [10]:

$$g_i = \ln \frac{1 - k_i}{1 + k_i} \tag{5.17}$$

Os parâmetros g_i são denominados coeficientes razão log-área e são iguais ao logaritmo da razão das áreas de seções adjacentes do modelo de tubos sem perdas, o qual modela o trato vocal.

A figura 5.2 mostra as curvas de sensibilidade para os coeficientes razão log-área [10]. Nessa figura pode-se observar a sensibilidade plana desses coeficientes.

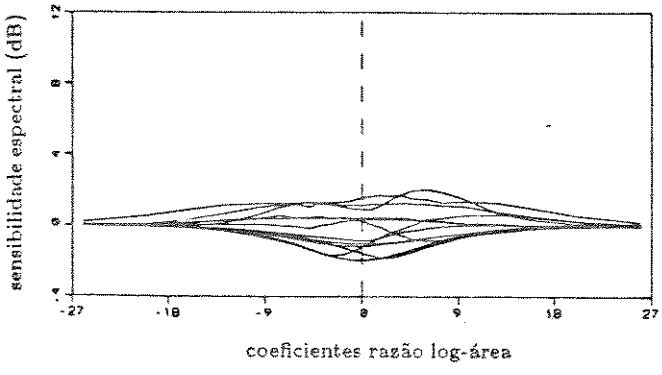


Figura 5.2: Curvas de sensibilidade para os coeficientes razão log-área [10].

Outra transformação não linear também usada é o método do seno inverso, o qual transforma o coeficiente parcor k_i no ângulo θ_i, dada por [10]:

$$\theta_i = \arcsin(k_i) \tag{5.18}$$

5.2.4 Quantização Linear ' Piecewise ' [11]

Esse método utiliza a probabilidade de ocorrência dos parâmetros a serem quantizados para alocar mais níveis de quantização para as regiões de alta probabilidade. Nesse método, o intervalo de variação de cada parâmetro é dividido em um certo número de regiões, e para cada região é definido um número de níveis de quantização. Dentro de cada região é realizada uma quantização uniforme. Os limitantes de cada região e o número de níveis de quantização em cada uma delas dependem da freqüência de ocorrência dos valores de cada parâmetro e da precisão da quantização desejada.

A figura 5.3 mostra a quantização ' piecewise ' para um coeficiente parcor.

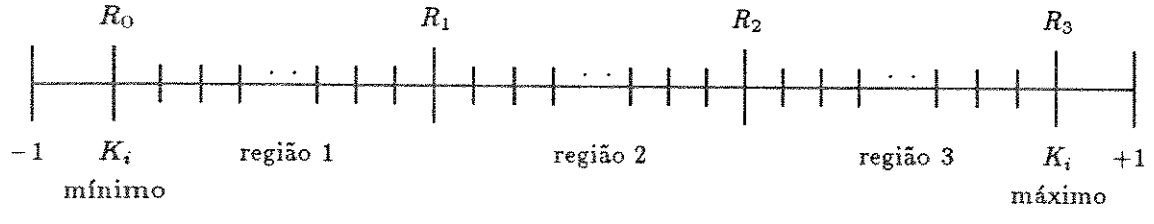


Figura 5.3: Quantização ' piecewise ' para um coeficiente parcor.

Esse tipo de quantização evita que em regiões onde não seja necessária uma boa quantização, sejam usados muitos níveis. Isso faz com que a taxa de bits necessária para a transmissão dos parâmetros seja reduzida.

5.2.5 Quantização dos Coeficientes Parcor e Razão log-área

Para a quantização dos coeficientes parcor e razão log-área, foi obtida a distribuição de probabilidade dos coeficientes parcor, usando vozes de cinco homens e cinco mulheres. Cada um dos locutores contribuiu com 1 minuto de voz, resultando em um segmento de 10 minutos de voz. Foi realizada uma análise LPC de ordem 8, utilizando-se intervalos de 30 ms e coeficiente de pré-ênfase igual a 0.9. Os coeficientes parcor foram atualizados a cada 20 ms. As figuras 5.4 a 5.11 mostram os histogramas dos oito coeficientes parcor.

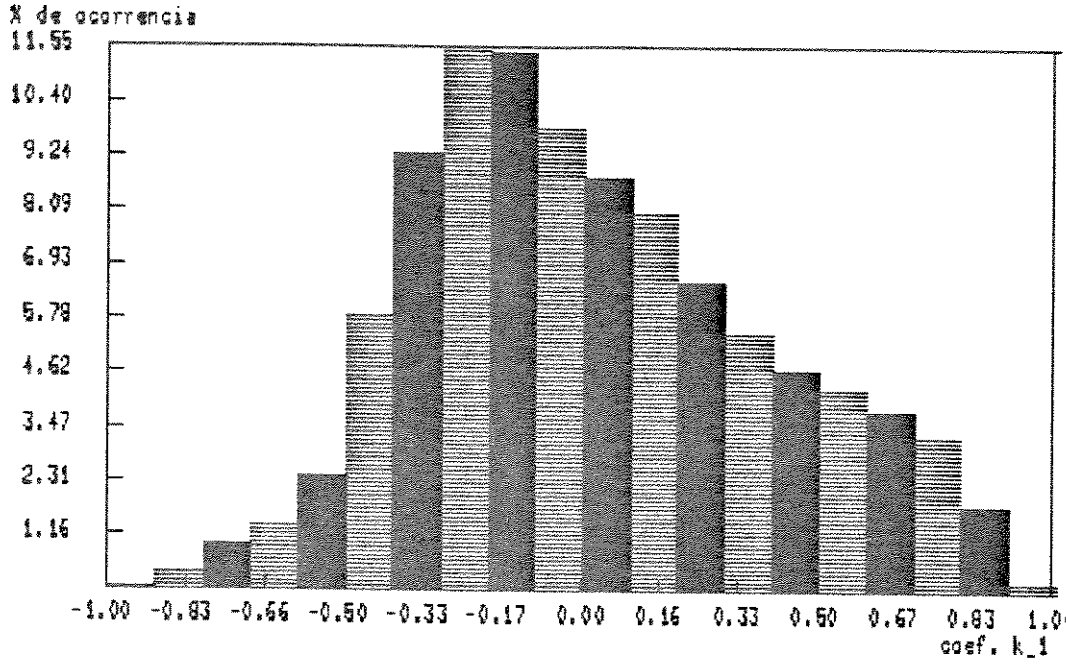


Figura 5.4: Histograma do coeficiente parcor k_1.

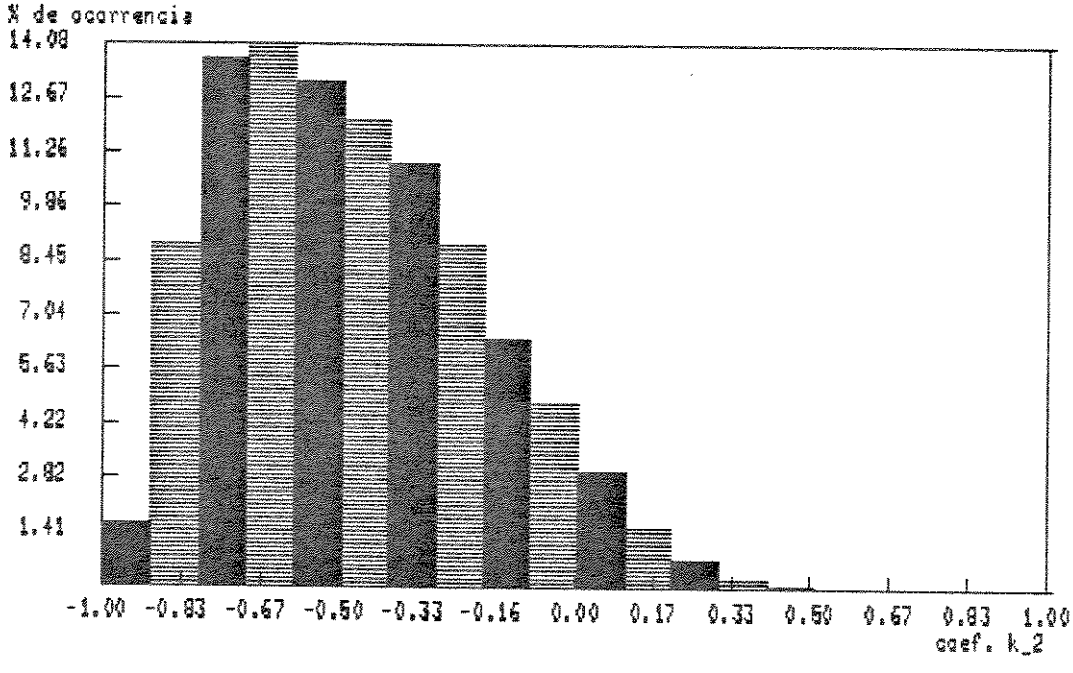


Figura 5.5: Histograma do coeficiente parcor k_2.

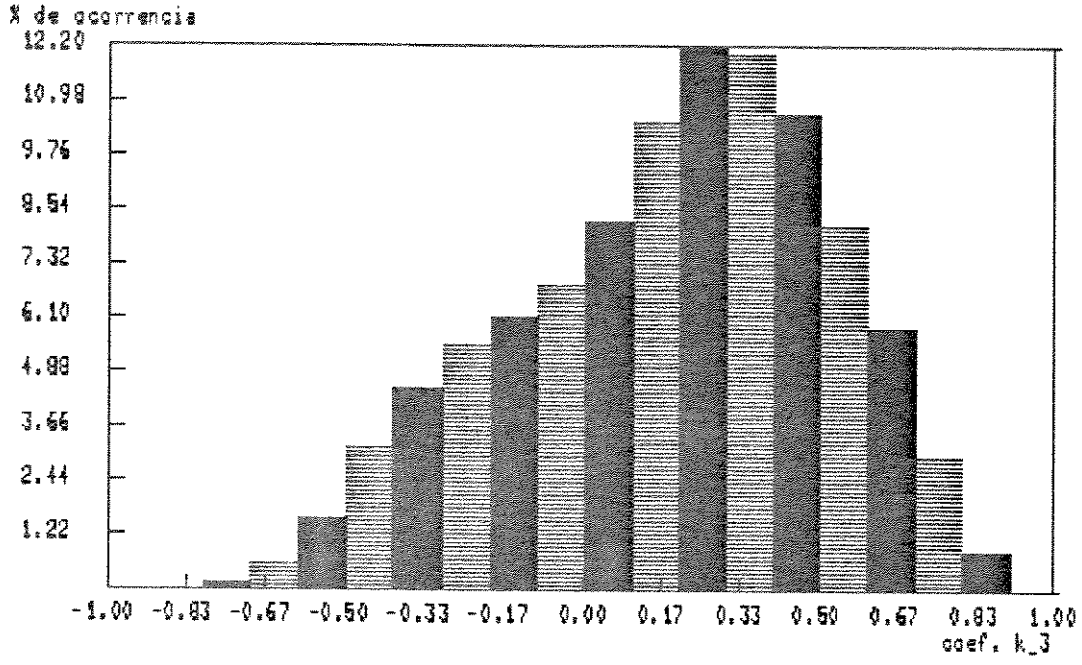


Figura 5.6: Histograma do coeficiente parcor k_3.

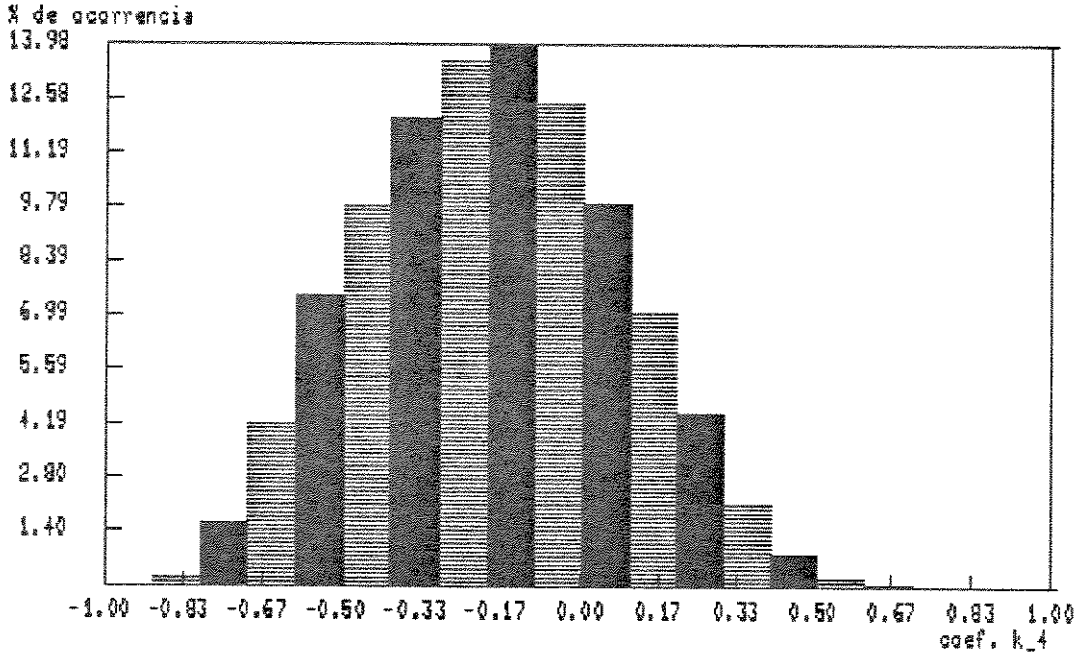


Figura 5.7: Histograma do coeficiente parcor k_4.

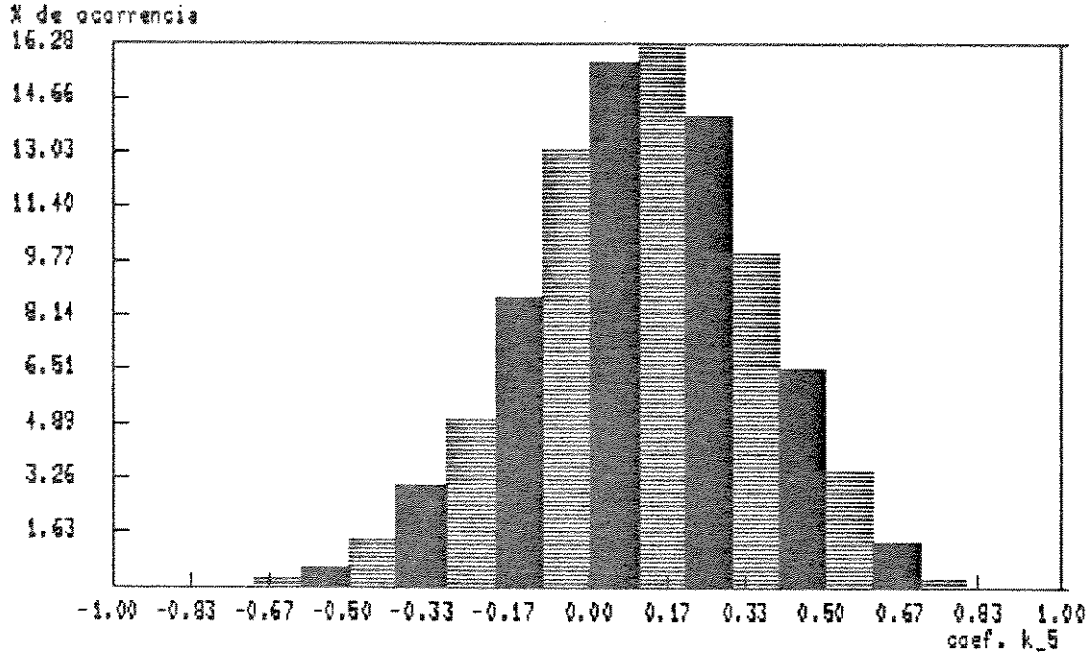


Figura 5.8: Histograma do coeficiente parcor k_5.

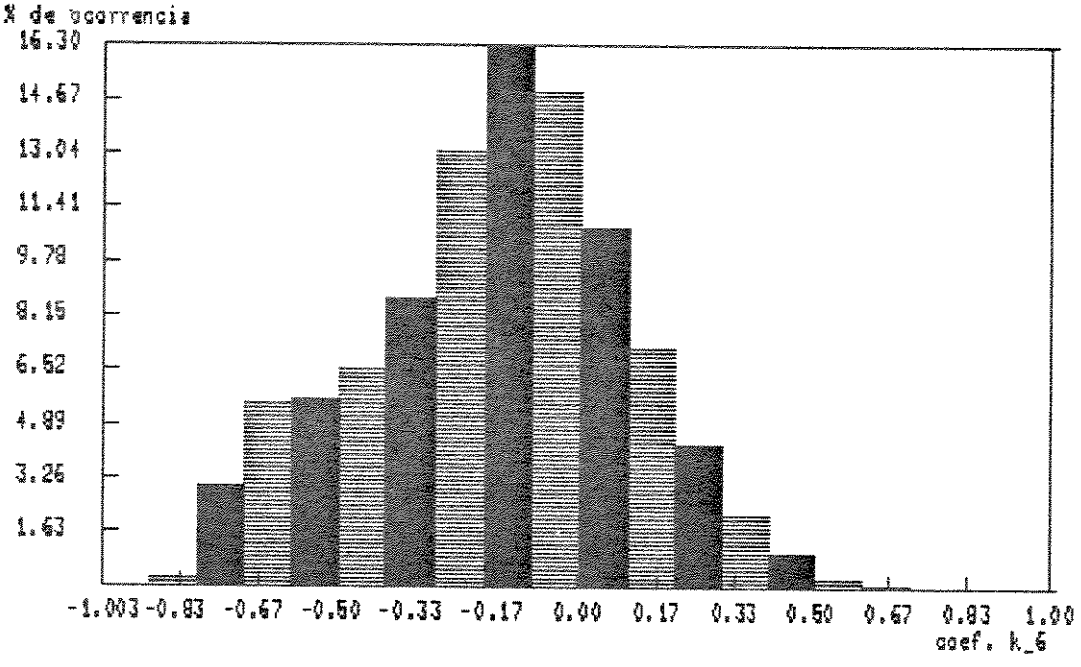


Figura 5.9: Histograma do coeficiente parcor k_6.

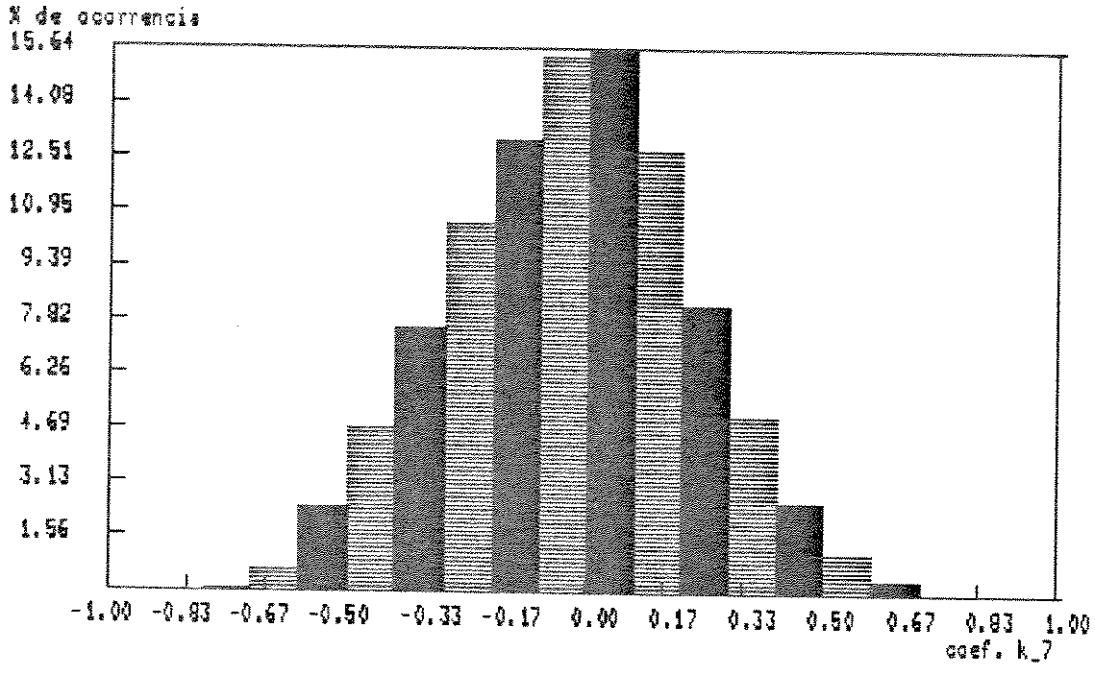


Figura 5.10: Histograma do coeficiente parcor k_7.

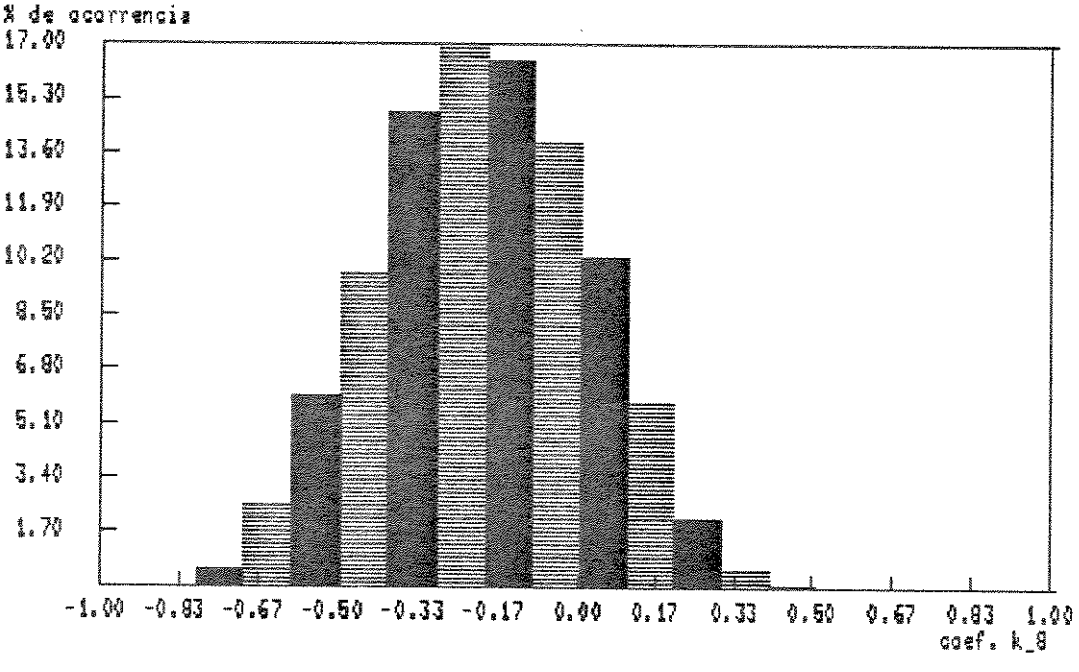


Figura 5.11: Histograma do coeficiente parcor k_8.

A partir dos dados apresentados pelos histogramas foram obtidos limitantes para cada coeficiente parcor, de modo que aproximadamente 97% das ocorrências ficassem entre esses limitantes. A tabela 5.1 mostra os limitantes para cada um dos oito coeficientes parcor.

coeficientes parcor	valor mínimo	valor máximo
k_1	-0.60	0.85
k_2	-0.95	0.10
k_3	-0.50	0.80
k_4	-0.70	0.35
k_5	-0.40	0.60
k_6	-0.75	0.35
k_7	-0.55	0.45
k_8	-0.65	0.25

Tabela 5.1: Limitantes dos coeficientes parcor.

Para obter-se voz sintetizada de boa qualidade, é necessária uma quantização precisa dos dois primeiros coeficientes. Assim, deve-se usar um maior número de níveis para a quantização desses coeficientes. Para os outros coeficientes, o número de níveis de quantização utilizados diminui à medida que o índice i do coeficiente aumenta.

Para avaliar o desempenho dos quantizadores, foram realizados testes subjetivos informais e foi calculada a distância cepstral média (apêndice I). Nos testes subjetivos, comparou-se a voz sintetizada utilizando-se coeficientes quantizados e a voz sintetizada obtida com coeficientes não quantizados.

Com o objetivo de se conseguir voz sintetizada com boa qualidade, utilizando-se o menor número de bits, foram testados vários quantizadores, quantizando-se os coeficientes parcor e os coeficientes razão log-área.

Usando-se a quantização uniforme dos coeficientes razão log-área, foram obtidos vários quantizadores, os quais apresentavam um número total de bits distinto. A tabela 5.2 mostra o número total de bits para cada quantizador e a tabela 5.3 mostra a alocação dos bits para cada coeficente nos diversos quantizadores e o degrau de quantização delta. A tabela 5.4 mostra a distância cepstral média para cada um desses quantizadores.

Nos testes subjetivos, a qualidade da voz sintetizada usando-se os quantizadores 1 e 2 foi considerada muito boa, não apresentando aparentemente nenhuma degradação. Com a utilização do quantizador 3, notou-se uma pequena degradação da voz sintetizada.

quantizador	número de bits
1	32
2	28
3	25

Tabela 5.2: Número total de bits de cada quantizador.

coeficientes razão log-área	quantizador 1		quantizador 2		quantizador 3	
	bits	delta	bits	delta	bits	delta
g_1	5	0.1218	5	0.1212	5	0.1218
g_2	5	0.1208	5	0.1208	4	0.2415
g_3	4	0.2060	4	0.2060	3	0.4120
g_4	4	0.1541	3	0.3082	3	0.3082
g_5	4	0.1396	3	0.2792	3	0.2792
g_6	4	0.1673	3	0.3346	3	0.3346
g_7	3	0.2758	3	0.2758	2	0.5515
g_8	3	0.2577	2	0.5154	2	0.5154

Tabela 5.3: Número de bits e degrau de quantização para cada coeficiente razão log-área nos três quantizadores.

quantizador	distância cepstral	distância cepstral (dB)
1	0.16	0.68
2	0.21	0.91
3	0.26	1.12

Tabela 5.4: Distância cepstral para cada quantizador.

Analisando-se a tabela 5.4 observa-se a mesma ordem de desempenho apresentada nos testes subjetivos.

Utilizando-se o método de quantização linear ' piecewise ', foram implementados quantizadores com número total de bits igual a 31 e 28. As tabelas 5.5 e 5.6 mostram os limitantes de cada região e o número de níveis de quantização de cada uma. Os coeficientes apresentados nessas tabelas são os coeficientes parcor. Também foram feitos testes com a quantização ' piecewise ' dos coeficientes razão log-área. O desempenho foi praticamente o mesmo. A tabela 5.7 mostra a distância cepstral média para cada um dos quantizadores obtidos quantizando-se os coeficientes parcor.

O desempenho dos quantizadores usando a técnica ' piecewise ' foi inferior ao desempenho dos quantizadores que usam os coeficientes razão log-área, como pode ser observado comparando-se as tabelas 5.4 e 5.7.

coeficientes parcor	limitantes das regiões				número de níveis de quantizacao			número de bits
	R_0	R_1	R_2	R_3	Q_1	Q_2	Q_3	
k_1	-0.60	-0.20	0.40	0.85	8	10	14	5
k_2	-0.95	-0.60	-0.35	0.10	19	7	6	5
k_3	-0.50	0.05	0.40	0.80	4	6	6	4
k_4	-0.70	-0.35	-0.05	0.35	4	8	4	4
k_5	-0.40	0.00	0.25	0.60	4	8	4	4
k_6	-0.75	-0.35	0.00	0.35	6	6	4	4
k_7	-0.55	-0.20	0.10	0.45	2	4	2	3
k_8	-0.65	-0.35	0.05	0.25	2	4	2	3

Tabela 5.5: Regioēs e número de níveis do quantizador com 32 bits, usando a técnica ' piecewise '.

coeficientes parcor	limitantes das regiões				número de níveis de quantizacao			número de bits
	R_0	R_1	R_2	R_3	Q_1	Q_2	Q_3	
k_1	-0.60	-0.20	0.40	0.85	8	10	14	5
k_2	-0.95	-0.60	-0.35	0.10	19	7	6	5
k_3	-0.50	0.05	0.40	0.80	2	3	3	3
k_4	-0.70	-0.35	-0.05	0.35	2	4	2	3
k_5	-0.40	0.00	0.25	0.60	2	4	2	3
k_6	-0.75	-0.35	0.00	0.35	3	3	2	3
k_7	-0.55	-0.20	0.10	0.45	2	4	2	3
k_8	-0.65	-0.35	0.05	0.25	2	4	2	3

Tabela 5.6: Regiões e número de bits do quantizador com 28 bits, usando a técnica ' piecewise '.

quantizador	distância cepstral	distância cepstral (dB)
32 bits	0.32	1.38
28 bits	0.40	1.74

Tabela 5.7: Distância cepstral dos quantizadores obtidos usando a técnica ' piecewise '.

5.2.6 Quantização do Ganho do modelo LPC

Para a quantização do ganho LPC, utilizou-se um histograma do mesmo mostrando a freqüência de ocorrência de cada valor . Esse histograma foi traçado usando o mesmo segmento de voz utilizado para traçar os histogramas dos coeficientes parcor. A figura 5.12 mostra o histograma do ganho LPC.

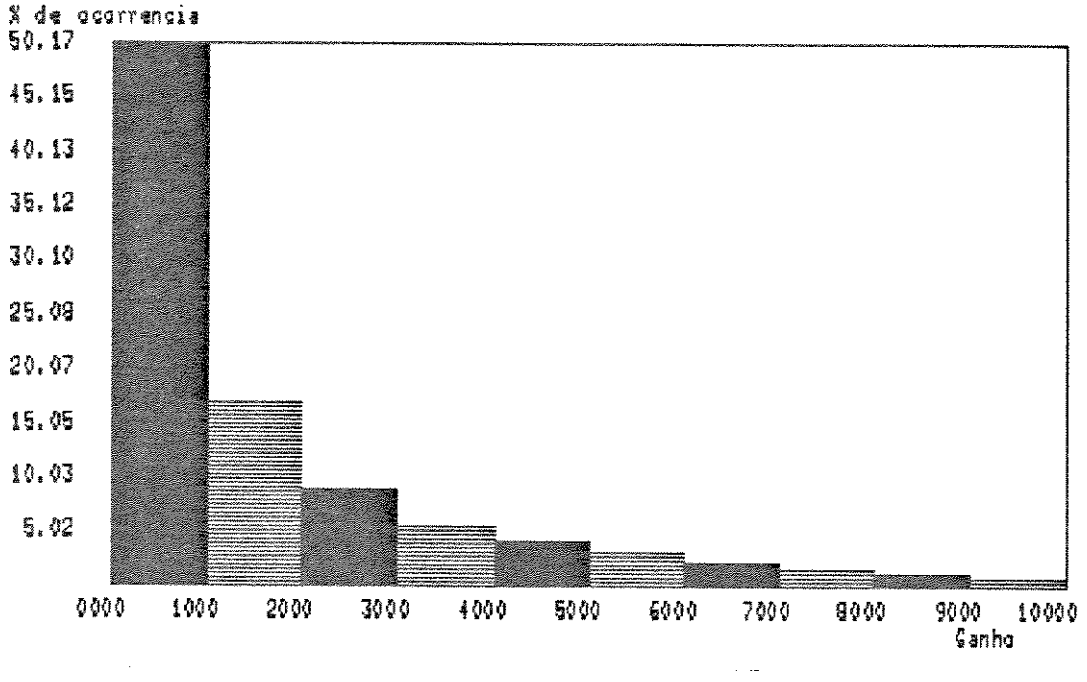


Figura 5.12: Histograma do ganho LPC.

Na quantização do ganho, foram testados os seguintes tipos de quantização:

- Quantização uniforme do ganho limitado em 9000 unidades.
- Quantização utilizando a técnica ' piecewise '. A tabela 5.8 mostra as regiões e o número de níveis de quantização de cada região. Para tornar os resultados pouco dependentes do comprimento do sinal de excitação, esses limitantes foram normalizados por $\sqrt{N}$, onde N é o número de amostras do sinal de excitação. Na implementação desse quantizador, usou-se N igual a 160.
- Quantização uniforme do logaritmo do ganho.
- Quantização do logaritmo do ganho utilizando a técnica ' piecewise '.

limitante superior de cada região	número de níveis de quantização
89	14
178	6
356	6
712	6

Tabela 5.8: Limitantes das regiões e número de níveis de quantização utilizados para quantizar o ganho.

Todos os quantizadores foram implementados usando 5 bits. O melhor resultado em teste subjetivos foi conseguido quantizando-se o ganho com a técnica ' piecewise '. A relação sinal-ruído de quantização média obtida por esse quantizador, foi 9.08 dB.

5.2.7 Quantização do Período de Pitch

O período de pitch foi quantizado uniformemente com 6 bits. Utilizou-se um nível de quantização para representar o valor zero, o qual indica a não sonoridade do segmento de voz. O valor mínimo do período de pitch quantizado é igual a 20 e o valor máximo é 146. A utilização de um menor número de bits para quantizar o período de pitch faz com que apareça um ' eco metálico ' na voz sintetizada. Esse efeito foi observado para vozes femininas e é devido ao fato do erro de quantização ser maior para as vozes femininas pois o período de pitch das mesmas é menor. A tabela 5.9 apresenta a relação sinal-ruído de quantização média para o quantizador de 6 bits e para o quantizador de 5 bits.

número de bits	relação sinal-ruído (db)
6	37.65
5	32.65

Tabela 5.9: Relação sinal-ruído de quantização média dos quantizadores do período de pitch.

5.3 INTERPOLAÇÃO

Explorando a dependência entre segmentos consecutivos do sinal de voz, a interpolação linear pode ser usada para reduzir em 50% a taxa de bits. Nesse processo, somente os parâmetros dos quadros ímpares são transmitidos e na recepção, os parâmetros dos quadros pares são obtidos por interpolação linear.

Para o período de pitch, durante as transições sonoro/não sonoro e transições não sonoro/sonoro, o período de pitch do quadro sonoro é repetido para o quadro que não foi transmitido.

Para os parâmetros LPC, o melhor desempenho em testes subjetivos foi conseguido pela interpolação dos coeficientes parcor. A diferença entre os resultados obtidos pela interpolação dos coeficientes parcor e pela interpolação dos coeficientes razão log-área é muito pequena e às vezes quase imperceptível. A tabela 5.10 mostra a distância cepstral média obtida, utilizando-se a interpolação dos coeficientes parcor e dos coeficientes razão log-área, os quais foram quantizados com 32 bits.

coeficientes	distância cepstral	distância cepstral (dB)
k_i	0.39	1.71
g_i	0.40	1.74

Tabela 5.10: Distância cepstral média para a interpolação dos coeficientes parcor e dos coeficientes razão log-área, quantizados com 32 bits.

Para o ganho LPC foram realizados testes interpolando-se o ganho e o logaritmo do ganho. Conseguiu-se o melhor resultado interpolando-se o ganho. A tabela 5.11 mostra a relação sinal-ruído média obtida ao realizar-se a interpolação do ganho e do logaritmo do ganho.

parâmetros	relação sinal ruído (dB)
ganho	7.91
logaritmo do ganho	7.68

Tabela 5.11: Relação sinal-ruído média obtida interpolando-se o ganho e o logaritmo do ganho.

5.4 VOCODERS IMPLEMENTADOS

Utilizando-se a quantização escalar e a interpolação dos parâmetros do vocoder, foram obtidos vocoders operando em taxas que variam de 2200 bit/s a 925 bit/s. Os parâmetros do vocoder foram calculados a uma taxa de 50 quadros por segundo. A tabela 5.12 apresenta o número de bits utilizados para quantizar cada parâmetro para cada vocoder implementado.

A qualidade da voz sintetizada pelos vocoder 1 e 2 foi considerada muito boa, não apresentando degradação em relação ao vocoder sem quantização. O vocoder 3 apresentou uma pequena degradação.

	vocoder 1	**vocoder 2**	**vocoder 3**
parâmetros	número de bits	número de bits	número de bits
ganho	5	5	5
período de pitch	6	6	6
coeficientes razão log-área	32	28	25
sicronização	1	1	1
número de bits por quadro	44	40	37
taxa (bit/s)	2200	2000	1850
distância cepstral	0.5	0.38	0.41
distância cepstral (dB)	1.5	1.64	1.77

Tabela 5.12: Vocoders obtidos usando quantização escalar.

Através do uso da interpolação linear dos parâmetros, conseguiu-se reduzir em 50% as taxas dos vocoders. Esses vocoders apresentam uma degradação na voz sintetizada, mas a inteligibilidade da mesma é perfeita. A tabela 5.13 apresenta as taxas de bits e as distâncias cepstrais médias obtidas para esses vocoders.

vocoder	taxa (bit/s)	distância cepstral	distância cepstral (dB)
1	1100	0.69	3.00
2	1000	0.71	3.09
3	925	0.73	3.16

Tabela 5.13: Vocoders obtidos usando interpolação linear.

Capítulo 6

QUANTIZAÇÃO VETORIAL

6.1 INTRODUÇÃO

No capítulo anterior foi apresentada a quantização escalar, onde cada um dos parâmetros é quantizado separadamente. Quando um conjunto de parâmetros é quantizado conjuntamente, como um vetor, o processo é conhecido como quantização vetorial.

Quantização vetorial é um processo de remoção de redundância, o qual faz uso das seguintes propriedades dos vetores: dependência linear, forma da função densidade de probabilidade, dependência não linear e dimensionalidade do vetor. Esse tipo de quantização requer um alto custo computacional e é utilizado principalmente em taxas de transmissão da ordem de 1 bit por parâmetro.

6.2 PRINCÍPIOS DA QUANTIZAÇÃO VETORIAL

Um quantizador vetorial K-dimensional de N níveis é um processo que determina para cada vetor de entrada $x = (x_1, ..., x_K)$, um vetor de reprodução $\hat{x}_i = q(x)$, o qual pertence a um alfabeto de reprodução $\hat{\mathrm{A}} = \{\hat{x}_i\}$, $i = 1, ..., N$. O alfabeto $\hat{\mathrm{A}}$ é denominado ' codebook ', N é o número de vetores código e cada $\hat{x}_i = (\hat{x}_{i1}, ..., \hat{x}_{iK})$ é um vetor código.

No projeto de um ' codebook ', divide-se o espaço K-dimensional do vetor x em N regiões ou células C_i, $(1 \leq i \leq N)$, sendo que a cada célula C_i está associado um vetor $\hat{x}_i$. Um vetor x é quantizado como $\hat{x}_i$ se x está contido em C_i. A figura 6.1 mostra as células C_i em um espaço de duas dimensões.

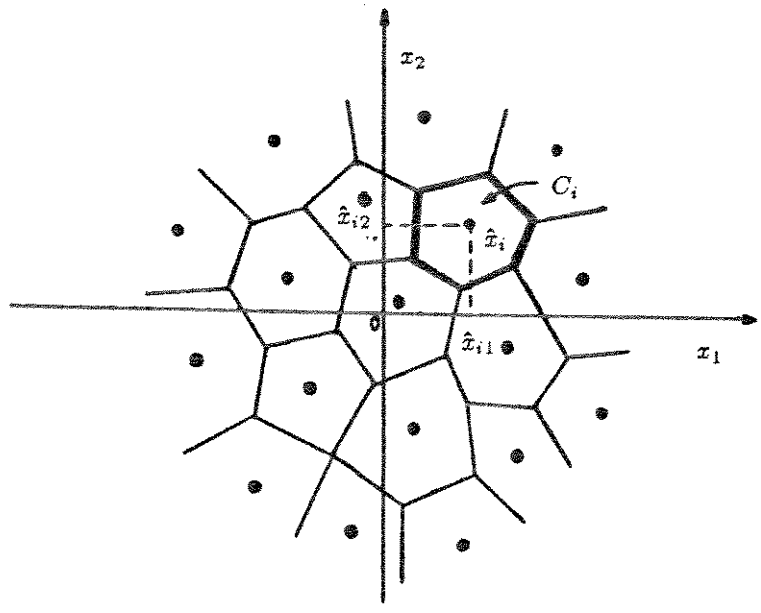


Figura 6.1: Células Ci de um espaço de duas dimensões.

Para ser transmitido ou armazenado, cada vetor $\hat{x}_i$ é codificado em uma palavra de dígitos binários c_i de comprimento igual a b_i bits. A taxa de transmissão T é dada por:

$$T = B.F_c \tag{6.1}$$

onde:

$$B = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} b_i \tag{6.2}$$

e F_c é o número de palavras transmitidas por segundo.

O número médio de bits por dimensão é dado por:

$$R = \frac{B}{K} \tag{6.3}$$

Quando um vetor x é quantizado como $\hat{x}_i$, resulta um erro de quantização. Esse erro é avaliado por uma medida de distorção definida entre x e $\hat{x}_i$. No projeto de um quantizador vetorial, deve-se minimizar a distorção na saída do mesmo. Para isso, deve-se escolher uma medida de distorção apropriada.

6.2.1 Medidas de distorção

A medida de distorção d é uma indicação do custo de um vetor $\hat{x}_i$ representar um vetor x. Idealmente, uma medida de distorção deve ser subjetivamente significativa, para que diferenças em valores de distorção possam ser usadas como indicação de diferenças na qualidade da voz.

As principais medidas de distorção são:

Erro quadrático

É a mais comum medida de distorção, principalmente devido à sua simplicidade. Essa medida é definida por [12]:

$$d_2(x, \hat{x}_i) = \sum_{j=1}^{K} |x_j - \hat{x}_{ij}|^2 \tag{6.4}$$

Uma forma mais geral para essa medida de distorção é:

$$d_r(x, \hat{x}_i) = \sum_{j=1}^{K} |x_j - \hat{x}_{ij}|^r \tag{6.5}$$

Para $r = 2$, as equações 6.4 e 6.5 tornam-se iguais.

Erro quadrático ponderado

Nessa medida, para parâmetros diferentes podem ser atribuídos pesos diferentes e assim eles contribuem de forma diferente para a distorção. Define-se essa medida como [13]:

$$d_w(x, \hat{x}_i) = (x - \hat{x}_i)' W (x - \hat{x}_i) \tag{6.6}$$

onde W é uma matriz definida positiva.

Medida de distorção de Itakura Saito

A distorção de Itakura Saito entre um sinal e seu modelo de predição linear é definida no domínio do tempo como [14]:

$$d_{IS}(x, \hat{x}_i) = \frac{\hat{a}_i' R_x' \hat{a}_i}{\hat{\alpha}^2} - \ln\left(\frac{\alpha_\infty^2}{\hat{\alpha}^2}\right) - 1 \tag{6.7}$$

onde: $\hat{a}_i' = (1, -\hat{a}_{i1}, ..., -\hat{a}_{iP})$. Os coeficientes $\hat{a}_{ik}$ são os coeficientes do preditor correspondente a $\hat{x}_i$

$R_x^{'}$ é a matriz de autocorrelação aumentada $(P+1) \times (P+1)$ do vetor de entrada x

$\hat{\alpha}$ é o ganho do modelo LPC quantizado para $\hat{x}_i$

α_∞^2 é a energia residual do modelo LPC de ordem infinita para x

A equação 6.7 também pode ser escrita da seguinte forma [15]:

$$d_{IS}(x, \hat{x}_i) = \ln\left(\frac{\hat{a}_i^t R_x^{'} \hat{a}_i}{\alpha_\infty^2}\right) + \left(\frac{\hat{a}_i^t R_x^{'} \hat{a}_i}{\hat{\alpha}^2} - \ln\left(\frac{\hat{a}_i^t R_x^{'} \hat{a}_i}{\hat{\alpha}^2}\right) - 1\right) \tag{6.8}$$

O primeiro termo da equação 6.8 independe de $\hat{\alpha}$ e é conhecido como distorção do ganho otimizado ou distorção de Itakura. Esse termo representa o valor da distorção quando o sinal de voz é representado pelo modelo LPC com coeficientes $\hat{a}_i$ e ganho do modelo LPC igual a $\sqrt{\hat{a}_i^t R_x^{'} \hat{a}_i}$. O termo à direita é a contribuição para a distorção quando o ganho $\sqrt{\hat{a}_i^t R_x^{'} \hat{a}_i}$ é quantizado por $\hat{\alpha}$.

Medida de distorção de Itakura Saito Modificada

Uma forma modificada da distorção de Itakura Saito entre um vetor de coeficientes do preditor $a = (a_1, \ldots a_P)^t$ e um outro vetor de coeficientes do preditor $\hat{a}_i^{'} = (\hat{a}_{i1}, \ldots, \hat{a}_{iP})^t$ é dada por [13]:

$$d_M(x, \hat{x}_i) = (a - \hat{a}_i^{'})^t \Phi_x (a - \hat{a}_i^{'}) \tag{6.9}$$

onde: Φ_x é a matriz de autocorrelação normalizada $(P \times P)$ correspondente ao vetor x, a qual é dada por:

$$\Phi_x(i-k) = \frac{R_x(i-k)}{R_x(0)} \quad 0 \leq i, k \leq P-1 \tag{6.10}$$

6.2.2 Projeto do ' codebook '

Um quantizador com N níveis é considerado ótimo se a distorção média é minimizada sobre todos os quantizadores de N níveis. Para que um quantizador seja ótimo são necessárias duas condições [13]:

- O quantizador deve ser projetado usando a distorção mínima, isto é, o quantizador escolhe o vetor código que resulta na distorção mínima em relação ao vetor de entrada.

$$q(x) = \hat{x}_i \qquad se \;\; d(x, \hat{x}_i) \leq d(x, \hat{x}_j) \;\; , j \neq i, \;\; 1 \leq j \leq N \tag{6.11}$$

- Cada vetor código $\hat{x}_i$ é escolhido para minimizar a distorção média na célula C_i, a qual é dada por:

$$D_i = E(d(x, \hat{x}_i) \;/\; x \in C_i) \tag{6.12}$$

$$D_i = \frac{1}{M_i} \sum_{x \in C_i} d(x, \hat{x}_i) \tag{6.13}$$

onde: M_i é o número de vetores na célula C_i.

O vetor $\hat{x}_i$ é denominado centróide da célula C_i. O cálculo do centróide depende da medida de distorção utilizada.

A geração de um ' codebook ', o qual minimiza uma medida de distorção sobre uma grande seqüência de treinamento, requer um processo iterativo. Um método para o projeto de ' codebook ' é um algoritmo iterativo conhecido como algoritmo ' K-means ' ou algoritmo de Lloyd [13].

Esse algoritmo divide a sequência de treinamento em N células C_i , satisfazendo as condições necessárias para ser considerado ótimo. Os passos desse algoritmo são os seguintes:

1. Inicialização

   O índice m de iteração é igualado a zero. Um conjunto inicial de vetores código é escolhido por um método adequado, $\{\hat{x}_i(0) \;,\; 1 \leq i \leq K\}$.

2. Classificação

   Cada componente da sequência de treinamento é classificado em uma das células C_i, pela regra do vizinho mais próximo,

$$x \in C_i \qquad \text{se} \;\; d(x, \hat{x}_i) \leq d(x, \hat{x}_j) \;\; \text{para} \;\; i \neq j \tag{6.14}$$

3. Calcula-se a distorção total $D(m)$. Se a diferença entre a distorção total $D(m)$ na iteração m e a distorção $D(m-1)$ está abaixo de um limiar, o algoritmo é finalizado.

4. Atualização dos vetores código

   O índice de iteração é aumentado de uma unidade ($m \leftarrow m + 1$). Os vetores código são atualizados, calculando-se os centróides de cada célula C_i e assumindo esses centróides como novos vetores código. Volte para o passo 2.

Esse algoritmo converge para um mínimo local. Assim, o ' codebook ' gerado é apenas localmente ótimo. Um ' codebook ' globalmente ótimo pode ser encontrado repetindo-se esse algoritmo com vários ' codebooks ' iniciais. O ' codebook ' que apresentar a menor distorção é escolhido.

O custo computacional para projetar-se um ' codebook ' com N vetores código de dimensão K, utilizando-se uma seqüência de treinamento com M vetores e I iterações do algoritmo de Lloyd é dado por [13]:

$$C = K.M.N.I \tag{6.15}$$

6.2.3 Tipos de quantização

Quantização com Busca Exaustiva

Nesse tipo de quantização, é calculada a distorção entre o vetor a ser quantizado e cada um dos vetores código, escolhendo-se o que resultar na menor distorção. Assim, são testados todos os vetores código para a quantização de cada vetor de entrada. Para um quantizador com N níveis, o número de distorções calculadas para quantizar um único vetor é N. Assumindo que para um quantizador K-dimensional, o cálculo de uma distorção requer K multiplicações ou adições, o custo computacional para quantizar um vetor é $N.K$ e o custo de armazenamento também é $N.K$ [13]. Os custos computacionais são proporcionais a N. O aumento de um bit no número de bits, equivale a dobrar o valor de N. Assim, o aumento de 1 bit no número de bits utilizado na quantização, dobra os custos computacionais.

Quantização com busca por árvore

O método mais simples dessa classe de quantizadores vetoriais é a árvore binária. Nesse método o espaço K-dimensional é divido de forma que o número de distorções calculadas para quantizar um único vetor seja proporcional a $\log_2 N$ (N deve ser uma potência de 2). Primeiramente, o espaço é divido em duas regiões, usando o algoritmo de Lloyd. A seguir, cada uma das duas regiões é divida em outras duas regiões. Esse procedimento é realizado até que o número de regiões em um mesmo estágio seja igual a N. A figura 6.2 mostra um esquema de uma árvore binária uniforme com $N = 8$.

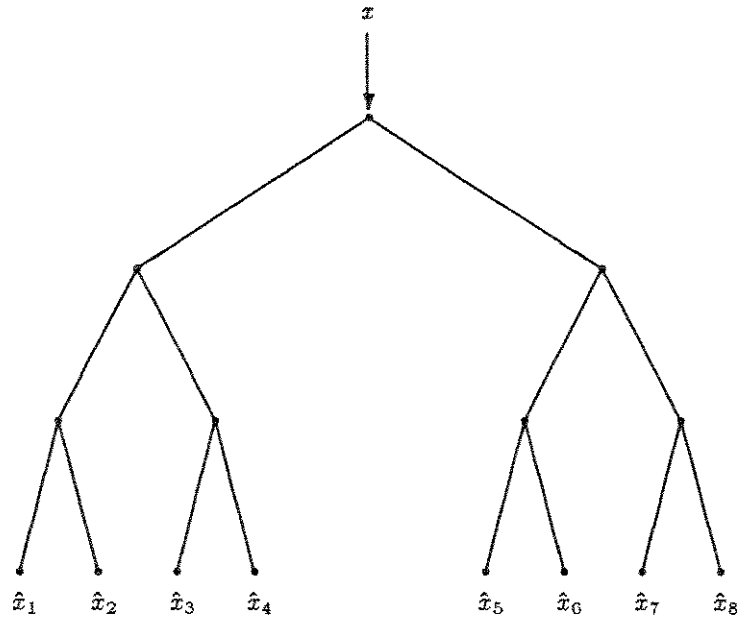


Figura 6.2: Esquema de árvore binária unforme com $N = 8$.

Nesse método vários ' codebooks ' de dois níveis são gerados, sendo que cada nó da árvore está associado a um vetor código de um ' codebook '. Os vetores código associados aos nós do último estágio da árvore formam um ' codebook ' de N níveis, o qual é usado para quantizar os vetores de entrada.

Quando um vetor x é quantizado, são realizados $2\log_2 N$ cálculos de distorção, e em cada nó da árvore é escolhido o vetor código do estágio seguinte que resultar na menor distorção.

O custo computacional para quantizar um vetor x é:

$$\mathcal{C} = 2K \log_2 N \tag{6.16}$$

e a quantidade de memória necessária é:

$$\mathcal{M} = 2K(N-1) \tag{6.17}$$

Embora a quantidade de memória necessária seja o dobro da quantidade requerida pela quantização por busca exaustiva, o custo computacional é menor. Isso é conseguido às custas de uma redução no desempenho do quantizador.

Para assegurar uma distorção menor e maximizar a utilização dos bits, árvores não uniformes podem ser usadas. Nesse caso, o número de níveis do quantizador pode ser qualquer inteiro, e não necessariamente uma potência de 2. A figura 6.3 mostra um esquema de árvore não uniforme.

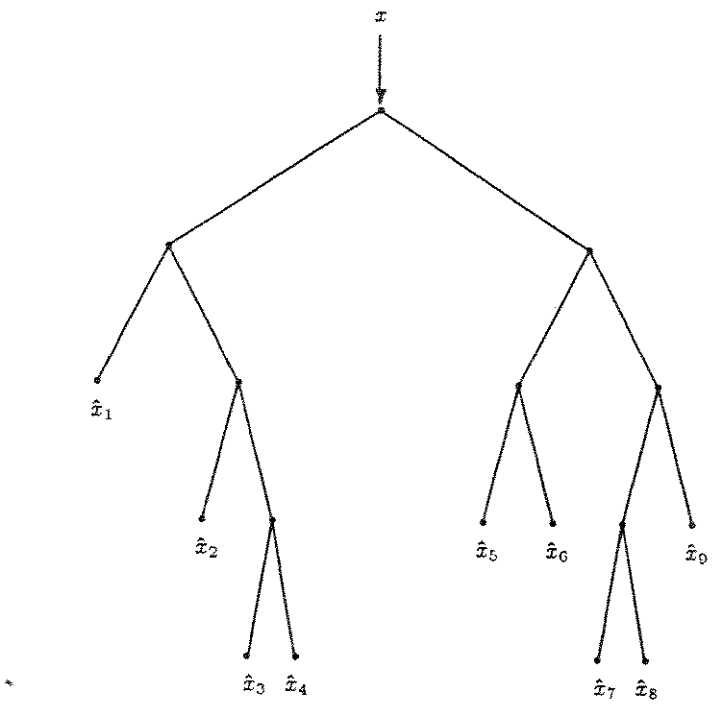


Figura 6.3: Esquema de árvore não uniforme.

Árvores não binárias também podem ser usadas, sendo que em cada estágio i são usados ' codebooks ' de taxa R_i. Nesse método, é alcançada uma maior redução dos custos computacionais.

' Product Code '

Um outro método para reduzir os custos computacionais é projetar um ' codebook ' como um produto cartesiano dos vetores código de outros ' codebooks '. O ' codebook ' gerado dessa maneira é conhecido como ' Product Code '.

Seja uma coleção de L ' codebooks ' de dimensão k_i e n_i vetores código, o ' product code ' resultante tem dimensão K e número de vetores código N dados por:

$$K = \sum_{i=1}^{L} k_i \tag{6.18}$$

$$N = \prod_{i=1}^{L} n_i \tag{6.19}$$

O custo computacional associado a esse ' codebook ' é igual a:

$$C = \sum_{i=1}^{L} n_i k_i \tag{6.20}$$

Para que o ' product code ' seja ótimo, os ' codebooks ' componentes do mesmo não podem ser projetados independentemente, a não ser que eles sejam estatisticamente independentes.

Na prática, para reduzir os custos computacionais, a quantização independente é empregada no projeto, resultando em ' codebooks ' sub-ótimos.

Um exemplo de ' product code ' é o ' product code LPC gain-shape '. Nesse ' codebook ', primeiro ocorre a quantização dos coeficientes LPC (shape) e os coeficientes LPC quantizados são usados para quantizar o ganho LPC, de modo que a distorção total seja minimizada.

6.2.4 ' Codebook ' Inicial

O ' codebook ' inicial pode ser obtido usando-se algum ' codebook ' simples com o número de níveis desejado ou usando-se um ' codebook ' com um pequeno número de níveis e recursivamente obter um ' codebook ' com o número de níveis desejado. Como exemplos de técnicas usadas para gerar um ' codebook ' inicial pode-se citar: Códigos aleatórios, ' product code ' e ' splitting '.

Códigos Aleatórios

Esse código é obtido escolhendo-se aleatoriamente N vetores da seqüência de treinamento, onde N é o número de vetores código desejado. Pode-se escolher N vetores consecutivos ou N vetores espaçados por uma razoável distância.

Esse método praticamente não apresenta custos computacionais , sendo que sua utilização é recomendada apenas para grandes valores de N.

' Product Code '

Pode-se obter um ' codebook ' de dimensão K e N vetores código usando-se um ' product code '. A geração desse ' codebook ', pode ser feita utilizando-se o produto cartesiano de K quantizadores escalares com número de bits igual a $\log_2 N/K$. Pode-se usar quantizadores uniformes idênticos.

' Splitting ' [14]

Essa técnica é usada para construir ' codebooks ' com um grande número de níveis , a partir de ' codebooks ' menores, mas com a mesma dimensão. O primeiro passo nesse procedimento consiste em encontrar o centróide da sequência de treinamento. A seguir, perturba-se esse vetor código, de modo a obter-se dois vetores código. Usando-se esses vetores como ' codebook ' inicial, utiliza-se o algoritmo de Lloyd para obter-se um ' codebook ' de dois níveis com distorcão total menor que um determinado limiar. Cada um desses vetores código é perturbado e o algoritmo de Lloyd é utilizado para gerar um ' codebook ' de quatro níveis. Esse procedimento é repetido até obter-se um ' codebook ' com o número de níveis desejado. A figura 6.4 mostra as fases dessa técnica [14].

O procedimento para a obtenção do ' codebook ' inicial pela técnica ' splitting ' [14] pode ser descrito da seguinte forma:

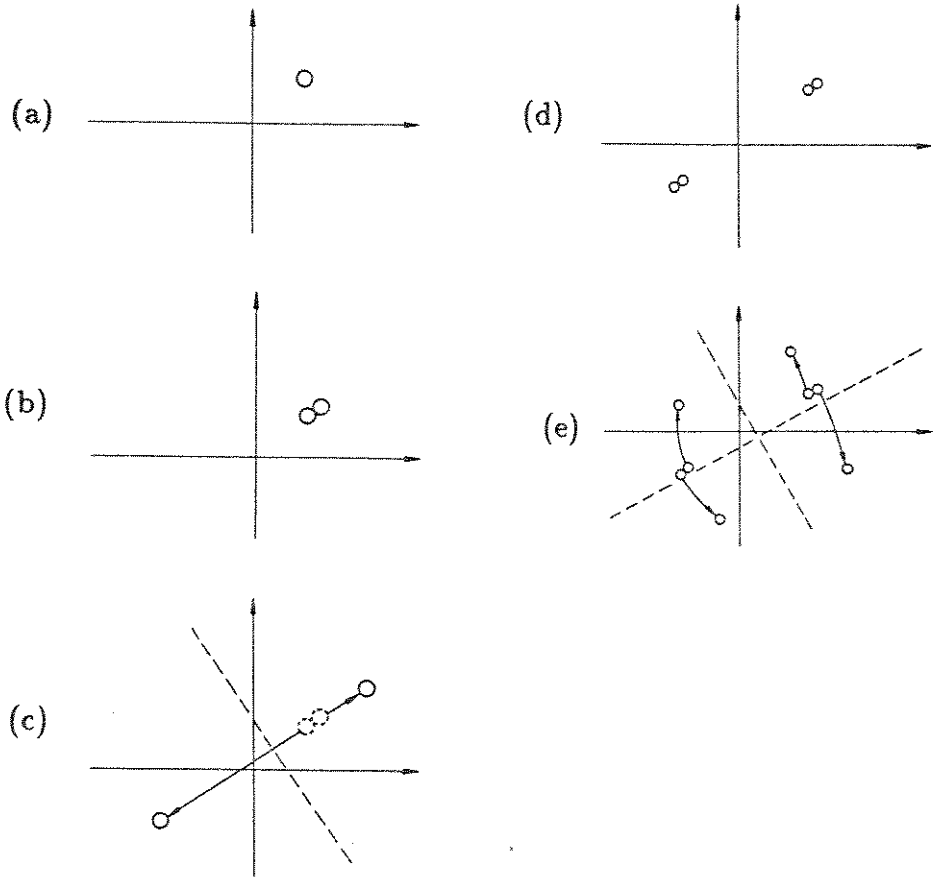
1. Calcule o centróide $\hat{x}$ da seqüência de treinamento e faça $Q = 1$ e $\hat{A}_0(1) = \hat{x}$
2. Dado o ' codebook ' $\hat{A}(Q)$, contendo Q vetores $\{\hat{x}_i\ ,\ \ i = 1, ..., Q\}$, perturbe cada vetor de forma a obter dois vetores, $\hat{x}_i + \epsilon$ e $\hat{x}_i - \epsilon$. Assim, obtém-se um ' codebook ' com $2Q$ vetores. Substitua Q por $2Q$.
3. Se Q é igual a N, o algoritmo está terminado. Caso contrário, utilize o algoritmo de Lloyd para obter um ' codebook ' com Q vetores código e distorção média menor que um determinado limiar. A seguir, retorne ao ítem 2.

6.3 ' CODEBOOKS ' IMPLEMENTADOS

Foram implementados ' codebooks ' com busca exaustiva, com busca por árvore e o ' product code gain-shape '. Esses ' codebooks ' foram projetados usando-se o algoritmo de Lloyd, sendo que o mesmo era finalizado quando a diferença entre as distorções médias de duas iterações consecutivas fosse menor que $0,01\%$. O ' codebook ' inicial para cada ' codebook ' foi gerado usando a mesma técnica. Utilizou-se também a mesma sequência de treinamento para todos os algoritmos.

' Codebook ' Inicial

O ' codebook ' inicial foi projetado usando-se a técnica ' splitting ' com limiar para perturbação dos vetores código igual a 1%, e limiar para finalizar o algoritmo também igual a 1%.



(a) Centróide da seqüência de treinamento. (b) Perturbação do centróide para obter-se dois vetores código. (c) ' Codebook ' ótimo com dois vetores código. (d) Perturbação dos vetores código para obter-se uma estimativa inicial de um ' codebook ' com quatro vetores código. (e) ' codebook ' ótimo com quatro vetores código.

Figura 6.4: Técnica ' splitting '.

Seqüência de Treinamento

A seqüência de treinamento era formada por vozes de 10 locutores, sendo 5 homens e 5 mulheres. Cada locutor contribuiu com 1 minuto de voz, resultando em uma seqüência de treinamento com 10 minutos de voz. Utilizou-se 8000 Hz como freqüência de amostragem para digitalizar a voz. Uma análise LPC de ordem 8 e com coeficiente de pré-ênfase igual a 0.9 foi realizada usando-se um intervalo de análise de 30 ms (240 amostras) e calculando-se os coeficientes a cada 20 ms. Isso resultou em uma sequência de treinamento com aproximadamente 30000 vetores. Nessa seqüência não foram isolados os trechos de silêncio.

Para um bom desempenho do ' codebook ' é necessário que a sequência de treinamento apresente no mínimo 10 vetores para cada vetor código [13].

6.3.1 Algoritmos com busca exaustiva

Implementou-se algoritmos utilizando-se as seguintes medidas de distorção: erro quadrático, medida de distorção de Itakura Saito e medida de distorção de Itakura Saito modificada.

Erro quadrático

Com essa medida de distorção implementou-se algoritmos para quantizar os coeficientes parcor e os coeficientes razão log-área, com números de bits iguais a 8 e 10.

Cálculo do centróide

O centróide $\hat{x}_i$ de cada célula C_i é dado por [12]:

$$\hat{x}_i = \frac{1}{M_i} \sum_{j:x_j \in C_i} x_j \tag{6.21}$$

onde: x_j são os vetores da seqüência de treinamento que estão na célula C_i

M_i é o número de vetores na célula C_i

As tabelas 6.1 e 6.2 apresentam a distância cepstral média para cada um dos quantizadores obtidos usando o erro quadrático, para segmentos de voz independentes da seqüência de treinamento e para segmentos de voz pertencentes à mesma, respectivamente.

número de bits	8		10	
coeficientes	k_i	g_i	k_i	g_i
distância cepstral	0.70	0.71	0.62	0.62
distância cepstral (dB)	3.03	3.08	2.69	2.69

Tabela 6.1: Distância cepstral média para quantizadores usando o erro quadrático como medida de distorção. Essas medidas foram obtidas utilizando-se segmentos de voz independentes da seqüência de treinamento.

número de bits	8		10	
coeficientes	k_i	g_i	k_i	g_i
distância cepstral	0.59	0.58	0.48	0.47
distância cepstral (dB)	2.55	2.54	2.10	2.08

Tabela 6.2: Distância cepstral média para quantizadores usando o erro quadrático como medida de distorção. Essas medidas foram obtidas utilizando-se segmentos de voz pertencentes à seqüência de treinamento.

Medida de Itakura Saito modificada

Os ' codebooks ' obtidos utilizando-se essa medida de distorção foram usados para quantizar os coeficientes LPC, com os seguintes números de bits: 7, 8, 9 e 10.

Cálculo do centróide

O centróide $\hat{x}_i$ de cada célula C_i é calculado usando-se a seguinte equação [12]:

$$\hat{x}_i = \left[\sum_{j:x_j \in C_i} \frac{R_{x_j}}{R_{x_j}(0)} \right]^{-1} \sum_{j:x_j \in C_i} \frac{R_{x_j}}{R_{x_j}(0)} a_j' \qquad (6.22)$$

onde: R_{x_j} é a matriz de autocorrelação $(P \times P)$ associada a x_j.

a_j são os coeficientes do preditor associado a x_j.

A solução da equação 6.22 pode ser obtida utilizando-se o algoritmo de Durbin (capítulo 3).

As tabelas 6.3 e 6.4 mostram a distância cepstral média obtida para esses quantizadores, usando-se segmentos de voz independentes da seqüência de treinamento e segmentos de voz pertencentes à mesma, respectivamente. As tabelas 6.5 e 6.6 apresentam a distorção de Itakura Saito modificada obtida para esses quantizadores, usando-se segmentos de voz independentes da seqüência de treinamento e segmentos de voz pertencentes à seqüência de treinamento, repectivamente.

número de bits	7	8	9	10
distância cepstral	0.67	0.62	0.56	0.52
distância cepstral (dB)	2.90	2.68	2.45	2.28

Tabela 6.3: Distância cepstral média para quantizadores usando a medida de distorção de Itakura Saito modificada. Essas medidas foram obtidas usando-se segmentos de voz independentes da seqüência de treinamento.

número de bits	7	8	9	10
distância cepstral	0.56	0.51	0.45	0.41
distância cepstral (dB)	2.43	2.21	1.98	1.78

Tabela 6.4: Distância cepstral média para quantizadores usando a medida de distorção de Itakura Saito modificada. Essas medidas foram obtidas usando-se segmentos de voz pertencentes à seqüência de treinamento.

número de bits	7	8	9	10
distorção de Itakura Saito modificada (10^{-2})	8.95	7.71	6.35	5.54

Tabela 6.5: Distorção de Itakura Saito modificada, obtida usando-se segmentos de voz independentes da seqüência de treinamento.

número de bits	7	8	9	10
distorção de Itakura Saito modificada (10^{-2})	6.88	5.59	4.51	3.55

Tabela 6.6: Distorção de Itakura Saito modificada, obtida usando-se segmentos de voz pertencentes à seqüência de treinamento.

Medida de distorção de Itakura Saito

Com essa medida de distorção foram projetados ' codebooks ' para quantizar os coeficientes LPC e ganho LPC conjuntamente, com os seguintes números de bits: 7, 8, 9 e 10.

A medida de distorção de Itakura Saito é dada por:

$$d_{IS}(x, \hat{x}_i) = \frac{\hat{a}_i^t R_x' \hat{a}_i}{\hat{\alpha}^2} + \ln \hat{\alpha}^2 - \ln \alpha_\infty^2 - 1 \qquad (6.23)$$

Os dois últimos termos dessa equação não dependem de $\hat{x}_i$, dependendo apenas do sinal de entrada. Assim, esses termos podem ser desprezados ao avaliar-se a distorção $d_{IS}(x, \hat{x}_i)$.

Cálculo do centróide

O centróide de cada célula C_i é obtido minimizando-se a equação $\hat{a}_i^t R_{AI}' \hat{a}_i$, onde R_{AI}' é a média aritmética das matrizes de autocorrelação aumentadas dos vetores componentes de cada célula, e calculando-se a energia residual mínima, $\alpha = \hat{a}_i^t R_{AI}' \hat{a}_i$. O vetor $\hat{a}_i$ é obtido utilizando-se o algoritmo de Durbin [16].

As tabelas 6.7 e 6.8 apresentam a distância cepstral média para cada um dos quantizadores, usando-se segmentos de voz independentes da seqüência de treinamento e segmentos de voz pertencentes à mesma, respectivamente. Esses quantizadores foram obtidos utilizando-se a medida de distorção de Itakura Saito.

As tabelas 6.9 e 6.10 apresentam a distorção de Itakura Saito para cada um dos quantizadores, usando-se segmentos de voz independentes da seqüência de treinamento e segmentos de voz pertencentes à mesma, respectivamente. No cálculo dessa medida de distorção não foi considerado o termo $\ln \alpha_\infty^2$.

Os ' codebooks ' projetados utilizando-se a medida de distorção de Itakura Saito são ' codebooks ' ótimos. Esses ' codebooks ' apresentam desempenho superior ao desempenho de outros ' codebooks ' que utilizam o mesmo número de bits, levando-se em consideração a quantização do ganho LPC e dos coeficientes do preditor.

número de bits	7	8	9	10
distância cepstral	1.15	1.05	0.96	0.86
distância cepstral (dB)	4.98	4.57	4.20	3.75

Tabela 6.7: Distância cepstral média para quantizadores usando a medida de distorção de Itakura Saito. Essas medidas foram obtidas usando-se segmentos de voz independentes da seqüência de treinamento.

número de bits	7	8	9	10
distância cepstral	0.97	0.84	0.73	0.65
distância cepstral (dB)	4.19	3.67	3.21	2.80

Tabela 6.8: Distância cepstral média para quantizadores usando a medida de distorção de Itakura Saito. Essas medidas foram obtidas usando-se segmentos de voz pertencentes à seqüência de treinamento.

número de bits	7	8	9	10
distorção de Itakura Saito	14.02	13.93	13.84	13.76

Tabela 6.9: Distorção de Itakura Saito. Essas medidas foram obtidas usando-se segmentos de voz independentes da seqüência de treinamento.

número de bits	7	8	9	10
distorção de Itakura Saito	13.79	13.69	13.60	13.54

Tabela 6.10: Distorção de Itakura Saito. Essas medidas foram obtidas usando-se segmentos de voz pertencentes à seqüência de treinamento.

6.3.2 Algoritmos com busca por árvore

Utilizando-se busca por árvore foram projetados ' codebooks ' com 1024 vetores código (10 bits), usando-se as seguintes medidas de distorção:

- Erro quadrático (coeficientes parcor e coeficientes razão log-área)
- Medida de distorção de Itakura Saito modificada.

As distâncias cepstrais médias obtidas por esses quantizadores são mostradas nas tabelas 6.11 e 6.12, usando-se respectivamente, segmentos de voz independentes da seqüência de treinamento e segmentos de voz pertencentes à mesma.

medida de distorção	erro quadrático		Itakura Saito modificada
coeficientes	k_i	g_i	a_k
distância cepstral	0.75	0.75	1.69
distância cepstral (dB)	3.27	3.27	7.34

Tabela 6.11: Distância cepstral média para quantizadores com busca por árvore e segmentos de voz independentes da seqüência de treinamento.

medida de distorção	erro quadrático		Itakura Saito modificada
coeficientes	k_i	g_i	a_k
distância cepstral	0.87	0.87	1.45
distância cepstral (dB)	3.81	3.81	6.33

Tabela 6.12: Distância cepstral média para quantizadores com busca por árvore e segmentos de voz pertencentes à seqüência de treinamento.

6.3.3 ' Product Code Gain-Shape '

Esse ' codebook ' é formado pelo produto cartesiano dos componentes de um ' codebook ', o qual é usado para quantizar os coeficientes LPC e os componentes de outro ' codebook ' usado para quantizar o ganho LPC. O número de bits desse ' codebook ' é igual à soma dos números de bits dos ' codebooks ' componentes. Assim, usando-se 10 bits para quantizar os coeficientes LPC e 5 bits para quantizar o ganho, resulta um número total de bits igual a 15.

A medida de distorção utilizada é a medida de distorção de Itakura Saito dada por [15]:

$$d_{IS}(x,\hat{x}_i) = \ln\left(\frac{\hat{a}_i^t R_x' \hat{a}_i}{\alpha_\infty^2}\right) + \left(\frac{\hat{a}_i^t R_x' \hat{a}_i}{\hat{\alpha}^2} - \ln\left(\frac{\hat{a}_i^t R_x' \hat{a}_i}{\hat{\alpha}^2}\right) - 1\right) \qquad (6.24)$$

Essa equação pode ser escrita da seguinte forma:

$$d_{IS}(x, \hat{x}_i) = d'(x, \hat{a}_i) + d''(\hat{\alpha}, \alpha^*(x, \hat{a}_i)) \tag{6.25}$$

onde:

$$d'(x, \hat{a}_i) = \ln\left(\frac{\hat{a}_i^t R_x' \hat{a}_i}{\alpha_\infty^2}\right) \tag{6.26}$$

$$d''(\hat{\alpha}, \alpha^*(x, \hat{a}_i)) = \frac{\hat{a}_i^t R_x' \hat{a}_i}{\hat{\alpha}^2} - \ln\left(\frac{\hat{a}_i^t R_x' \hat{a}_i}{\hat{\alpha}^2}\right) - 1 \tag{6.27}$$

$$\alpha^* = \sqrt{\hat{a}_i^t R_x' \hat{a}_i} \tag{6.28}$$

No projeto desse ' codebook ', o ' codebook ' para os coeficientes LPC minimiza a distorção d', enquanto o ' codebook ' do ganho minimiza a distorção d''.

Cálculo dos centróides

- ' Codebook ' dos coeficientes LPC

  O centróide de cada célula C_i é obtido minimizando-se a equação $\hat{a}_i^t R_A' \hat{a}_i$, onde R_A' é a média aritmética das matrizes de autocorrelação aumentadas dos vetores que estão dentro da célula, normalizadas por σ^2 [15]. O valor de σ^2 depende do método empregado para gerar o ' codebook '. Utiliza-se o algoritmo de Durbin para a obtenção do vetor $\hat{a}_i$.

- ' Codebook ' do ganho

  Obtém-se o centróde de cada célula C_i utilizando-se a seguinte equação [15]:

$$\alpha_j^{*2} = \hat{a}_j^t R_x' \hat{a}_j \tag{6.29}$$

$$\hat{\alpha}_i^2 = \frac{1}{M_i} \sum_{j=1}^{M_i} \alpha_j^{*2} \tag{6.30}$$

  onde: M_i é o número de vetores na célula C_i.

Foram implementados os seguintes algoritmos:

- Otimização conjunta dos ' codebooks ' [15]
- Otimização individual dos ' codebooks ' [15]
- Algoritmo proposto por Buzo, Gray Jr., Gray e Markel (BGGM) [16]

Algoritmo com otimização conjunta (o.c.)

Nesse algoritmo o ' codebook ' para os coeficientes LPC e o ' codebook ' para o ganho LPC são calculados conjuntamente. Para o cálculo dos centróides das células C_i para o ' codebook ' dos coeficientes LPC, o valor σ é igual ao valor do ganho que minimiza d'', $\sigma^2 = \hat{\alpha}^2$. A implementação do algoritmo apresenta os seguintes passos:
Seja: N_1 número de níveis do ' codebook ' dos coeficientes
N_2 número de níveis do ' codebook ' do ganho
$\hat{\mathrm{A}}_m$ ' codebook ' do coeficientes
$\hat{\Sigma}_m$ ' codebook ' do ganho
$\mathcal{E}$ limiar para finalizar o algoritmo

1. Dados N_1, N_2, $\hat{\mathrm{A}}_0$, $\hat{\Sigma}_0$ e $\mathcal{E}$
2. Calcule as células $C_i(\hat{\mathrm{A}}_m, \hat{\Sigma}_m)$
3. Calcule a distorção $D_m = D(\hat{\mathrm{A}}_m, \hat{\Sigma}_m, C_i(\hat{\mathrm{A}}_m, \hat{\Sigma}_m))$. Se $(D_{m-1} - D_m)/D_m < \mathcal{E}$, o algoritmo está terminado. Caso contrário, continue.
4. Calcule o ' codebook ' dos coeficientes: $\hat{\mathrm{A}}_{m+1} = \hat{\mathrm{A}}(\hat{\Sigma}_m, C_i(\hat{\mathrm{A}}_m, \hat{\Sigma}_m))$
5. Calcule as células $C_i(\hat{\mathrm{A}}_{m+1}, \hat{\Sigma}_m)$
6. Calcule o ' codebook ' do ganho: $\hat{\Sigma}_{m+1} = \hat{\Sigma}(\hat{\mathrm{A}}_{m+1}, C_i(\hat{\mathrm{A}}_{m+1}, \hat{\Sigma}_m))$
7. Faça $m = m + 1$ e retorne ao item 2.

Algoritmo com otimização individual (o.i.)

Nesse algoritmo o ' codebook ' dos coeficientes LPC é gerado em primeiro lugar, minimizando d'. Esse ' codebook ' é utilizado para gerar o ' codebook ' do ganho, o qual minimiza d''. O centróide de cada célula do ' codebook ' dos coeficientes LPC é obtido utilizando-se σ igual ao valor do ganho associado ao vetor código $\hat{x}_i$, $\sigma = \alpha^*(x, \hat{a}_i)$.

O seguinte procedimento iterativo é usado para projetar o ' codebook ' dos coeficientes LPC:

1. Dados N_1, $\hat{\mathrm{A}}_0$ e $\mathcal{E}$
2. Calcule as células $C_i(\hat{\mathrm{A}}_m)$

3. Calcule a distorção $D_m = D(\hat{A}_m, C_i(\hat{A}_m))$. Se $(D_{m-1} - D_m)/D_m < \mathcal{E}$, o algoritmo está terminado. Caso contrário, continue.

4. Calcule o ' codebook ' dos coeficientes: $\hat{A}_{m+1} = \hat{A}(\hat{A}_m, C_i(\hat{A}_m))$

5. Faça $m = m + 1$ e retorne ao item 2.

O projeto do ' codebook ' do ganho utiliza esse mesmo procedimento, substituindo-se $\hat{A}_m$ por $\hat{\Sigma}_m$ e N_1 por N_2. O centróide de cada célula do ' codebook ' do ganho é obtido através da equação 6.30.

Algoritmo BGGM

Esse algoritmo difere do algoritmo de otimização individual apenas no cálculo dos centróides para o ' codebook ' dos coeficientes. Aqui, o valor de σ usado é o valor do ganho LPC associado aos vetores de entrada x, $\sigma^2 = \alpha^2$, e não o valor quantizado.

Nas tabelas 6.13 e 6.14 são apresentadas as distâncias cepstrais médias para cada um desses quantizadores, usando-se respectivamente, segmentos de voz independentes da seqüência de treinamento e segmentos de voz pertencentes à mesma.

Nas tabelas 6.15 e 6.16 são apresentadas a distorção de Itakura Saito para cada um desses quantizadores, usando-se respectivamente, segmentos de voz independentes da seqüência de treinamento e segmentos de voz pertencentes à mesma. No cálculo dessa medida de distorção não considerou-se o termo $\ln \alpha_\infty^2$.

algoritmo	o.i.	o.c.	BGGM
distância cepstral	0.63	0.63	0.58
distância cepstral (dB)	2.73	2.73	2.56

Tabela 6.13: Distância cepstral média para quantizadores usando ' product codes '. Essas medidas foram obtidas utilizando-se segmentos de voz independentes da seqüência de treinamento.

algoritmo	o.i.	o.c.	BGGM
distância cepstral	0.55	0.48	0.45
distância cepstral (dB)	2.41	2.06	1.97

Tabela 6.14: Distância cepstral média para quantizadores usando ' product codes '. Essas medidas foram obtidas utilizando-se segmentos de voz pertencentes à seqüência de treinamento.

algoritmo	o.i.	o.c.	BGGM
distorção de Itakura Saito	13.59	13.60	13.58

Tabela 6.15: Distorção de Itakura Saito para quantizadores usando ' product codes '. Essas medidas foram obtidas utilizando-se segmentos de voz independentes da seqüência de treinamento.

algoritmo	o.i.	o.c.	BGGM
distorção de Itakura Saito	13.48	13.43	13.42

Tabela 6.16: Distorção de Itakura Saito para quantizadores usando ' product codes '. Essas medidas foram obtidas utilizando-se segmentos de voz pertencentes à seqüência de treinamento.

6.4 VOCODERS IMPLEMENTADOS

A partir dos resultados da seção anterior, observa-se que os melhores desempenhos dos ' codebooks ' ocorreram para segmemtos de voz pertencentes à seqüência de treinamento. À medida que aumentamos o tamanho da seqüência de treinamento , o ' codebook ' tende a apresentar o mesmo desempenho para segmentos de voz pertencentes à seqüência de treinamento e para segmentos de voz não pertencentes à mesma. Os ' codebooks ' projetados usando-se o erro quadrático como medida de distorção apresentam um desempenho não muito inferior aos ' codebooks ' que utilizam a medida de distorção de Itakura Saito e um custo computacional muito menor. Os ' codebooks ' gerados utilizando-se a medida de distorção de Itakura Saito são ótimos e apresentam melhor desempenho que outros ' codebooks ' que utilizam o mesmo número de bits, considerando-se a quantização do ganho LPC e dos coeficientes do preditor. Entretanto, os custos computacionais são muito altos quando deseja-se um ' codebook ' com um número elevado de vetores código. Para evitar esse problema, tem-se como opção o ' product code gain shape ', o qual é um ' codebook ' sub-ótimo. Foram implementados três ' codebooks ' desse tipo, sendo que os mesmos apresentaram praticamente o mesmo desempenho. O algoritmo proposto por BGGM apresentou um desempenho levemente superior. Dessa forma, foram implementados os seguintes vocoders:

- Quantização dos coeficientes LPC utilizando-se algoritmo com busca exaustiva e medida de distorção de Itakura Saito modificada, com 10 bits. O ganho foi quantizado escalarmente com 5 bits (vocoder 1).
- Quantização dos coeficientes LPC e do ganho usando-se o algoritmo BGGM, com 15 bits (vocoder 2).

- Quantização dos coeficientes LPC e do ganho usando-se algoritmo com busca exaustiva e medida de distorção de Itakura Saito, com 10 bits (vocoder 3).
- Quantização dos coeficientes parcor usando-se algoritmo com busca exaustiva e o erro quadrático como medida de distorção, com 10 bits. O ganho foi quantizado escalarmente com 5 bits (vocoder 4).

Em todos os vocoders o período de pitch foi quantizado com 6 bits, e os parâmetros foram calculados a cada 20 ms.

As tabelas 6.17 e 6.18 mostram a taxa de bits e a distância cepstral média para cada um desses vocoders, usando-se respectivamente, segmentos de voz independentes da seqüência de treinamento e segmentos de voz pertencentes à mesma.

vocoder	1	2	3	4
distância cepstral	0.63	0.58	0.86	0.69
distância cepstral (dB)	2.74	2.56	3.75	2.99
taxa (bit/s)	1100	1100	850	1100

Tabela 6.17: Distância cepstral média e taxa de bits para os vocoders implementados usando-se quantização vetorial. Essas medidas foram obtidas para segmentos de voz independentes da seqüência de treinamento.

vocoder	1	2	3	4
distância cepstral	0.52	0.45	0.65	0.55
distância cepstral (dB)	2.26	1.97	2.80	2.42
taxa (bit/s)	1100	1100	850	1100

Tabela 6.18: Distância cepstral média e taxa de bits para os vocoders implementados usando-se quantização vetorial. Essas medidas foram obtidas para segmentos de voz pertencentes à seqüência de treinamento.

Empregando-se a interpolação linear dos parâmetros quantizados, conseguiu-se reduzir em 50% as taxas de bits, ocorrendo uma degradação da qualidade da voz sintetizada. A interpolação foi realizada usando-se os coeficientes parcor. Nas tabelas 6.19 e 6.20 são apresentadas as taxas de bits e as distâncias cepstrais médias para esses vocoders, usando-se respectivamente segmentos de voz independentes da seqüência de treinamento e segmentos de voz pertencentes à mesma.

vocoder	1	2	3	4
distância cepstral	0.89	0.87	1.08	0.93
distância cepstral (dB)	3.86	3.76	4.67	4.04
taxa (bit/s)	550	550	425	550

Tabela 6.19: Distância cepstral média e taxa de bits para os vocoders implementados usando-se quantização vetorial e interpolação. Essas medidas foram obtidas para segmentos de voz independentes da seqüência de treinamento.

vocoder	1	2	3	4
distância cepstral	0.75	0.72	0.87	0.79
distância cepstral (dB)	3.28	3.14	3.79	3.44
taxa (bit/s)	550	550	425	550

Tabela 6.20: Distância cepstral média e taxa de bits para os vocoders implementados usando-se quantização vetorial e interpolação. Essas medidas foram obtidas para segmentos de voz pertencentes à seqüência de treinamento.

Capítulo 7

DIFERENTES TIPOS DE EXCITAÇÃO

7.1 INTRODUÇÃO

O modelo usado para a excitação do vocoder LPC é o principal responsável pela qualidade não natural da voz sintetizada pelo mesmo. Esse modelo envolve uma decisão sonoro/não sonoro e uma detecção de pitch, não sendo válido para todos os tipos de sons.

Para melhorar a qualidade da voz sintetizada, tornando-a mais natural, outros tipos de excitação têm sido propostos. Algumas dessas excitações utilizam o sinal de voz ou o sinal de resíduo, transmitindo apenas as componentes de baixa freqüência. Esses modelos também utilizam os coeficientes LPC para sintetizarem voz. A principal vantagem desses modelos é a não utilização do detector de pitch e da decisão sonoro/não sonoro.

A seguir serão apresentados alguns codificadores de voz com diferentes tipos de excitação:

7.2 CODIFICADORES COM DIFERENTES TIPOS DE EXCITAÇÃO

7.2.1 ' Residual Excited Linear Predictive Vocoder ' (RELP)

Esse modelo baseia-se no fato que as componentes de baixa freqüência do sinal de voz são as mais importantes. O sinal de resíduo reconstruído é usado como excitação para o sintetizador.

Nesse modelo, o sinal de resíduo é filtrado por um filtro passa-baixas, sendo a seguir dizimado. O filtro passa-baixas usualmente apresenta freqüência de corte entre 800 e 1000 Hz. O fator de dizimação depende da quantidade de ' aliasing ' que é permitida. Dessa forma, são transmitidos os coeficientes LPC e as componentes de baixa freqüência do sinal de resíduo.

No sintetizador ocorre a interpolação das componentes de baixa freqüência transmitidas, para a recuperação da banda base (conteúdo de baixa freqüência) do sinal residual. As componentes de alta freqüência são reconstruídas a partir da banda base do resíduo. A soma das componentes de alta e baixa freqüência do resíduo constitui o sinal de excitação para o filtro de síntese. A figura 7.1 mostra o diagrama em blocos do RELP [5].

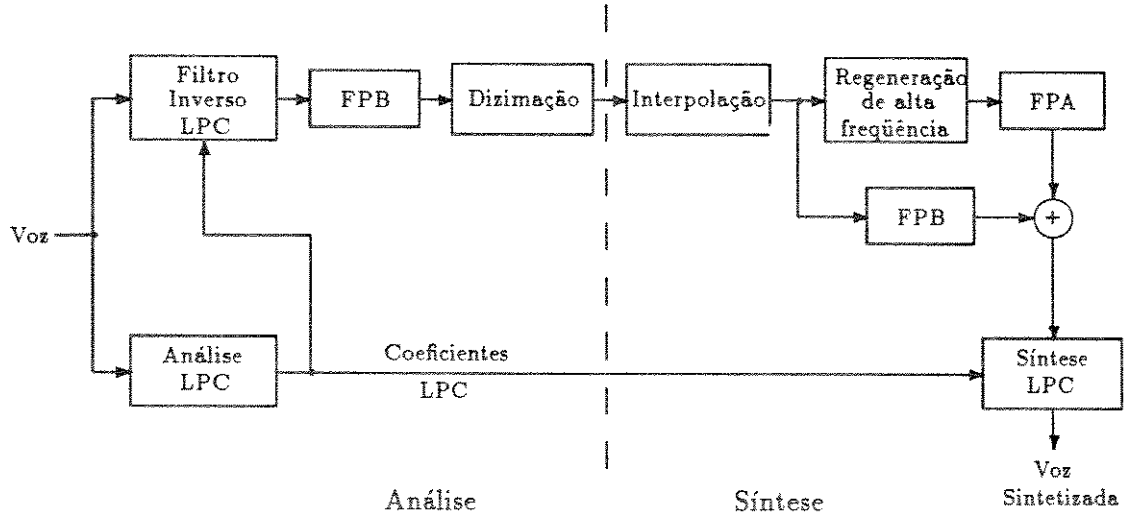


Figura 7.1: Diagrama em blocos do RELP.

Reconstrução das altas freqüências

As componentes de alta freqüência podem se recuperadas usando-se métodos simples como a retificação ou métodos mais complexos como a translação espectral ou o dobramento espectral.

A translação espectral envolve o deslocamento da banda base (limitada em $B\ Hz$) em múltiplos de B. Para um fator de dizimação igual a N, $N-1$ cópias da banda base são necessárias.

O dobramento espectral é similar à translação, ocorrendo uma inversão de toda segunda cópia da banda base. O dobramento espectral é obtido automaticamente no processo de interpolação, após a inserção de $N-1$ zeros depois de cada amostra.

A figura 7.2 apresenta um esquema para a obtenção do sinal de excitação do RELP, mostrando a reconstrução das componentes de alta freqüência do sinal residual [5].

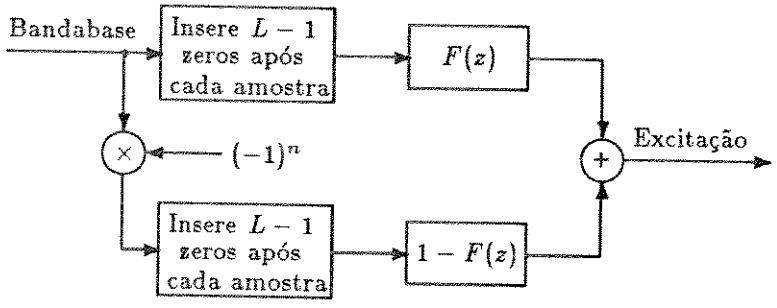


Figura 7.2: Esquema para a obtenção do sinal de excitação do RELP.

Devido ao fato do sinal de resíduo regenerado no sintetizador não ser idêntico ao sinal de resíduo original, a voz sintetizada apresenta aspectos de ' rouquidão '.

7.2.2 ' Voice Excited Linear Predictive Vocoder ' (VELP)

Esse sistema difere do RELP, pelo fato de transmitir como excitação a banda base do próprio sinal de voz, ao invés da banda base do resíduo. O sinal de excitação desse modelo não apresenta um espectro plano como o RELP, tornando-se necessária a utilização de uma transformação para tornar o espectro plano. A figura 7.3 apresenta o diagrama em blocos do VELP [1].

7.2.3 ' Multipulse Excited Linear Predictive Coding ' (MPE)

Nesse sistema, o filtro de síntese LPC é excitado por pulsos múltiplos, independente do sinal ser sonoro ou não sonoro. Esses pulsos tentam aproximar o sinal de resíduo, sendo que grandes excursões desse sinal são modeladas por grandes pulsos e pequenas excursões por pequenos pulsos. O principal problema desse sistema é a determinação das amplitudes e localizações ótimas dos pulsos. Na figura 7.4 é mostrado o diagrama em blocos do MPE [5].

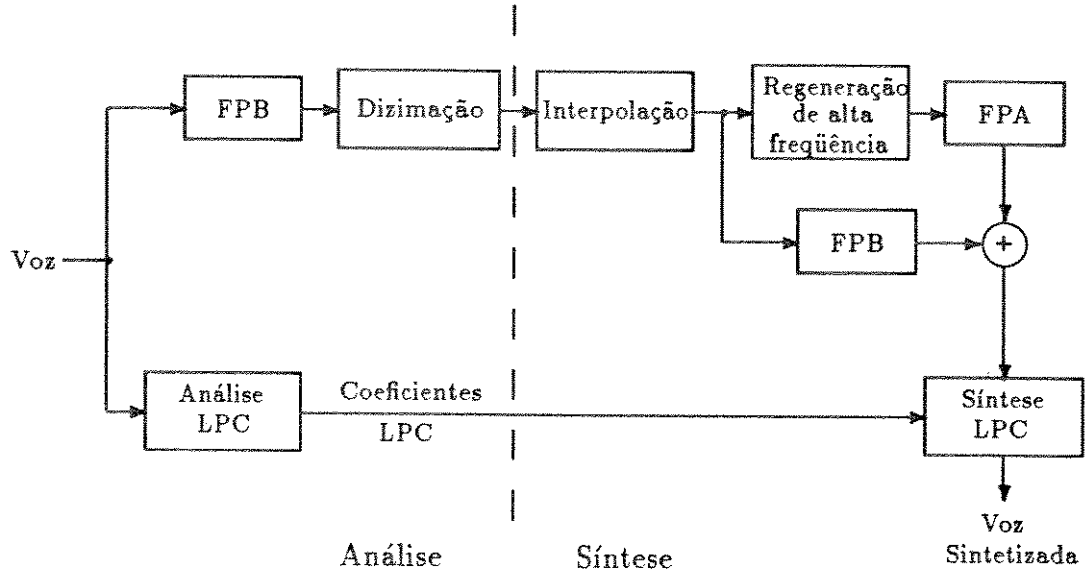


Figura 7.3: Diagrama em blocos do VELP.

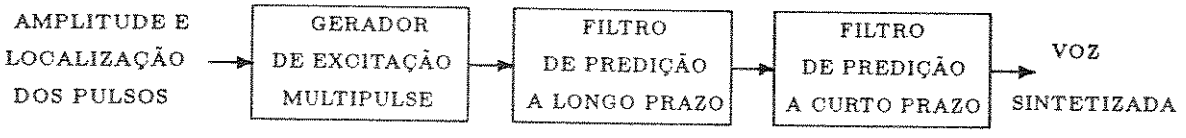


Figura 7.4: Diagrama em blocos do sintetizador do MPE.

Como o RELP, esse sistema não necessita de um detector de pitch e decisão sonoro/não sonoro. Os parâmetros transmitidos são os parâmetros dos filtros e as amplitudes e as localizações dos pulsos múltiplos. Os coeficientes dos filtros, normalmente são calculados a cada 20 ms, e o sinal de excitação é atualizado a cada 5 ou 10 ms [1].

A amplitude e localização de cada pulso são determinadas de modo a minimizar o erro quadrático médio ponderado entre a voz original e a voz sintetizada. A figura 7.5 apresenta o diagrama em blocos de um procedimento para determinação da amplitude e localização ótimas dos pulsos [1].

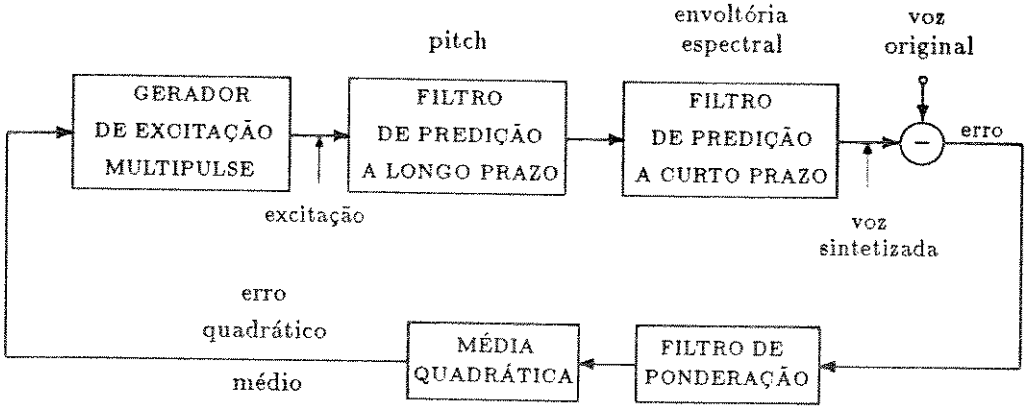


Figura 7.5: Diagrama em blocos do procedimento para a determinção da amplitude e localização ótimas dos pulsos [1].

Nesse procedimento a diferença entre a voz sintetizada utilizando a excitação ' multipulse ' e a voz original, é calculada iterativamente até tornar-se menor que um limiar pré-determinado.

O filtro de ponderação, o qual de-enfatiza as regiões dos formantes é dado por [6]:

$$W(z) = \frac{1 - \sum_{i=1}^{P} a_i z^{-i}}{1 - \sum_{i=1}^{P} \gamma^i a_i z^{-i}} \tag{7.1}$$

onde γ é uma constante entre 0 e 1 que controla a forma do ruído. Um valor típico para γ é 0.8 [5].

Determina-se a amplitude e posição dos pulsos maximizando-se a equação [1]:

$$g_k = \max \left| \frac{\Phi_{hs}(m) - \sum_{i=1}^{K-1} g_i R_{hh}(|m_i - m|)}{R_{hh}(0)} \right| \quad 1 \leq m \leq N \tag{7.2}$$

onde: N é o comprimento do quadro

g_i é a amplitude do i-ésimo pulso

m_i é a posição do i-ésimo pulso

K é o número de pulsos usados para sintetizar a voz

$R_{hh}(m)$ é a função de autocorrelação da resposta impulsiva do filtro de síntese

$\Phi_{hs}(m)$ é a correlação cruzada entre o sinal de voz original e a resposta impulsiva do filtro

Resultados experimentais mostram que pode-se obter voz sintetizada de boa qualidade usando uma taxa de 9.6 $Kbit/s$ [1].

7.2.4 ' Coded Excited Linear Predictive Coding ' (CELP)

Nesse método, o sinal de resíduo é quantizado vetorialmente por uma seqüência de pulsos aleatórios. O sinal de resíduo é produzido pela predição de longo prazo baseada na periodicidade do sinal e na predição de curto prazo baseada na correlação entre amostras adjacentes. Esse sistema é igual ao MPE, substituindo-se os pulsos múltiplos pela seqüência de pulsos aleatórios. Essa sequência de pulsos aleatórios irá excitar o filtro de síntese para produzir a voz. A seleção de cada seqüência de pulsos é feita de modo a minimizar o erro quadrático médio ponderado entre a voz sintetizada e a voz original. A figura 7.6 mostra o diagrama em blocos para a escolha da sequência de pulsos [1].

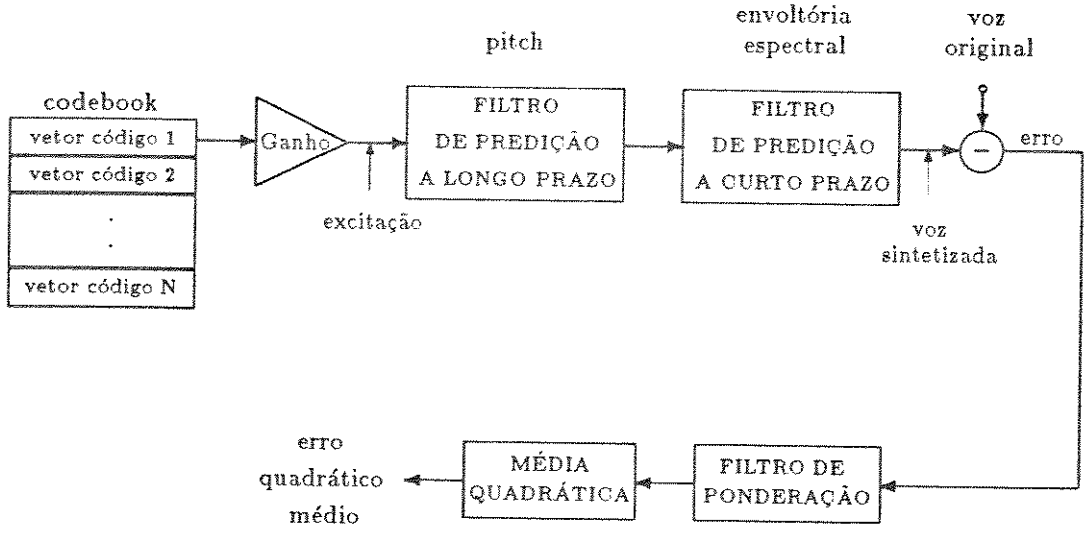


Figura 7.6: Diagrama em blocos para a escolha da sequência de pulsos para o CELP.

7.3 RELP COM QUANTIZAÇÃO VETORIAL

Implementou-se um RELP usando-se a quantização vetorial para quantizar os coeficientes LPC e o sinal de excitação. Usou-se um fator de dizimação igual a 4. Para filtrar o sinal de resíduo e obter a banda base do mesmo utilizou-se um filtro FIR passa-baixas com 100 ' taps ' e freqüência de corte igual a 900 Hz.

Para reconstruir as componentes de alta freqüência utilizou-se o esquema apresentado na figura 7.2. O filtro passa-baixas ($F(z)$) utilizado nesse procedimento foi um filtro FIR com 100 ' taps ' e freqüência de corte igual a 900 Hz.

Um novo sinal de excitação foi calculado a cada 5 ms e os coeficientes LPC foram atualizados a cada 20 ms.

Quantização Vetorial da Excitação

Para quantizar a excitação foi gerado um ' codebook ' com 1024 vetores código, usando o erro quadrático como medida de distorção. Foi utilizada a quantização por busca exaustiva , como descrita no capítulo anterior.

O ' codebook ' inicial foi gerado usando-se a técnica ' splitting ' com limiar para perturbação dos vetores código igual a 1% e limiar para otimizar os vetores código também igual a 1%. O algoritmo para gerar o ' codebook ' utilizou um limiar igual a 0.01% para ser finalizado.

Quantização Vetorial dos Coeficientes LPC

Utilizou-se a medida de distorção de Itakura Saito modificada para gerar um ' codebook ' com 1024 palavras, como descrito no capítulo anterior.

Resultados Obtidos

Quantizando-se vetorialmente a excitação e os coeficientes LPC foi obtido um RELP operando à taxa de 2500 bit/s. Utilizando-se a quantização escalar para quantizar os coeficientes LPC com 32 bits, e quantizando-se a excitação vetorialmente, obteve-se um RELP à taxa de 3600 bit/s.

Nas tabelas 7.1 e 7.2 são apresentadas as distâncias cepstrais para cada um dos codificadores implementados (RELP). Essas medidas foram obtidas utilizando-se segmentos de voz independentes da seqüência de treinamento (tabela 7.1) e segmentos de voz pertencentes à seqüência de treinamento (tabela7.2).

RELP	taxa (bit/s)	distância cepstral	distância cepstral (dB)
1	3600	1.43	6.22
2	2500	1.58	6.85

Tabela 7.1: Distância cepstral média obtida utilizando-se segmentos de voz independentes da seqüência de treinamento.

RELP	taxa (bit/s)	distância cepstral	distância cepstral (dB)
1	3600	1.02	4.43
2	2500	1.10	4.76

Tabela 7.2: Distância cepstral média obtida utilizando-se segmentos de voz pertencentes à seqüência de treinamento.

Os resultados obtidos nos testes subjetivos mostram que a voz sintetizada utilizando-se o RELP é mais natural que a voz sintetizada pelo vocoder LPC, não apresentando características ' metálicas '. Porém, a voz sintetizada pelo RELP é mais ruidosa, quando comparada com a voz sintetizada pelo vocoder LPC, utilizando-se a mesma taxa de bits. Quando utiliza-se parâmetros não quantizados, o RELP apresenta um desempenho muito bom, sendo pequenas as degradações em relação ao sinal de voz original. Assim, aumentando-se a taxa de bits do RELP, este apresenta um melhor desempenho.

Capítulo 8

CONCLUSÕES

O vocoder LPC apresenta um bom desempenho em baixas taxas (menores que 2500 bit/s), possibilitando a obtenção de voz sintetizada com alta inteligibilidade.

A utilização de um detector de pitch eficiente e preciso é muito importante para o bom desempenho do vocoder LPC.

Quantizando-se escalarmente os coeficientes razão log-área, pode-se obter voz sintetizada com alta qualidade à taxa de 2200 bit/s e com baixos custos computacionais. Com uma degradação muito pequena na qualidade da voz sintetizada, essa taxa pode ser reduzida para 1850 bit/s, diminuindo-se o número de bits utilizados para quantizar os coeficientes razão log-área.

Com o uso da interpolação linear dos parâmetros quantizados, pode-se reduzir em 50% a taxa de bits, com o custo de uma pequena degradação da qualidade da voz sintetizada, mas mantendo-se a inteligibilidade da mesma.

Uma maior redução na taxa de bits pode ser alcançada empregando-se a quantização vetorial. Utilizando-se um quantizador com 1024 níveis com procura por busca exaustiva e medida de distorção de Itakura Saito modificada , obtém-se um vocoder à taxa de 1100 bit/s com um bom desempenho. Quantizando-se o ganho LPC juntamente com os coeficientes LPC, com o uso da medida de distorção de Itakura Saito, consegue-se vocoders à taxa de 850 bit/s, com uma pequena degradação na voz sintetizada. Bons resultados também são conseguidos utilizando-se o ' product code gain shape ', à taxa de 1100 bit/s. O desempenho dos ' codebooks ' gerados usando-se a medida de distorção de Itakura Saito não é muito superior ao desempenho dos ' codebooks ' que usam o erro quadrático como

medida de distorção, embora os primeiros apresentem um custo computacional muito maior. O tempo de cpu em um Vax 8800 da Digital utilizado para gerar esses ' codebooks ' é da ordem de dezenas de horas. Esse tempo pode ser bastante reduzido, utilizando-se algoritmos com busca por árvore, mas com degradação da voz sintetizada. As taxas de bits podem ser reduzidas para 550 e 425 bit/s com a utilização da interpolação linear, ocorrendo uma degradação na qualidade da voz sintetizada, mas a inteligibilidade da mesma é mantida.

Em taxas de bits em torno de 2500 bit/s, o desempenho do vocoder LPC é superior ao RELP. À medida que a taxa de bits aumenta o desempenho do RELP torna-se superior ao desempenho do vocoder LPC.

O vocoder LPC surge como uma boa opção para aplicações onde necessita-se economizar memória e onde não é necessário o reconhecimento do locutor, mas apenas a inteligibilidade da mensagem.

Apêndice I

MEDIDAS PARA A AVALIAÇÃO OBJETIVA

Para avaliar a quantização de cada parâmetro do vocoder LPC, foram utilizadas as seguintes medidas:

I.1 RELAÇÃO SINAL-RUÍDO DE QUANTIZAÇÃO

A relação sinal-ruído é definida por [14]:

$$SNR = 10 \log_{10} \frac{E(\|x\|^2)}{E(d(x,\hat{x}))} \tag{I.1}$$

$$E(\|x\|^2) = \frac{1}{K} \sum_{i=1}^{K} x_i^2 \tag{I.2}$$

$$E(d(x,\hat{x})) = \frac{1}{K} \sum_{i=1}^{K} (x_i - \hat{x}_i)^2 \tag{I.3}$$

onde: x_i é o parâmetro não quantizado
$\hat{x}_i$ é o parâmetro quantizado

Grandes valores de SNR correspondem a pequenas distorções. Essa medida foi utilizada para a avaliação da quantização do ganho e do período de pitch.

I.2 DISTÂNCIA CEPSTRAL

A distância cepstral é a distância entre as envoltórias dos espectros representados pelos coeficientes cepstrais. Essa distância é definida por [17]:

$$d_2^2 = \int_{-\pi}^{\pi} \left| \ln\left(\frac{G^2}{|A(e^{j\theta})|^2}\right) - \ln\left(\frac{G^{q\,2}}{|A^q(e^{j\theta})|^2}\right) \right|^2 \frac{d\theta}{2\pi} \tag{I.4}$$

onde: $G/A(e^{j\theta})$ é a resposta em freqüência do filtro de síntese com parâmetros não quantizados

$G^q/A^q(e^{j\theta})$ é a resposta em freqüência do filtro de síntese com parâmetros quantizados

Com a utilização da série de Taylor para expandir $\ln(G^2/|A(e^{j\theta})|^2)$, obtém-se:

$$\ln\left(\frac{G^2}{|A(e^{j\theta})|^2}\right) = \sum_{k=-\infty}^{\infty} c_k e^{-jk\theta} \tag{I.5}$$

onde: c_k são os coeficientes cepstrais.

A partir da aplicação do Teorema de Parseval e a utilização da equação I.5, a equação I.4 pode ser escrita da seguinte forma:

$$d_2^2 = \sum_{i=-\infty}^{+\infty} (c_i - c_i^q)^2 \tag{I.6}$$

onde: c_i são os coeficientes cepstrais do filtro não quantizado

c_i^q são os coeficientes cepstrais do filtro quantizado

Os coeficientes cepstrais podem ser facilmente obtidos a partir dos coeficientes LPC usando-se as equações 5.6 e 5.7, sendo esta uma das razões do grande uso desta distância.

A somatória da equação I.6 é normalmente truncada em $n = n_0$, onde n_0 deve ser maior ou igual a ordem P do preditor. Assumindo $n_0 = P$, a equação I.6

pode ser escrita como:

$$L^2 = (c_0 - c_0^q)^2 + 2\sum_{i=1}^{P}(c_i - c_i^q)^2 \tag{I.7}$$

onde: $c_0 = \ln(G^2)$

G é o ganho do filtro de síntese

A distância cepstral L pode ser interpretada como a distância *rms* entre os logaritmos dos espectros após cada um deles ter sido cepstralmente suavizado para P coeficientes.

A distância cepstral L pode ser expressa em dB usando o fator de multiplicação $10/\ln 10$.

$$L(\text{dB}) = \frac{10}{\ln 10} L \tag{I.8}$$

Para se calcular a distância cepstral, quando apenas os coeficientes LPC foram quantizados, usou-se $c_0 = c_0^q$.

Apêndice II

AMBIENTE DE TRABALHO

As simulações realizadas neste trabalho foram executadas em tempo não real no computador VAX 8800 da DIGITAL, tendo com ambiente o sistema VMS versão 5.3 – 1.

Os programas desenvolvidos para a simulação dos vocoders foram escritos em linguagem de alto nível FORTRAN 77.

Os arquivos de voz utilizados para as simulações foram gravados usando o sistema de aquisição de dados do CPqD-TELEBRÁS. Esse sistema utiliza a placa DSP-16 Data Aquisition Processor, fabricada pela Ariel. Esta placa está acoplada a um microcomputador PC-AT, apresentando as seguintes configurações para aquisição de dados:

1. Os arquivos de voz sob a forma analógica são limitados em faixa a 7000 Hz , amostrados a 16000 Hz e quantizados com 16 bits por amostra.

2. Os arquivos de voz sob a forma analógica são limitados em faixa a 3400 Hz, amostrados a 8000 Hz e também quantizados com 16 bits por amostra.

Um conversor A/D de 16 bits permite uma excursão dos sinais no intervalo -32768 a 32767 unidades.

Na simulação do vocoder LPC usou-se arquivos de voz limitados em faixa a 3400 Hz e amostrados em 8000 Hz. Esses arquivos eram constituídos por vozes feminas e masculinas. Procurou-se utilizar sentenças foneticamente equilibradas, isto é, constituidas por sons fricativos, explosivos e sonoros.

Para a geração dos gráficos dos sinais de voz, foi utilizado o ' software ' ILS (Iterative Laboratory System), desenvolvido pela Signal Technology Incorporation.

Referências Bibliográficas

[1] Furui, S., " Digital Speech Processing, Synthesis and Recognition ", Marcel Dekker, Inc., 1989.

[2] Rabiner, L. R. e Schafer, R. W., " Digital Processing of Speech Signal ", Prentice Hall, Inc., 1978.

[3] Cegalla,D. P., " Novíssima Gramática da Língua Portuguesa " , Companhia Editora Nacional, 1984.

[4] Rocha Lima, C. H., " Gramática Normativa da Língua Portuguesa ", Livraria José Olympio Editora, 1982.

[5] O' Shaughnessy, D. , " Speech Communication: Human and Machine ", Addison-Wesley Publishing Company, 1987.

[6] Markel, J. D., Gray Jr., A. H., " Linear Prediction of Speech ", Springer-Verlag, 1982.

[7] Un, C. K., Yang, S. C., " A Pitch Extraction Algorithm Based on LPC Inverse Filtering and AMDF ", IEEE Transaction on Acoustics, Speech, and Signal Processing, vol. ASSP-25, nº 6, pag: 565-572, Dezembro de 1977.

[8] Schäfer-Vincent, Kurt, " Pitch Period Detection and Chaining: Method and Evaluation ", Phonetica 40, pag: 177-202, 1983.

[9] Makhoul, John, " Linear Prediction: A Tutorial Rewiew ", Proceedings of IEEE, vol. 63, nº 4, pag: 561-580, Abril de 1975.

[10] Viswanathan, R., Makhoul, J., " Quantization Properties of Transmission Parameters in Linear Predictive Systems ", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-23, nº 3, pag: 309- 321, Junho de 1975 .

[11] UN, C. K., Yang, S. C., " Piecewise Linear Quantization of LPC coefficents ", IEEE ICASSP, pag: 417-420, Hartford, 1977.

[12] Linde, J., Buzo, A., Gray, R. M., " An Algorithm for Vector Quantizer Design ", IEEE Transactions on Communications, vol. Com. 28, nº 1, pag: 84-95, Janeiro de 1980.

[13] Makhoul, J., Roucos, S., Gish, H., " Vector Quantization in Speech Coding ", Proceedings of the IEEE, vol. 73, no 11, pag: 1551- 1588, Novembro de 1985.

[14] Gray, R. M., " Vector Quantization ", IEEE ASSP Magazine, pag: 4-29, Abril de 1984.

[15] Sabin, M. J., Gray, R. M., " Product Code Vector Quantizers for Waveform and Voice Coding ", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-32, no 3, pag: 474-488, Junho de 1984.

[16] Buzo, A., Gray Jr., A. H., Gray, R. M., Markel, J. D., " Speech Coding Upon Vector Quantization ", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-28, no 5, pag: 562-574, Outubro de 1980.

[17] Gray Jr., A., Markel, J. D., " Distance Measures for Speech Processing ", IEEE Transactions on Acoustics, Speech and Signal Processing, pag: 380-391, Outubro de 1976.

[18] Juang, B. H., Wong, D. Y., Gray Jr., A., " Distortion Performance of Vector Quantization for LPC Voice Coding ", IEEE Transactions ons Acoustics, Speech and Signal Processing, vol. ASSP-30, no 2, pag: 294-303, Abril de 1982.

[19] Juang, B. H., Wong, D. Y., Gray Jr., A. H., " An 800 bit/s Vector Quantization LPC Vocoder ", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-30, no 5, Outubro de 1982.

[20] Ross, M. J., Shaffer, H. L., Cohen, A., Freudeberg, R., Manley, H. J., " Average Magnitude Difference Function Pitch Extractor ", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-22, no 5, pag: 353-361, Outubro de 1974.

[21] Sluyter, R. J., " Digitalization of Speech ", Philips Techincal Review, vol. 41, no 7/8, 1983/84.