UNIVERSIDADE ESTADUAL DE CAMPINAS – UNICAMP
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO – FEEC
DEPARTAMENTO DE COMUNICAÇÕES – DECOM

OTIMIZAÇÃO DOS CODIFICADORES VSELP E EFR POR REFINAMENTO NA MODELAGEM AUTOREGRESSIVA

Autora: Irene Heleonora Sêda Pinto Fantini

Orientador: Prof. Dr. Luís Geraldo Pedroso Meloni

Banca Examinadora: Prof. Dr. Luís Geraldo Pedroso Meloni (FEEC/UNICAMP)
Prof. Dr. José Sindi Yamamoto (Univer. São Francisco)
Prof. Dr. Fábio Violaro (FEEC/UNICAMP)
Prof. Dr. Amauri Lopes (FEEC/UNICAMP)

Tese submetida à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos paraobtenção do título de **Mestre em Engenharia Elétrica.**

Campinas, setembro de 2000

Agradecimento

Agradeço ao meu orientador Prof. Dr. Luís Geraldo Meloni por seu apoio, incentivo e colaboração na elaboração do trabalho. Agradeço aos meus amigos desta caminhada: Raquel, Leonardo, Fabrício, Luís, Flávio, Ynoguchi, Rodrigo, Hélder, dentre outros. Agradeço aos Profs. Drs. Antônio Marcos e Fábio Violaro que me ajudaram pacientemente tantas vezes. Agradeço o apoio e compreensão de meu esposo, Ildeu e principalmente de minha filha Mariana. Agradeço aos meus pais pelo incentivo aos estudos desde cedo.

Agradeço, principalmente, à Deus por mais uma vitória alcançada.

Abstract

This thesis introduces an enhancement for Linear Predictive coders looking for a better performance. The refined redesign is based on the observation that a better formant estimation is obtained by the use of frame length multiple of pitch period and a frame position synchronous to glottal opening or closure.

This procedure was applied to two speech coders, the VSELP and the EFR, and performance comparisons were realized. Objective (segmental SNR) and subjective qualities were analyzed. To evaluate subjective quality, the PSQM measurement was used due to its high correlation with the MOS measurement.

The refined autorregressive modeling offers better subjective and objective qualities on both coders. Two kinds of simulations were conducted, the first was called exhaustive method and the second was called simplified method. The EFR coder had a better performance than VSELP coder. The results for the exhaustive method were better than the simplified method. For the exhaustive method, the EFR gain is up to 3.5 dB in the segmental SNR and up to 0.75 in the PSQM, when compared to the original coder. The VSELP coder gain is up to 3.2 in the segmental SNR and up to 0.97 in the PSQM for the same conditions.

Resumo

Este trabalho estuda um aprimoramento de codificadores baseados em predição linear visando melhorar seu desempenho. O refinamento é baseado na observação de que uma estimação melhor de formantes é obtida com o uso de quadros de tamanho múltiplo do período de *pitch* e com posição síncrona à abertura ou fechamento da glote.

Este procedimento foi realizado para dois codificadores de fala, o VSELP e o EFR, e foram comparados os seus desempenhos. Foram analisadas as qualidades objetiva (SNR segmentar) e subjetiva. Para fins de mensurar a qualidade subjetiva foi utilizada a medida PSQM (*Perceptual Speech Quality Measurement*), que tem forte correlação com o MOS (*Mean Opnion Score*).

A modelagem autoregressiva refinada oferece melhor qualidade para ambos os codificadores. Foram realizadas duas simulações, uma chamada de método simplificado e outra chamada de método exaustivo. O desempenho do codificador EFR é melhor que o do VSELP. Os resultados obtidos no método exaustivo foram melhores, sendo que para o EFR houve ganho de até 3,5 dB na SNR segmentar e ganho de até 0,75 na PSQM em comparação ao codificador original. Para o VSELP, o ganho foi de até 3,2 dB na SNR segmentar e ganho de até 0,97 na PSQM, para as mesmas condições.

Índice

Introdução.....1

O codificador Vector Sum Excited Linear Predictive – VSELP.....7
2.1 Introdução.....7
2.2 Pré-Processamento..... 10
2.3 Matriz de Covariância e Técnica de Suavização Espectral..... 10
2.4 Predição linear..... 11
2.4.1 Algoritmo de Covariância *lattice* de ponto fixo FLAT..... 11
2.4.2 Quantização dos Coeficientes de Reflexão..... 15
2.5 Interpolação dos Coeficientes LPC..... 16
2.6 Energia do Quadro..... 17
2.7 Filtro Perceptual W(n)..... 18
2.8 Princípios de Busca do Dicionário..... 18
2.9 Atraso de Preditor de Longo Termo, L..... 20
2.10 Busca no Dicionário..... 23
2.11 Otimização Conjunta de Ganhos..... 26
2.12 Transformação dos Ganhos em G_S, P_0 *e* P_1..... 28
2.13 Quantização Vetorial e Codificação de G_S, P_0 e P_1..... 29
2.14 Atualização do Estado do Filtro Preditor de *Pitch*..... 30
2.15 Decodificador..... 31
2.15.1 Pós-Filtro Espectral Adaptativo..... 31

O codificador *Enhanced Full Rate* – EFR..... 33
3.1 Introdução..... 33
3.2 Pré-Processamento..... 34
3.3 Predição linear..... 36
3.4 Conversão para LSF..... 37
3.5 Quantização das LSF..... 39
3.6 Interpolação dos LSPs..... 41
3.7 Filtro perceptual..... 41
3.8 Resposta ao Impulso..... 42
3.9 Cálculo do sinal alvo..... 42
3.10 Cálculo do valor de *Pitch*..... 43
3.10.1 Análise em malha aberta..... 43
3.10.2 Análise em malha fechada..... 44
3.11 Dicionário Algébrico..... 45
3.12 Quantização dos ganhos..... 47
3.13 Atualização de memória..... 48
3.14 Decodificador..... 49
3.14.1 Pós-processamento..... 51

A Medida Perceptual de Qualidade de Voz..... 54
4.1 Introdução..... 54
4.2 A medida de qualidade PSQM..... 55
4.3 Iniciação..... 57
4.3.1 Alinhamento no tempo..... 57
4.4 Mapeamento tempo-freqüência..... 58
4.5 Conversão e filtragem da escala de freqüência..... 59
4.5.1 Escalonamento Global..... 59
4.5.2 Filtragem na banda telefônica..... 60
4.5.3 Ruído *Hoth*..... 60
4.6 Conversão Não Linear da Escala de Intensidades..... 61
4.7 Modelagem Cognitiva..... 61

4.7.1 Escalonamento do Nível de Audibilidade..... 62
4.7.2 Densidade amostrada de perturbação de ruído..... 62
4.7.3 Processamento Assimétrico..... 62
4.7.4 Perturbação de ruído incluindo o processamento no intervalo de silêncio..... 63
4.8 PSQM+..... 64

Refinamento da Modelagem Autoregressiva..... 66
5.1 Introdução..... 66
5.2 Refinamento da Modelagem Autoregressiva..... 67
5.3 Covariância Modificada..... 69
5.4 Determinação do *Pitch*..... 70
5.4.1 Determinação do *Pitch*..... 71
5.5 O Classificador Sonoro/Surdo..... 74

Resultados..... 76
6.1 Medidas Utilizadas..... 76
6.2 Sinais Utilizados..... 77
6.3 Resultados obtidos para VSELP..... 78
6.3 Resultados obtidos para EFR..... 81

Conclusão..... 84

Bibliografia..... 86

Índice de Figuras

Introdução
1.1 Irrelevância e redundância 1
1.2 Diagrama em blocos do vocoder 2
1.3 Diagrama em blocos do MLPC 3
2.4 Diagrama em blocos do codificador CELP 4

O codificador Vector Sum Excited Linear Predictive – VSELP
2.1 Diagrama em blocos do codificador VSELP 8
2.2 Respostas em magnitude e em fase do filtro Chebyshev II 10
2.3 Estrutura treliça 13
2.4 Diagrama em blocos do procedimento de busca do dicionário 18
2.5 Diagrama em blocos do sintetizador ponderado 29
2.6 Diagrama em blocos do decodificador VSELP 30
2.7 Magnitude espectral para pós-filtro só-polos para diferentes valores de λ 32

O codificador *Enhanced Full Rate* – EFR
3.1 Diagrama em blocos do codificador EFR 35
3.2 Resposta em magnitude e fase do filtro de pré-processamento 36
3.3 Janela assimétrica 36
3.4 Diagrama em blocos do quantizador 41
3.5 Estrutura geradora do vetor de excitação 46
3.6 Diagrama em blocos do decodificador EFR 50
3.7 Resposta em magnitude e em fase do filtro de pós-processamento 53

A Medida Perceptual de Qualidade de Voz
4.1 Conceito da medida PSQM 55
4.2 Diagrama em blocos do algoritmo da medida PSQM 56

Refinamento da Modelagem Autoregressiva
5.1 Determinação do *pitch* inteiro 73

Resultados
6.1 Gráfico da relação MOS x PSQM 77
6.2 Resposta em magnitude e fase do filtro FIR passa-baixa 78
6.3 Variação da SNR em um quadro de análise de tamanho 39 79
6.4 Variação da SNR em um quadro de análise de tamanho 39 81

Índice de Tabelas

O codificador Vector Sum Excited Linear Predictive – VSELP
2.1 Parâmetros do codificador VSELP……..... 9
2.2 Janela binomial……..... 11
2.3 Algoritmo FLAT ….…….....… 14
2.4 Quantizador para energia do quadro.....…….. 17
2.5 Faixa de variação do *pitch* ……..... 20
2.6 Algoritmo do cálculo do atraso de *pitch*……..... 22

O codificador *Enhanced Full Rate* – EFR
3.1 Posições para os pulsos no dicionário algébrico 46
3.2 Resposta em magnitude e fase do filtro de pré-processamento..... 36
3.3 Janela assimétrica..... 36
3.4 Diagrama em blocos do quantizador..... 41
3.5 Estrutura geradora do vetor de excitação..... 46
3.6 Diagrama em blocos do decodificador EFR..... 50
3.7 Resposta em magnitude e em fase do filtro de pós-processamento..... 53

Refinamento da Modelagem Autoregressiva
5.1 Algoritmo do método exaustivo..... 68
5.2 Algoritmo para o método simplificado..... 69
5.3 Algoritmo do classificador sonoro/surdo..... 75

Resultados
6.1 Descrição dos sinais utilizados nas simulações .....…..... 79
6.2 SNRseg para quadros sonoros do codificador VSELP original, utilizando o método exaustivo e utilizando o método simplificado…….. 80
6.3 PSQM para o codificador VSELP original, o método exaustivo e o método simplificado 80
6.4 SNRseg para o codificador EFR original, o método exaustivo e o método simplificado.....…. 82
6.5 PSQM para o codificador EFR original, o método exaustivo e o método simplificado.....…… 82

Capítulo 1

Introdução

A codificação de fala é utilizada para se obter representações digitais compactas do sinal de fala com o propósito de transmissão ou armazenamento. A codificação busca extrair a parte relevante e não redundante do sinal. Na figura 1.1 temos a ilustração do conceito de redundância e relevância.

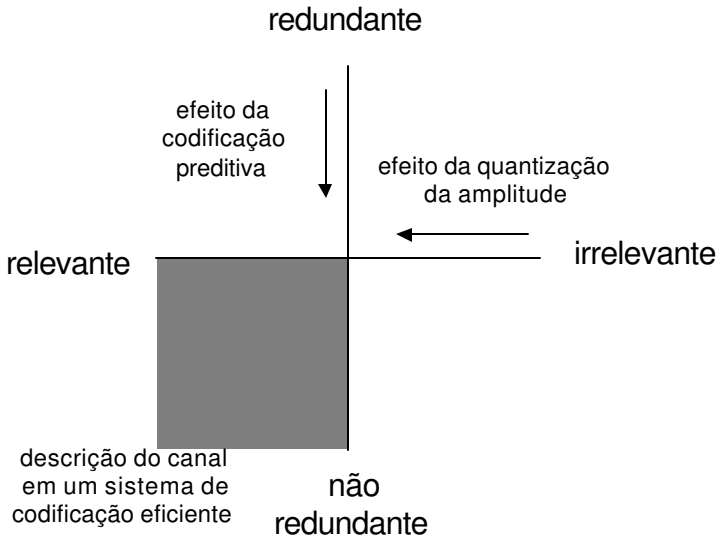

Figura 1.1 – Irrelevância e redundância.

Algumas aplicações de codificação do sinal de fala são

- transmissão do sinal em canais de banda estreita,
- comunicação segura do sinal por meio de criptografia,
- multimídia,
- telefonia celular e

- transmissão de voz sobre IP (*Internet Protocol*).

Os codificadores utilizados pela telefonia móvel são do tipo híbrido, ou seja, são extraídos e transmitidos os parâmetros do sinal de fala tais como coeficientes LPC (*Linear Prediction Coefficient*), ganho, e busca-se representar a forma de onda do sinal de excitação. O vocoder (*voice coder*) LPC é o codificador paramétrico mais simples (retira os parâmetros do sinal), baseado na predição linear do sinal para a extração da redundância do sinal fala, tendo sido desenvolvido por Fant [2]. Este codificador encontra-se ilustrado na figura 1.2. O sistema assume a excitação como sendo de dois tipos: por trem de impulsos para trechos sonoros e por ruído branco para trechos surdos. O trato vocal é modelado por um filtro de síntese.

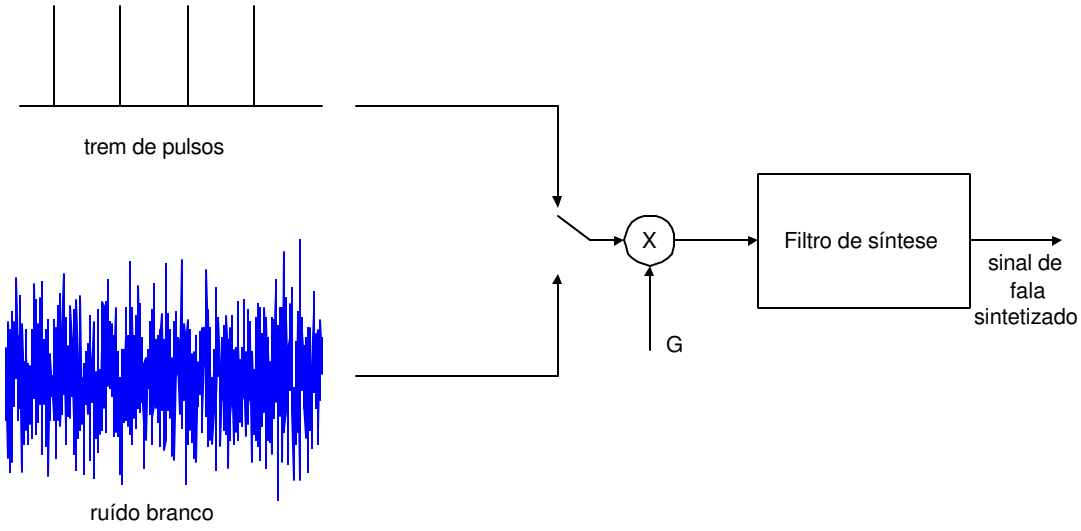

Figura 1.2 – Diagrama em blocos do vocoder.

O filtro de síntese é dado por :

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}}. \qquad (1.1)$$

O vocoder apresenta um sinal reconstruído de baixa qualidade, devido à excitação muito simplificada. Como, às vezes, os sinais de fala não são puramente sonoros ou surdos, mas mistos (com excitação sonora para baixas freqüências e surda em altas freqüências, por exemplo [3]), esta simplificação traz sérias conseqüências para a qualidade da fala.

O codificador multipulso com predição linear (MLPC) forma uma seqüência de excitação, a qual consiste de múltiplos pulsos espaçados não uniformemente. Ele está

representado na figura 1.3. Durante a análise, tanto a amplitude quanto a posição dos pulsos são determinadas, seqüencialmente, um pulso por vez, até que o erro quadrático médio seja minimizado. Usualmente empregam-se alguns pulsos por período de pitch para oferecer uma boa qualidade do sinal codificado. Neste codificador é acrescentado o filtro perceptual $W(z)$. O papel deste filtro é o de permitir maior energia do erro nas regiões de formantes. Este princípio baseia-se no fato de que nas regiões de formantes, o ruído de codificação pode ser mascarado pela fala [4]. O filtro $W(z)$ é dado por:

$$W(z) = \frac{1 - \sum_{i=1}^{p} a_i z^{-i}}{1 - \sum_{i=1}^{p} a_i \lambda^i z^{-i}}, \tag{1.2}$$

onde λ é o fator de ponderação dos coeficientes autoregressivos.

Este codificador apresenta um bom desempenho, porém exige uma taxa de bits elevada para a representação da excitação multipulso.

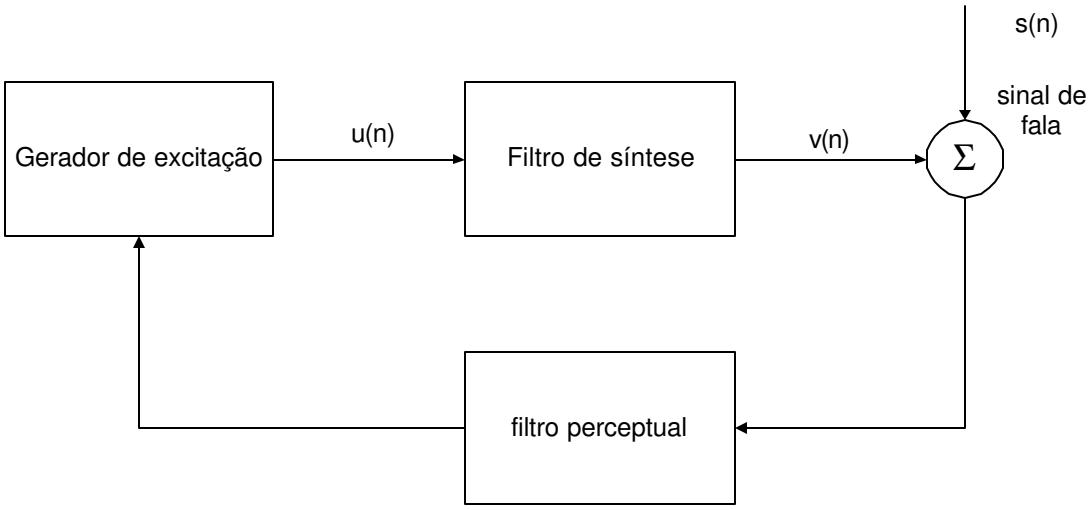

Figura 1.3 – Diagrama em blocos do MLPC.

Codificadores *Code Excited Linear Prediction* (CELP) [5] fornecem sinais de alta qualidade. Eles têm como base a predição linear e a excitação é quantizada vetorialmente e buscada em dicionários por meio de um procedimento de análise por síntese. Utilizam-se dois dicionários que somados formam o sinal de excitação. Na Figura 1.4 tem-se a ilustração do diagrama em blocos do codificador CELP.

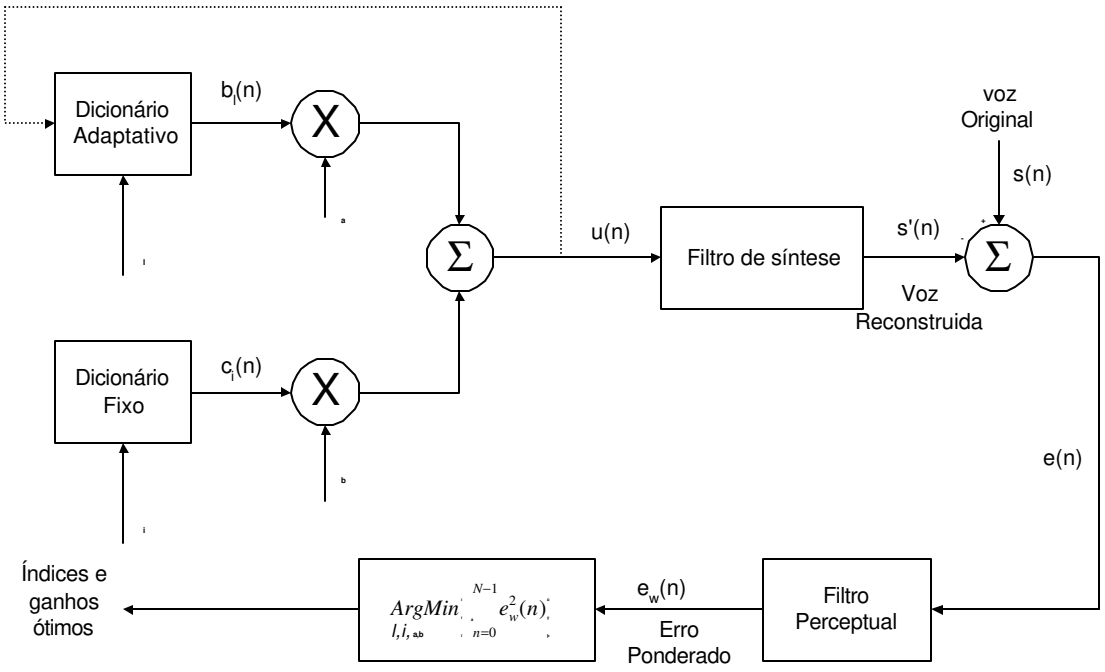

Figura 1.4 Diagrama em blocos do codificador CELP.

Os dois dicionários têm suas contribuições somadas para produzir o sinal de excitação $u(n)$. O dicionário adaptativo tem a função de um preditor de *pitch*, ou seja, reconstrói a periodicidade dos segmentos sonoros do sinal de voz. Por esta razão, é também chamado de preditor de longo termo. O dicionário fixo reproduz o sinal de resíduo dos filtros de predição linear.

O sinal de voz amostrado é segmentado em blocos de curta duração, da ordem de 20 ms, para realizar a análise LPC ou modelagem autoregressiva. Esta modelagem pode ser realizada por dois métodos: o da autocorrelação e o da covariância. O método da covariância é normalmente empregado sem utilizar janelamento e permite trabalhar com blocos menores do que o método da autocorrelação. No método da autocorrelação, uma janela de ponderação é aplicada para determinar o quadro (bloco) de sinal a ser analisado e para realizar uma transição gradual nas bordas. A duração da janela empregada é maior que a do quadro, ocorrendo então superposição entre as janelas adjacentes.

Usualmente na família de codificadores CELP, o quadro é subdividido em quatro subquadros. A busca dos índices dos dicionários adaptativo e fixo (l,i), bem como de seus respectivos ganhos (α, β), é feita na cadência de subquadros. Após a análise de cada

quadro, o sinal $u(n)$ é armazenado em memória para utilização no próximo quadro. O filtro perceptual é da forma da equação (1.2).

Em decodificadores da família CELP, muitas vezes, utiliza-se de um pós-filtro adaptativo que visa melhorar as imperfeições esporádicas do codificador, como por exemplo, uma oscilação exagerada causada por uma imprecisão na estimativa de formantes. Este pós-filtro também pode empregar um filtro de curto termo e um filtro de longo termo. O pós-filtro de longo termo utiliza o valor estimado e decodificado do *pitch*. O pós-filtro de curto termo enfatiza as estruturas dos formantes da fala e compensa globalmente inclinações espectrais.

Neste trabalho veremos dois codificadores pertencentes à família CELP. São eles o *Vector-Sum Excited Linear Predictive* (VSELP), IS 136 [6], e o *Enhanced Full Rate* (EFR), IS641 [7]. Ambos são padronizados pela *Telecommunications Industry Association* (TIA).

O VSELP faz a análise LPC pelo método da covariância e transmite os coeficientes de reflexão. É formado por dois dicionários fixos e um adaptativo, que combinados geram o sinal de excitação. As particularidades deste codificador estão descritas no capítulo 2.

O EFR é um codificador mais recente, padronizado em 1996. A análise LPC neste caso é feita pelo método da autocorrelação, e os parâmetros transmitidos são as *Line Spectral Frequencies* (LSF). A predição de *pitch* é mais refinada e o codificador utiliza a técnica *Algebraic Code Excited Linear Prediction* (ACELP) para codificar o sinal de excitação. As particularidades deste codificador estão descritas no capítulo 3.

Neste trabalho é estudada a melhoria da qualidade dos codificadores através de um refinamento na modelagem autoregressiva. É conhecido que a modelagem da fala usando o tamanho do quadro igual ou múltiplo do período de *pitch* e sua posição síncrona com a abertura ou fechamento da glote melhora a modelagem autoregressiva no sentido de uma melhor estimativa dos formantes [8][9]. Esta técnica está descrita no capítulo 5.

Para medir a qualidade do sinal, temos métodos avaliadores de qualidade objetiva e subjetiva. O método subjetivo mais utilizado é *Mean Opinion Score* (MOS), o qual qualifica o sinal através de testes por grupos de ouvintes. O método objetivo é uma análise numérica, normalmente a relação sinal-ruído (SNR). Uma boa qualidade objetiva não representa necessariamente uma boa qualidade subjetiva. Como é difícil realizar medidas subjetivas, pois envolvem um grande número de ouvintes avaliadores, a *International*

Telecommunication Union (ITU) recomenda uma medida computacional que tenta analisar o sinal subjetivamente e tem resultados próximos aos do MOS [10]. Esta medida, chamada de *Perceptual Speech Quality Measure* (PSQM), está descrita no capítulo 4.

No capítulo 6, temos os resultados das simulações com as comparações para as medidas objetivas e subjetivas. Neste capítulo, podemos verificar que o comportamento dos dois codificadores perante o refinamento ocorre de maneira parecida, e comprovar a melhoria na qualidade do sinal de fala codificado.

A análise conclusiva do trabalho, com referência ao seu desempenho, está descrita no capítulo 7.

Infelizmente, as especificações dos codificadores não apresentam em detalhes todas as passagens matemáticas. Nos capítulos que descrevem os mesmos, buscou-se expor seus princípios de funcionamento de forma mais didática.

Capítulo 2

O codificador *Vector Sum Excited Linear Predictive* - VSELP

Neste capítulo é revisado o codificador de fala VSELP padronizado pela *Telecommunications Industry Association* (TIA/EIA) para telefonia celular TDMA, padrão IS 136 [6].

2.1 Introdução

O VSELP é um codificador de fala da família CELP. O diagrama de blocos do codificador VSELP está representado na figura 2.1. O VSELP representa o sinal de excitação através de dois dicionários e há um terceiro dicionário para o preditor de *pitch*. O método de busca do vetor ótimo nos dicionários se dá de forma eficiente e a memória utilizada para armazenar os vetores de base é reduzida.

O sinal de fala na entrada pode ser analógico ou digital PCM de 8 bits. O sinal analógico é amostrado a 8 kHz e codificado linearmente com 16 bits. O sinal digital é filtrado por um filtro passa-altas Chebyshev do tipo II para eliminar as baixas freqüências que causam ruídos. Então se calcula a matriz de covariância do sinal para a análise LPC.

Antes de se realizar a análise LPC, é utilizada a técnica de suavizamento espectral (SST). A energia do quadro é calculada a partir da matriz de covariância (anterior ao suavizamento espectral). A cada quadro, calcula-se o filtro preditor linear $P(z)$ pelo método da covariância, através do algoritmo FLAT (*Fixed-Point Covariance Lattice*). O algoritmo (FLAT) fornece os coeficientes de reflexão quantizados $\{r_i\}$ do filtro preditor $P(z)$. Os coeficientes LPC são interpolados para a análise por subquadros.

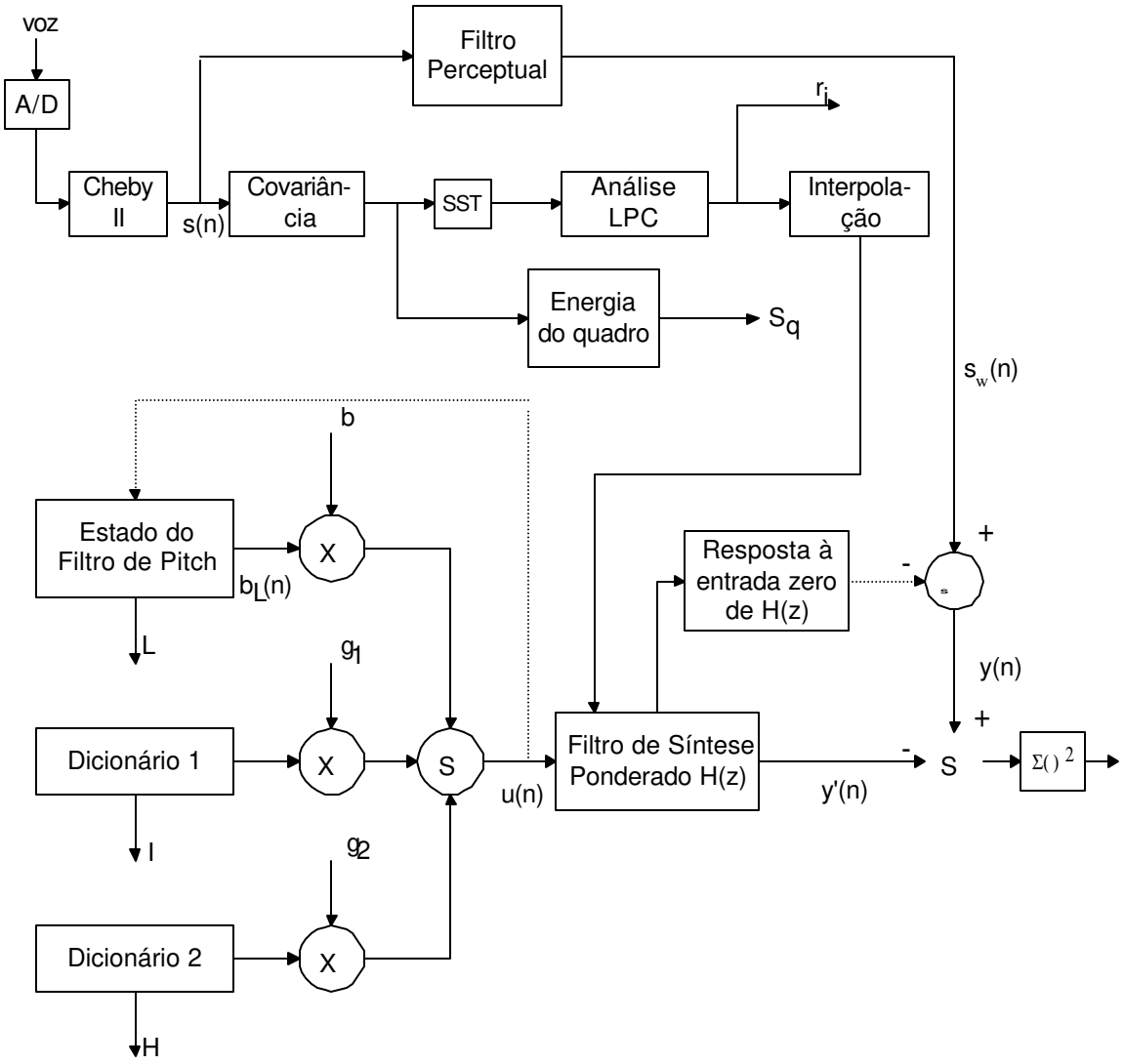

Figura 2.1: Diagrama em blocos do codificador VSELP.

A cada subquadro, calcula-se o período de *pitch* e o sinal de excitação. O período de *pitch* é buscado no dicionário de longo termo, observando-se a excitação passada (quadro anterior) e a atual. O VSELP utiliza dois dicionários, os quais contêm apenas M vetores

base. Os dicionários de 2^M vetores-código são construídos com os M vetores base. O sinal de excitação provém destes três dicionários.

O sinal de excitação é filtrado pelo filtro de síntese ponderado para a recomposição do sinal de fala. O filtro de síntese ponderado $H(z)$ é dado por:

$$H(z) = \frac{1}{1 - \sum_{i=1}^{N_P} a_i \lambda^i z^{-i}} \tag{2.1}$$

Este filtro corresponde à cascata dos filtros de síntese e perceptual, como pode ser observado na Figura 1.4.

O filtro perceptual $W(z)$ explora as características de audição do ouvido humano visando melhorar a qualidade do sinal codificado.

Alguns parâmetros básicos são utilizados em todo o Capítulo 2 e estão sumarizados na Tabela 2.1.

F_s	taxa de amostragem	8 kHz
N	tamanho do subquadro	40 amostras (5 ms)
N_F	tamanho do quadro	160 amostras (20 ms)
N_P	ordem do preditor de curto termo	10
N_A	intervalo de análise do FLAT : $N_F + N_P$	170 amostras
L	o atraso do preditor de longo termo (*pitch*)	20-146 amostras 400 – 50 Hz
M	número de vetores base dos dicionários 1 e 2	7
λ	parâmetro do filtro perceptual	0,8

Tabela 2.1 : Parâmetros do codificador VSELP

2.2 Pré-Processamento

O sinal de fala é filtrado por um filtro passa-altas para eliminar freqüências indesejáveis, como a de 60 Hz da rede e a componente DC do sinal. É utilizado um filtro passa-altas

Chebyshev tipo II de quarta ordem. Os coeficientes do filtro estão definidos no padrão do codificador [6]. A freqüência de corte de -3 dB deste filtro é de 120 Hz e em 60 Hz a atenuação é de –40 dB. A figura 2.2 mostra as respostas em magnitude e fase do filtro Chebyshev II.

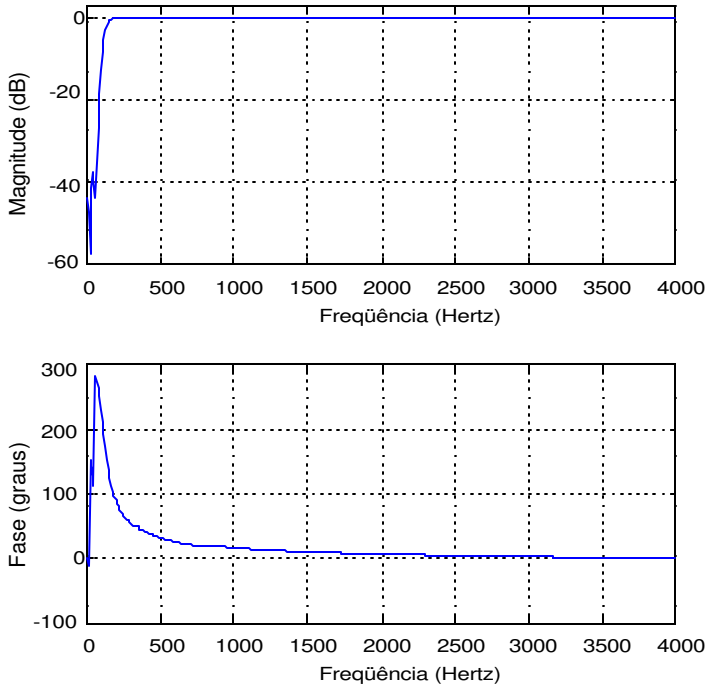

Figura 2.2 : Respostas em magnitude e em fase do filtro Chebyshev II.

2.3 Matriz de Covariância e Técnica de Suavização Espectral

Calcula-se a matriz de covariância a partir do sinal de fala de entrada, cujos elementos são:

$$\phi(i,k) = \sum_{n=N_p}^{N_A-1} s(n-i)s(n-k) \quad \text{para } 0 \le i,k \le N_p \,. \tag{2.2}$$

Esta matriz tem dimensão 11x11 para uma análise LPC de 10ª ordem.

Normalmente a análise LPC subestima a largura de banda dos formantes do sinal de fala [11]. Para compensar esta sub-estimação, utiliza-se a técnica de expansão da largura de banda dos formantes, que consiste em multiplicar as estimativas da covariância por meio de uma janela binomial. Esta janela é definida na Tabela 2.2, reproduzida do padrão do codificador [6].

w(0) = 1,0
w(1) = 0,999644
w(2) = 0,998577
w(3) = 0,996802
w(4) = 0,994321
w(5) = 0,991141
w(6) = 0,987268
w(7) = 0,982710
w(8) = 0,977478
w(9) = 0,971581
w(10) = 0,965032

Tabela 2.2: Janela binomial

2.4 Predição linear

A predição linear é realizada uma vez por quadro (160 amostras). A determinação da excitação por meio de análise por síntese é realizada a cada subquadro (40 amostras). A obtenção dos coeficientes do filtro preditor para cada subquadro é realizada por meio de interpolação a ser vista na seção 2.5.

2.4.1 Algoritmo de Covariância *lattice* de ponto fixo FLAT

Este algoritmo é baseado no algoritmo de Cumani [12], ao qual foi introduzida a quantização dos coeficientes LPC. É um algoritmo eficiente para ponto fixo e utilizado para determinar os coeficientes do filtro de curto termo. O intervalo de análise usado é de 170 amostras.

A Figura 2.3 ilustra a estrutura treliça que é utilizada para predição linear. Existem duas saídas, $f_j(n)$ e $b_j(n)$, que correspondem, respectivamente, ao erro de predição direto e reverso, sendo r_j o coeficiente treliça do *j*-ésimo estágio. Estes erros podem ser interpretados como erros residuais de predição do sinal, calculados estágio a estágio.

Os elementos das matrizes de correlação são dados por:

$$F(i,k) = \mathbf{f}_j^{\mathrm{T}}(n-i)\mathbf{f}_j(n-k), \tag{2.3a}$$

$$B(i,k) = \mathbf{b}_j^{\mathrm{T}}(n-1-i)\mathbf{b}_j(n-1-k), \tag{2.3b}$$

$$C(i,k) = \mathbf{f}_j^{\mathrm{T}}(n-i)\mathbf{b}_j(n-1-k), \tag{2.3c}$$

para $0 \le j \le N_p$ -1 e $0 \le i,k \le N_p$. Na notação acima, $\mathbf{f}_j^{\mathrm{T}}$ representa o vetor transposto de $\mathbf{f}_j$. Essas correlações serão calculadas empregando-se N_A amostras disponíveis do sinal.

Cumani considera estas matrizes como de energias residuais generalizadas (GRE). Essas GRE's são definidas como produtos internos dos vetores residuais. Elas definem matrizes de covariância dos vetores residuais em estágios sucessivos da estrutura treliça. Os vetores residuais que aparecem nestas expressões podem ser expressos como processos de filtragem inversa pelo preditor progressivo e retrógrado:

$$\mathbf{f}_j(n) = -\sum_{m=0}^{j} a_m^j \mathbf{x}(n-m) \tag{2.4a}$$

$$\mathbf{b}_j(n) = -\sum_{l=0}^{j} a_{j-l}^j \mathbf{x}(n-l) \tag{2.4b}$$

onde $a_0^j = -1$.

O erro retrógrado ainda se escreve invertendo-se a ordem da soma:

$$\mathbf{b}_j(n) = -\sum_{l=0}^{j} a_l^j \mathbf{x}(n-j+l). \tag{2.4c}$$

Substituindo as expressões (2.4a) e (2.4c) em (2.3), e tomando-se outra notação para os índices, vem

$$F_{j,i,k}(n) = \sum_{m=0}^{j}\sum_{l=0}^{j} a_m^j a_l^j \mathbf{x}^{\mathrm{T}}(n-m-i)\mathbf{x}(n-l-k), \tag{2.5a}$$

$$B_{j,i,k}(n) = \sum_{m=0}^{j}\sum_{l=0}^{j} a_m^j a_l^j \mathbf{x}^{\mathrm{T}}(n-i-1-j+m)\mathbf{x}(n-k-1-j+l), \tag{2.5b}$$

$$C_{j,i,k}(n) = \sum_{m=0}^{j}\sum_{l=0}^{j} a_m^j a_l^j \mathbf{x}^{\mathrm{T}}(n-m-i)\mathbf{x}(n-k-1-j+l). \tag{2.5c}$$

Note que os produtos internos que aparecem entre os vetores *x* podem ser interpretados como elementos da matriz de covariância do processo observado:

$$\phi_{m+i,l+k}(n) = \mathbf{x}^{\mathbf{T}}(n-m-i)\mathbf{x}(n-l-k), \tag{2.6a}$$

$$\phi_{j+1+i-m,j+1-l+k}(n) = \mathbf{x}^{\mathbf{T}}(n-i-1-j+m)\mathbf{x}(n-k-1-j+l), \tag{2.6b}$$

$$\phi_{m+i,j+1-l+k}(n) = \mathbf{x}^{\mathbf{T}}(n-m-i)\mathbf{x}(n-k-1-j+l). \tag{2.6c}$$

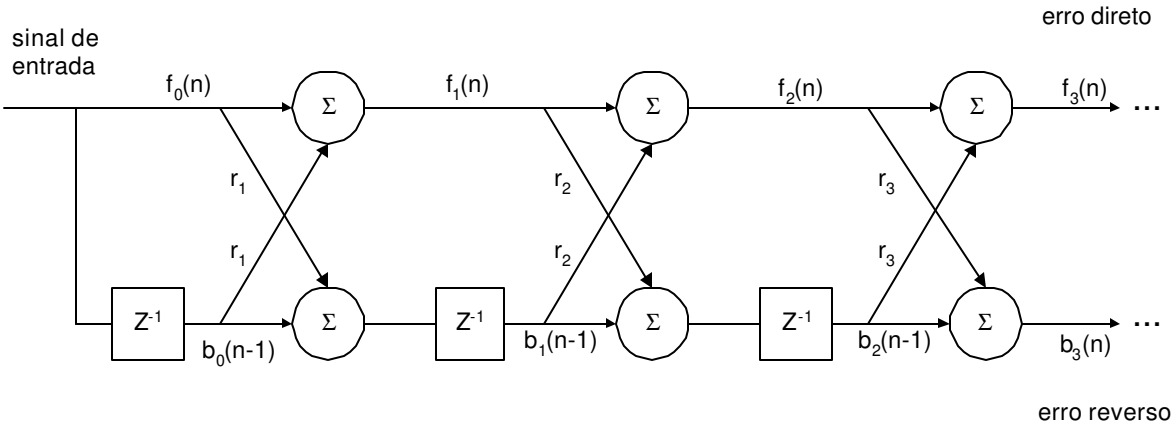

Figura 2.3: Estrutura treliça

Substituindo-se a recursão de Levinson de atualização do preditor nas expressões (2.5a-c), obtém-se um novo conjunto de recursões:

$$F_{j,i,k}(n) = F_{j-1,i,k}(n) + r_j(C_{j-1,i,k}(n) + C_{j-1,k,i}(n)) + r_j^2 B_{j-1.i.k}(n), \tag{2.7a}$$

$$B_{j,i,k}(n) = B_{j-1,i+1,k+1}(n) + r_j(C_{j-1,i+1,k+1}(n) + C_{j-1,k+1,i+1}(n)) + r_j^2 F_{j-1,i+1,k+1}(n), \tag{2.7b}$$

$$C_{j,i,k}(n) = C_{j-1,i,k+1}(n) + r_j(F_{j-1,i,k+1}(n) + R_{j-1,i,k+1}(n)) + r_j^2 B_{j-1,k+1,i}(n). \tag{2.7c}$$

Observe de (2.3a,b) que as energias residuais escalares aparecem para *i*=k=0.

As matrizes GRE no estágio zero aparecem como submatrizes da matriz de covariância do processo de entrada.

$$F_{0,i,k}(n) = \phi_{i,k}(n), \tag{2.8a}$$

$$B_{0,i,k}(n) = \phi_{i+1,k+1}(n), \tag{2.8b}$$

$$C_{0,i,k}(n) = \phi_{i,k+1}(n). \tag{2.8c}$$

Assumindo que o sinal de entrada é estacionário uma função de minimização para obter os valores ótimos dos coeficientes de reflexão pode ser estabelecida como :

$$\arg\min_{r_j} \left\{E\left[f_j^2(n)\right] + E\left[b_j^2(n-1)\right]\right\} \tag{2.9}$$

onde *E* é o operador estatístico esperança. A minimização desta função pode ser obtida

como [15][16]

$$r_j = -2\frac{E\left[f_j(n)b_j(n-1)\right]}{E\left[f_j^2(n)\right]+E\left[b_j^2(n-1)\right]} \tag{2.10}$$

A ortogonalidade do sinal e do erro assegura uma minimização estágio a estágio segundo o critério de mínimos quadrados, sem que os coeficientes prévios necessitem ser alterados. A equação acima se escreve

$$r_j = -2\frac{C_{j-1,0,0}(n)}{F_{j-1,0,0}(n)+B_{j-1,0,0}(n)}. \tag{2.11}$$

Objetivando a continuidade entre quadros subsequentes, a técnica abaixo de média é introduzida. Esta é uma das razões para se utilizar o intervalo de análise como sendo 170 amostras.

$$r_j = -2\frac{C_{j-1,0,0}(n)+C_{j-1,N_P-j,N_P-j}(n)}{F_{j-1,0,0}(n)+F_{j-1,N_P-j,N_P-j}(n)+B_{j-1,0,0}(n)+B_{j-1,N_P-j,N_P-j}(n)} \tag{2.12}$$

Na tabela 2.3 temos a descrição do algoritmo FLAT .

1. Primeiro calcule a matriz de covariância, expressa em (2.3a-c), para $0 \le i,k \le N_P$.
2. Então, calcule :

$$\begin{aligned} F_0(i,k) &= \phi(i,k), && \text{para } 0 \le i,k \le N_P - 1 \\ B_0(i,k) &= \phi(i+1,j+1), && \text{para } 0 \le i,k \le N_p - 1 \\ C_0(i,k) &= \phi(i,k+1), && \text{para } 0 \le i,k \le N_P - 1. \end{aligned}$$

3. Faça j=1.
4. Calcule r_j usando (2.12)
5. Quantize r_j usando (2.13)
6. Se $j = N_P$ então pare.
7. Calcule $F_{j,i,k}(n)$, $B_{j,i,k}(n)$ e $C_{j,i,k}(n)$ usando (2.7a-c), para $0 \le i,k \le N_P - j - 1$
8. j=j+1; vá para 4.

Tabela 2.3: Algoritmo FLAT

2.4.2 Quantização dos Coeficientes de Reflexão

Os coeficientes de reflexão são quantizados por meio de dez tabelas, uma para cada coeficiente de reflexão. O valor ótimo é o mais próximo do valor não quantizado (erro absoluto mínimo). Não é realizada nenhuma transformação dos coeficientes de reflexão antes da quantização.

Desde que cada estágio da estrutura treliça é otimizado independentemente para encontrar os coeficientes de reflexão, os primeiros estágios estimam as características mais dominantes do sinal e os últimos estágios são um refinamento do modelo. Então os primeiros coeficientes são quantizados com precisão maior que os outros por meio da seguinte distribuição de bits:

$$\begin{aligned} r_1 &= 6 \text{ bits} & r_2 &= 5 \text{ bits} \\ r_3 &= 5 \text{ bits} & r_4 &= 4 \text{ bits} \\ r_5 &= 4 \text{ bits} & r_6 &= 3 \text{ bits} \\ r_7 &= 3 \text{ bits} & r_8 &= 3 \text{ bits} \\ r_9 &= 3 \text{ bits} & r_{10} &= 2 \text{ bits} \end{aligned} \qquad (2.13)$$

Cada coeficiente utiliza um quantizador escalar não uniforme tabulado no padrão [6] (uma tabela para cada coeficiente).

2.5 Interpolação dos Coeficientes LPC

Os coeficientes autoregressivos LPC são obtidos por meio do procedimento de conversão dos coeficientes de reflexão, denominado de *step-up*. Estes coeficientes LPC são interpolados linearmente, o que melhora o desempenho do codificador.

$$\begin{aligned} \alpha_i(novo) &= 0{,}75a_i(prévio) + 0{,}25a_i(corrente), && \text{para subquadro 1.} \\ \alpha_i(novo) &= 0{,}50a_i(prévio) + 0{,}50a_i(corrente), && \text{para subquadro 2.} \\ \alpha_i(novo) &= 0{,}25a_i(prévio) + 0{,}75a_i(corrente), && \text{para subquadro 3.} \\ \alpha_i(novo) &= a_i(corrente), && \text{para subquadro 4.} \end{aligned} \qquad (2.14)$$

Os valores interpolados são convertidos em coeficientes de reflexão para testar a estabilidade. Se o módulo de um dos coeficientes de reflexão for maior ou igual a 1, então o filtro é instável. Se o filtro for instável são empregados os coeficientes não interpolados como descrito a seguir. No caso de instabilidade, para o subquadro 1, os coeficientes não interpolados utilizados são os coeficientes do quadro anterior. Para o subquadro 3 são utilizados os coeficientes do quadro atual. Para o subquadro 2 pode-se utilizar os coeficientes não interpolados do quadro anterior ou atual, escolhendo-se os coeficientes do quadro de maior energia. Se os quadros anterior e atual tiverem a mesma energia, emprega-se os coeficientes do quadro anterior.

2.6 Energia do Quadro

A energia do quadro é interpretada como a potência média do sinal de fala no intervalo de 20 ms. Para propósito de suavização de $R(0)$, este parâmetro é obtido pela média de duas janelas diferentes, uma de N_P até $N_A - 1$, e outra de 0 até $N_F - 1$. E os valores são normalizados pelo número de amostras,

$$R(0) = \frac{\phi(0,0) + \phi(N_P, N_P)}{2(N_A - N_P)}. \tag{2.15}$$

$R(0)$ é convertido em escala em dB relativo ao valor de fundo de escala, $R_{\max}$:

$$R_{dB} = 10 \log_{10}[R(0) / R_{\max}] \tag{2.16}$$

O fundo de escala $R_{\max}$ é um fator sistêmico definido no padrão. São reservados 5 bits para quantizar R_{dB} (32 níveis). Definindo-se o passo do quantizador em 2 dB e o nível máximo em –4 dB, obtém-se a Tabela 2.4.

A energia do quadro quantizada $R_q(0)$ é também interpolada :

$$\begin{aligned} R'_q(0) &= R_q(0)_{\,prévio} && \text{para subquadro 1} \\ R'_q(0) &= \sqrt{R_q(0)_{\,prévio} R_q(0)_{corrente}} && \text{para subquadro 2} \\ R'_q(0) &= R_q(0)_{\,corrente} && \text{para subquadro 3 e 4} \end{aligned} \tag{2.17}$$

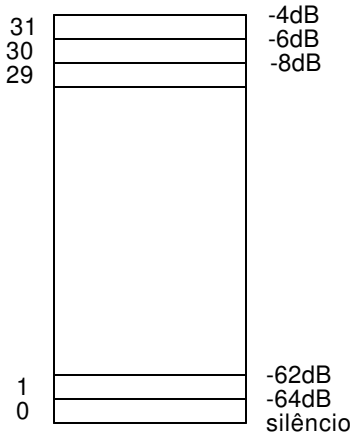

Tabela 2.4: Quantizador para energia do quadro

2.7 Filtro Perceptual W(z)

Sabendo que a percepção auditiva sofre efeitos de mascaramento de sinais, os erros causados por processos de codificação e decodificação podem ser otimizados por um filtro de ponderação perceptual. É utilizado um filtro linear o qual atenua as freqüências onde o erro é perceptualmente menos importante e amplifica as freqüências onde o erro é mais importante. O sinal de fala de entrada para cada subquadro deve ser filtrado pelo filtro de ponderação perceptual, definido por:

$$W(z) = \frac{1 - \sum_{i=1}^{N_P} \alpha_i z^{-i}}{1 - \sum_{i=1}^{N_P} \alpha_i \lambda^i z^{-i}}, \qquad (2.18)$$

onde $\alpha_i's$ são os coeficientes LPC interpolados para o subquadro e λ é o parâmetro de ponderação de ruído, que controla o nível de ruído sobre os formantes do sinal de fala.

2.8 Princípios de Busca do Dicionário

O codificador VSELP utiliza um procedimento de análise por síntese na determinação do vetor ótimo. A Figura 2.4 ilustra o diagrama em blocos do procedimento

de busca do dicionário. Uma explanação do processo de busca nos dicionários é dada a seguir.

Seja $g_i(n)$ a resposta de $H(z)$ ao estado zero para o vetor i de um dicionário genérico. Seja C_i a correlação cruzada de atraso zero entre os sinais :

$$C_i = \sum_{n=0}^{N-1} g_i(n)\, p(n)\,, \tag{2.19}$$

onde $p(n)$ é o sinal de saída do filtro perceptual menos a contribuição da resposta à entrada zero de $H(z)$ devido à energia do subquadro anterior.

A energia do vetor do dicionário filtrado é dada por :

$$G_i = \sum_{n=0}^{N-1} \{g_i(n)\}^2\,. \tag{2.20}$$

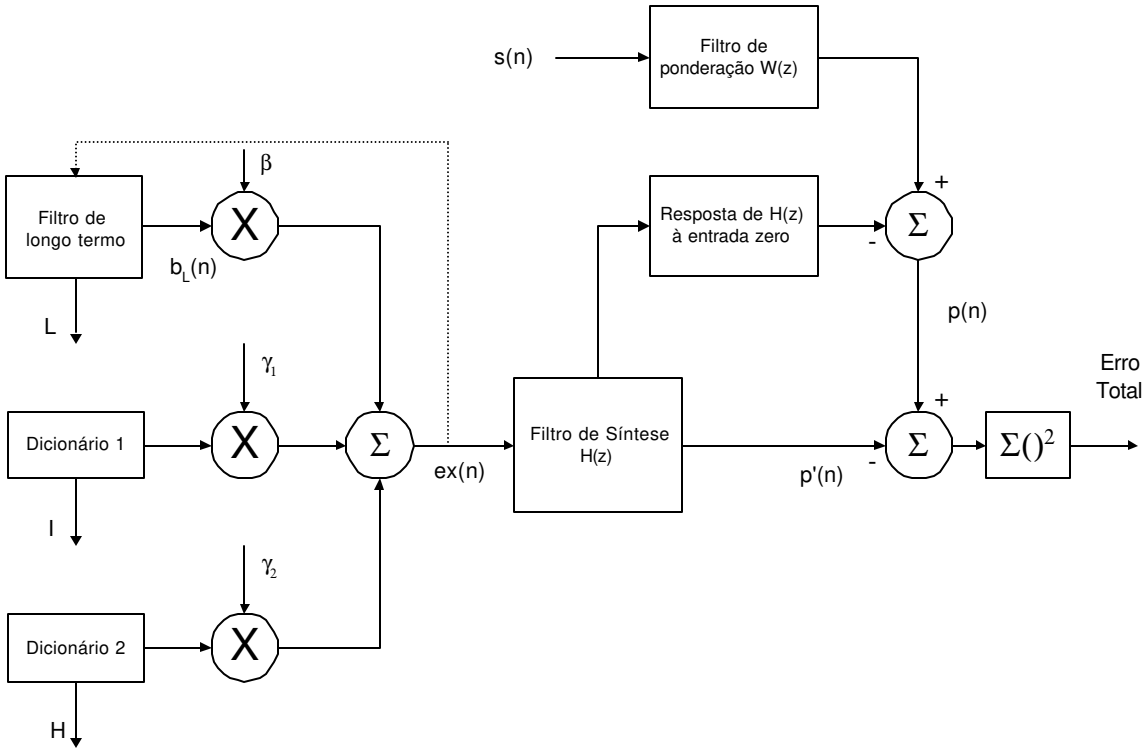

Figura 2.4: Diagrama em blocos do procedimento de busca do dicionário

O erro quadrático E_i total é dado por :

$$E_i = \sum_{n=0}^{N-1} \left[p(n) - \gamma_i g_i(n) \right]^2 , \tag{2.21}$$

onde γ_i é um fator de ganho do vetor i do dicionário. Seu valor ótimo é encontrado derivando-se E_i em relação a γ_i e igualando-se a zero :

$$\frac{\partial E_i}{\partial \gamma_i} = \sum_{n=0}^{N-1} \left[-2\, p(n)\, g_i(n) + 2\gamma_i\, g_i^2(n) \right] = 0 \tag{2.22}$$

$$\gamma_i = \frac{\sum_{n=0}^{N-1} p(n)\, g_i(n)}{\sum_{n=0}^{N-1} g_i^2(n)} = \frac{C_i}{G_i} \tag{2.23}$$

Substituindo em (2.21) e tomando-se as equações (2.19) e (2.20):

$$E_i = \left[\sum_{n=0}^{N-1} p^2(n) \right] - \frac{C_i^2}{G_i} \tag{2.24}$$

Como o termo entre colchetes é a energia de $p(n)$ no subquadro (constante), logo quando C_i^2 / G_i é maximizado o erro ponderado total é minimizado. O vetor do dicionário que maximiza C_i^2 / G_i é o vetor ótimo. Este princípio é aplicado tanto para o preditor de *pitch* quanto para os dois dicionários fixos como explanado a seguir.

2.9 Atraso de Preditor de Longo Termo, L

Este atraso é freqüentemente chamado de atraso de *pitch*. Em codificadores de análise por síntese, existem dois métodos para se determinar o atraso de *pitch*. O primeiro método é o de malha aberta, no qual o atraso é determinado diretamente a partir do sinal de entrada ou do sinal residual. O segundo método de malha fechada é um método de otimização. Em [17] foi mostrado que um preditor de malha fechada de primeira ordem fornece resultados superiores aos de malha aberta.

No VSELP, é empregado um preditor de primeira ordem com busca em malha fechada. A faixa de busca do *pitch* está definida na Tabela 2.5.

55Hz	400 Hz
146 amostras	20 amostras
18,25 ms	2,5 ms.

Tabela 2.5: Faixa de variação do *pitch*.

É relativamente simples encontrar o atraso pelo método de malha fechada quando o atraso é maior ou igual ao tamanho do subquadro, $L \geq N$. Trata-se de um problema de otimização linear. Porém, quando $L < N$, o problema torna-se não linear [35]. O filtro preditor de *pitch* empregado no codificador VSELP foi modificado de forma a garantir sempre a otimização linear:

$$B_n(z) = \frac{1}{1 - \beta z^{-\left\lfloor \frac{n+L}{L} \right\rfloor L}}, \tag{2.25}$$

onde $\lfloor x \rfloor$ é o maior inteiro menor ou igual a x (função *floor*), n é a amostra no subquadro no intervalo $0 \leq n \leq N-1$. Quando a condição de otimização linear não é satisfeita, o preditor acima busca o dobro do *pitch* real. Existem dois casos a serem analisados:

1º caso: $L \geq N$,

neste caso $\left\lfloor \frac{n+L}{L} \right\rfloor = 1$. Então o filtro preditor se reduz a

$$B_n(z) = \frac{1}{1 - \beta z^{-L}}. \tag{2.26}$$

2º caso: $L < N$ e $L \leq n \leq 2L-1$,

neste caso $\left\lfloor \frac{n+L}{L} \right\rfloor = 2$. O filtro preditor fica

$$B_n(z) = \frac{1}{1 - \beta z^{-2L}}. \tag{2.27}$$

Em [35] é mostrado que este procedimento torna o problema linear. Esta condição permite a otimização simultânea de todos os termos de ganho γ_1 e γ_2 e do coeficiente β.

A busca do atraso de *pitch* é similar a uma busca no dicionário, onde o dicionário é definido pelo estado do filtro de longo termo (constante para o dicionário em questão), e o vetor específico do dicionário é o atraso *L* do preditor de longo termo.

Sejam :

$L_{\min}$ - menor valor possível para o atraso L ,

$r(n)$ - resposta devido ao estado do filtro preditor de longo termo, $n < 0$,

$b_L(n)$ - saída do filtro preditor de *pitch* para o atraso L,

$h(n)$ - resposta ao impulso de $H(z)$,

$b'_L(n)$ - sinal $b_L(n)$ filtrado por $H(z)$.

Vimos que o preditor ótimo é o que maximiza C_L^2/G_L onde G_L é a energia de $b'_L(n)$

$$G_L = \sum_{n=0}^{N-1} \{b'_L(n)\}^2 \tag{2.28}$$

e C_L é a correlação cruzada de $b'_L(n)$ e $p(n)$

$$C_L = \sum_{n=0}^{N-1} b'_L(n)p(n)\,. \tag{2.29}$$

O período de *pitch* em amostras varia de 20 a 146, sendo necessários 127 passos (7 bits) para quantizá-lo. O código 128 é reservado para indicar segmentos surdos, para os quais o preditor de *pitch* não é usado. Quando a correlação não é positiva o preditor de *pitch* é desabilitado. Visando tornar a busca eficiente, $b'_L(n)$ é definido como

$$b'_L(n) = \sum_{i=0}^{\left\lfloor \frac{n}{L} \right\rfloor} z_L(n - iL), \tag{2.30}$$

onde $z_L(n) = \sum_{i=0}^{\min(n,L-1)} r(i-L)h(n-i)$.

Além do mais $z_L(n)$ pode ser calculado recursivamente :

$$\begin{aligned} z_L(n) &= z_{L-1}(n-1) + r(-L)h(n), \quad \text{para } 1 \le n \le N-1; \\ z_L(0) &= r(-L)h(0), \qquad\qquad \text{para} \qquad n = 0. \end{aligned} \tag{2.31}$$

Observe que a resposta ao impulso do filtro de síntese, $h(n)$, na prática tem a energia concentrada nas primeiras dez ou vinte amostras. Isto permite truncar a duração da resposta ao impulso com o objetivo de redução do esforço computacional:

$$h'(n) = \begin{cases} h(n) & \text{para } 0 \le n \le N_T - 1 \\ 0 & \text{para } N_T \le n \end{cases} . \qquad (2.32)$$

No padrão foi tomado N_T=21.

Estas equações permitem escrever o seguinte algoritmo :

1º) Calcula $z_{L\min}(n)$:

$$z_{L_{\min}}(n) = \sum_{i=0}^{\min(n, L_{\min}-1)} r(i - L_{\min}) h(n-i) \quad \text{para } 0 \le n \le N-1 .$$

2º) De $L = L_{\min}$ a $L = \mathrm{L}_{\max}$

Calcule $b'_L(n)$:

para $L \ge N$, $b'_L(n) = z_L(n)$ $\quad 0 \le n \le N-1$;

para $L < N$, $b'_L(n) = z_L(n)$ se $n < L$;

$b'_L(n) = z_L(n) + z_L(n-L)$ se $n \ge L$.

Calcule C_i^2 / G_i .

Armazene C_i^2 / G_i máximo.

Atualiza $z_L(n)$

$z_L(n) = z_{L-1}(n-1) + r(-L)h(n)$ para $1 \le n \le N-1$;

$z_L(0) = r(-L)h(0)$ para $n = 0$.

3º) Codifique *L*:

$LAG\_x = L - 19$ se o preditor não for desativado;

$LAG\_x = 0$ se o preditor for desativado;

onde *x* representa o número do subquadro.

Tabela 2.6 : algoritmo do cálculo do atraso de *pitch*

2.10 Busca no Dicionário

A diferença principal entre o codificador VSELP e outros codificadores da família CELP é

a estrutura do dicionário. O VSELP usa dois dicionários, os quais contêm apenas *M*=7 vetores base com dimensão *N*=40 elementos. Cada dicionário contém $2^M = 128$ vetores de código gerados a partir dos *M* vetores base por meio da seguinte combinação linear :

$$u_{k,i}(n) = \sum_{m=1}^{M} \theta_{i,m} v_{k,m}(n) \tag{2.33}$$

onde k = 1 e 2 para o primeiro e segundo dicionário, respectivamente, $0 \le i \le 2^M - 1$, $0 \le n \le N-1$ e

$\theta_{i,m} = +1$ se o bit *m* da palavra-código *i* for igual a 1,

$\theta_{i,m} = -1$ se o bit *m* da palavra-código *i* for igual a 0.

Portanto, o vetor-código *i* é construído como a soma de *M* vetores base onde o sinal de cada bit do vetor base é determinado pelo estado do bit correspondente na palavra-código *i*. A palavra-código é uma palavra de 7 bits. Tomando como exemplo, para a palavra-código 0000001 tem-se $\theta_{11} = 1$ e $\theta_{1,2} = \theta_{1,3} = \theta_{1,4} = \theta_{1,5} = \theta_{1,6} = \theta_{1,7} = -1$. Então

$$u_{1,1}(n) = \sum_{m=1}^{M} \theta_{i,m} v_{1,m} = v_{1,1} - v_{1,2} - v_{1,3} - v_{1,4} - v_{1,5} - v_{1,6} - v_{1,7}. \tag{2.34}$$

O procedimento de busca no dicionário acontece após a determinação do preditor de atraso de *pitch*. A busca é realizada seqüencialmente para o primeiro, e em seguida, para o segundo dicionário do VSELP. Duas palavras-código são definidas para o primeiro e segundo dicionário (*I* e *H* respectivamente).

Seja $q_{k,m}(n)$ a resposta ao estado zero de *H*(*z*) ao vetor base $v_{k,m}(n)$. Então a saída filtrada é dada pelo teorema da superposição :

$$f_{k,i}(n) = \sum_{m=1}^{M} \theta_{i,m} q_{k,m}(n). \tag{2.35}$$

Note a redução de complexidade computacional : calculam-se 7 respostas ao estado zero ao invés de 128. As demais são obtidas por meio da combinação linear acima.

Para desacoplar o procedimento de busca dos vetores ótimos nos diversos dicionários é empregado um método sub ótimo: uma busca seqüencial em um dicionário por vez com o emprego da ortogonalização de Gram-Schmidt.

O procedimento de ortogonalização de Gram-Schmidt para se obter uma base

ortogonal a partir de um conjunto de funções usa a propriedade de que o erro de projeção mínimo é sempre ortogonal aos elementos de base.

A busca do vetor-código para o primeiro dicionário (*I*) considera a seleção prévia do preditor de *pitch*. A busca do segundo dicionário (*H*) considera o atraso de *pitch* e a palavra código selecionada para o primeiro dicionário (*I*). O processo de ortogonalização de Gram-Schmidt é utilizado para desacoplar o processo de busca.

Da Figura 2.4 o sinal de erro pode ser expresso por:

$$e(n) = p(n) - \beta b'_L(n) - \gamma_1 f_{1,I}(n) - \gamma_2 f_{2,H}(n) \tag{2.36}$$

e o erro quadrado total :

$$\sum_{n=0}^{N-1} e^2(n) . \tag{2.37}$$

Antes de se fazer a busca do primeiro vetor-código $f_{1,I}(n)$, cada vetor base do primeiro dicionário pode ser feito ortogonal à seqüência já determinada do dicionário adaptativo $b'_L(n)$. Define-se as grandezas :

$$\Gamma = \sum_{n=0}^{N-1} (b'_L(n))^2 \tag{2.38}$$

e

$$\Psi_m = \sum_{n=0}^{N-1} b'_L(n) q_{1,m}(n) \tag{2.39}$$

para $0 \le m \le M$, lembrando que $q_{1,m}(n)$ é a resposta ao estado zero de $H(z)$ ao vetor base $v_{1,m}(n)$.

Os vetores base ortogonalizados são obtidos pelo procedimento de Gram-Schmidt:

$$q'_{1,m}(n) = q_{1,m}(n) - \frac{\Psi_m}{\Gamma} b'_L(n) \tag{2.40}$$

para $0 \le m \le M$ e $0 \le n \le N-1$.

Os vetores código ortogonalizados filtrados podem ser expressos pelo princípio da superposição:

$$f'_{1,i}(n) = \sum_{m=1}^{M} \theta_{im} q'_{1,m}(n) \tag{2.41}$$

para $0 \le i \le 2^M - 1$ e $0 \le n \le N-1$.

O erro total a ser minimizado para a otimização do primeiro dicionário (*I*) é :

$$E'_{1,i} = \sum_{n=0}^{N-1} \left[p(n) - \gamma'_1 f'_{1,i}(n) \right]^2 \tag{2.42}$$

onde γ'_1 é otimizado para cada vetor código. Observe que esta equação de erro é independente de β (ganho do preditor de *pitch*). A otimização para cada vetor código é assim simplificada a um problema de predição linear de primeira ordem por meio do processo de ortogonalização. A seqüência $q'_{1,m}(n)$ é ortogonal a $b'_L(n)$, permitindo a redução da ordem do problema de otimização. Após a determinação do primeiro vetor código, os vetores base filtrados do segundo dicionário devem ser ortogonalizados a $b'_L(n)$ e $f_{1,I}(n)$.

Os vetores códigos filtrados e ortogonalizados do segundo dicionário podem ser expressos por :

$$f'_{2,i}(n) = \sum_{m=1}^{M} \theta_{im} q'_{2,m}(n) \tag{2.43}$$

para $0 \le i \le 2^M - 1$.

Para o segundo dicionário o erro quadrado total a ser minimizado é :

$$E'_{2,i} = \sum_{n=0}^{N-1} \left[p(n) - \gamma'_2 f'_{2,i}(n) \right]^2 \tag{2.44}$$

2.11 Otimização Conjunta de Ganhos

Embora a determinação dos vetores ótimos seja feita seqüencialmente, os ganhos são otimizados conjuntamente. O erro perceptual pode ser expresso por :

$$e(n) = p(n) - \beta c'_0(n) - \gamma_1 c'_1(n) - \gamma_2 c'_2(n) \tag{2.45}$$

onde $0 \le n \le N-1$,

$p(n)$ é a entrada filtrada menos a resposta a entrada zero de $H(z)$,

$c'_o(n)$ é o vetor do preditor de *pitch* filtrado, $b'_L(n)$,

$c_1'(n)$ é o vetor-código filtrado selecionado do dicionário 1, $f_{1,I}(n)$,

$c_2'(n)$ é o vetor-código filtrado selecionado do dicionário 2, $f_{2,H}(n)$,

β é o coeficiente de ganho do preditor de *pitch*,

γ_1 é o ganho do vetor-código do dicionário 1,

γ_2 é o ganho do vetor-código do dicionário 2.

O erro quadrado total é

$$E = \sum_{n=0}^{N-1} e^2(n) = \sum_{n=0}^{N-1} \left(p(n) - \beta c_0'(n) - \gamma_1 c_1'(n) - \gamma_2 c_2'(n)\right)^2 . \tag{2.46}$$

Utilizando-se como notação das correlações :

$$R_{pp} = \sum_{n=0}^{N-1} p(n)\,p(n), \tag{2.47}$$

$$R_{pc}(k) = \sum p(n) c_k'(n); \quad k = 0,...,2, \tag{2.48}$$

$$R_{cc}(k,j) = \sum_{n=0}^{N-1} c_k'(n) c_j'(n); \quad k = 0,...,2; \quad j = k,...,2, \tag{2.49}$$

e sabendo-se que

$$R_{cc}(k,j) = R_{cc}(j,k), \tag{2.50}$$

o erro quadrado total pode ser expresso por :

$$\begin{aligned} E = R_{pp} - 2\beta R_{pc}(0) - 2\sum_{j=1}^{2} \gamma_j R_{pc}(j) = 2\beta \sum_{j=1}^{2} \gamma_j R_{cc}(0,j) + 2\gamma_1\gamma_2 R_{cc}(1,2) \\ + \beta^2 R_{cc}(0,0) + \sum_{j=1}^{2} \gamma_j^2 R_{cc}(j,j). \end{aligned} \tag{2.51}$$

Os termos de correlação são definidos e o problema consiste na determinação simultânea de β, γ_1 e γ_2. Para minimizar E toma-se sua derivada parcial em relação a β, γ_1 e γ_2 e iguala-as a zero, obtendo-se três equações lineares a três incógnitas. Porém, ao invés de solucionar o sistema de equações, emprega-se a quantização vetorial do conjunto de ganhos $\{\beta, \gamma_1, \gamma_2\}$. A determinação da palavra-código ótima requer o cálculo das correlações nas equações (2.47, 2.48 e 2.49) e o cálculo de E para cada vetor do dicionário. O vetor-código que minimizar o erro é escolhido.

2.12 Transformação dos Ganhos em G_S, P_0 e P_1

O sinal de excitação $e_x(n)$ para um dado subquadro é a combinação linear do preditor de *pitch* escalonado por β e dos vetores-código escalonados por γ_1 e γ_2, isto é, por seus respectivos ganhos.

$$e_x(n) = \beta c_0(n) + \gamma_1 c_1(n) + \gamma_2 c_2(n) \tag{2.52}$$

onde $0 \le n \le N-1$,

$c_o(n)$ é o vetor de predição de *pitch* não filtrado ($b_L(n)$),

$c_1(n)$ é o vetor-código não filtrado selecionado do dicionário 1 ($u_{1,I}(n)$),

$c_2(n)$ é o vetor-código não filtrado selecionado do dicionário 2 ($u_{2,H}(n)$).

Assumimos que $c_o(n)$, $c_1(n)$ e $c_2(n)$ são descorrelatados. Em geral isto não é verdade, mas se for aplicado no codificador e no decodificador fornece resultados coerentes.

A energia de cada vetor de excitação é dada por

$$R_x(k) = \sum_{n=0}^{N-1} c_k^2(n), \qquad k = 0,...,2. \tag{2.53}$$

A energia total da excitação do subquadro *R* é

$$R = \sum_{n=0}^{N-1} {e_x}^2(n) = \sum_{n=0}^{N-1} \left(\beta c_0(n) + \gamma_1 c_1(n) + \gamma_2 c_2(n)\right)^2 \tag{2.54}$$

Assumindo a ortogonalidade mencionada previamente, então *R* pode ser expressa como

$$R = \beta^2 R_x(0) + \gamma_1^2 R_x(1) + \gamma_2^2 R_x(2)\,. \tag{2.55}$$

Definimos os parâmetros: P_0, que é a contribuição de energia do vetor de predição de *pitch*,

$$P_0 = \frac{\beta^2 R_x(0)}{R} \tag{2.56}$$

onde $0 \le P_0 \le 1$; P_1, que é a contribuição de energia do vetor-código selecionado do primeiro dicionário,

$$P_1 = \frac{\gamma_1^2 R_x(1)}{R} \tag{2.57}$$

onde $P_0 + P_1 \le 1$; P_2, que é a contribuição de energia do vetor-código selecionado do

segundo dicionário :

$$P_2 = \frac{\gamma_2^2 R_x(2)}{R} \tag{2.58}$$

onde $P_0 + P_1 + P_2 = 1$ e R_S, a energia residual no subquadro. R_S pode ser expresso por meio da equação do erro quadrado mínimo:

$$R_S = NR'_q(0)\prod_{i=1}^{N_P}(1 - r_i^2) \tag{2.59}$$

onde r_i é o i-ésimo coeficiente de reflexão do subquadro e $R'_q(0)$ é a energia quantizada interpolada do sinal (equação 2.17).

O parâmetro de *offset* de energia G_S ajusta o valor estimado de R_S :

$$R = G_S R_S . \tag{2.60}$$

Então β, γ_1 e γ_2 são escritos em relação aos novos parâmetros

$$\beta = \sqrt{\frac{R_S . G_S . P_0}{R_x(0)}}, \tag{2.61}$$

$$\gamma_1 = \sqrt{\frac{R_S . G_S . P_1}{R_x(1)}} \quad \text{e} \tag{2.62}$$

$$\gamma_2 = \sqrt{\frac{R_S . G_S (1 - P_0 - P_1)}{R_x(2)}} \tag{2.63}$$

2.13 Quantização Vetorial e Codificação de G_S, P_0 e P_1

A quantização vetorial de β, γ_1 e γ_2 é substituída pela quantização simultânea de G_S, P_0 e P_1. A quantização de G_S, P_0 e P_1 é independente do nível do sinal de entrada, pois a quantização de $R(0)$ em $R_q(0)$ normaliza a energia absoluta do sinal. Outra vantagem é que G_S, P_0 e P_1 são limitados ($G_S, P_0, P_1 \leq 1$). Estes parâmetros são quantizados utilizando o algoritmo LBG eβ, γ_1 e γ_2 são obtidos através das equações (2.61), (2.62) e (2.63) e do valor de R_S, que é calculado a partir dos coeficientes de reflexão quantizados.

2.14 Atualização do Estado do Filtro Preditor de *Pitch*

Após a determinação de todos os parâmetros do subquadro e a respectiva quantização, o estado do filtro preditor de *pitch* deve ser preparado para o processamento do próximo subquadro.

Observando-se da figura 2.5 que a excitação combinada $e_x(n)$ se expressa por:

$$e_x(n) = \beta_q b_L(n) + \gamma_{1_q} u_{1,I}(n) + \gamma_{2_q} u_{2,H}(n) \tag{2.64}$$

para $0 \le n \le N-1$.

O estado do filtro preditor de *pitch* é atualizado por

$$\begin{aligned} r(n) &= r(n+40) \quad \text{para } -146 \le n \le -41, \\ r(n) &= e_x(n+40) \quad \text{para } -40 \le n \le -1. \end{aligned} \tag{2.65}$$

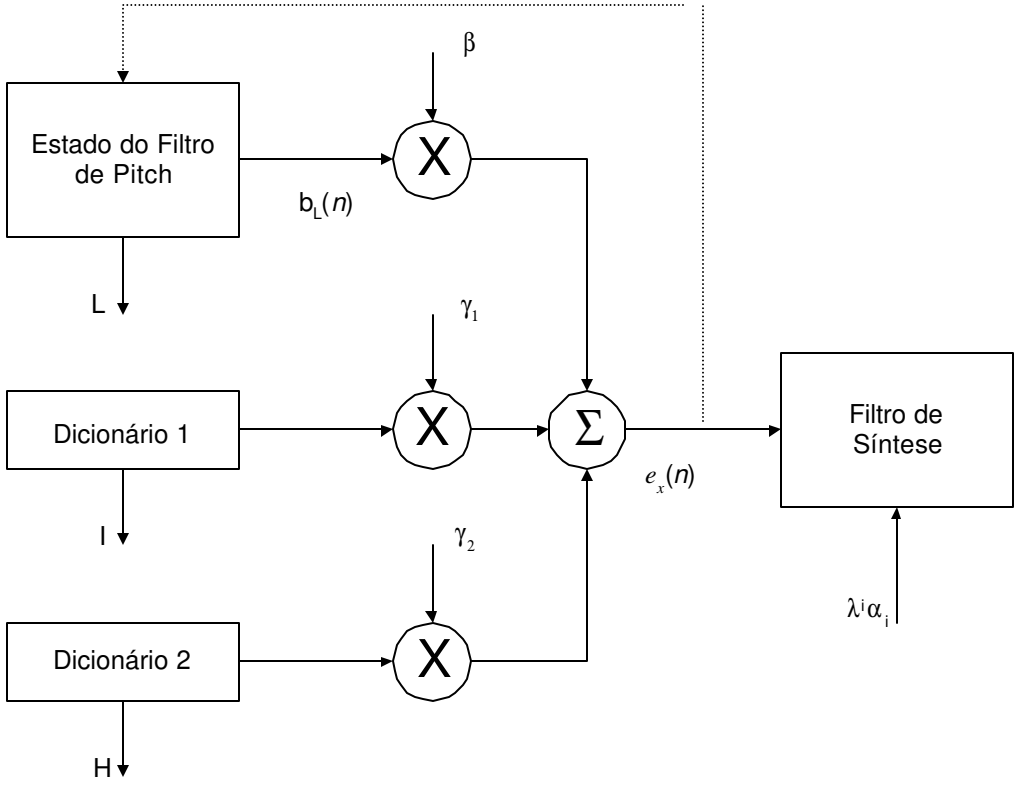

Figura 2.5: Diagrama em blocos do sintetizador ponderado.

O filtro de síntese ponderado é atualizado introduzindo-se as 40 amostras de $e_x(n)$ no mesmo, gerando o estado para o próximo subquadro.

2.15 Decodificador

A figura 2.6 ilustra o diagrama em blocos do decodificador VSELP. Observa-se que em relação ao codificador, tem-se os pós-filtros espectrais $\hat{H}(z)$ e $\tilde{H}(z)$, e o controle de escala. A seguir explanamos estes blocos adicionais.

2.15.1 Pós-Filtro Espectral Adaptativo

O filtro perceptual ponderador de erro é um processamento efetivo nos codificadores CELP que exploram o efeito de mascaramento auditivo, ou seja, distribui de forma não uniforme o ruído ao longo do espectro de freqüências. Entretanto, para taxas de bits muito baixas, não é possível concentrar totalmente o ruído sob os picos do espectro, e nas regiões de vales espectrais o ruído se torna audível. A técnica de pós-filtro adaptativo busca reduzir este efeito de forma a atenuar dinamicamente o espectro do sinal de voz nos vales espectrais. Esta técnica, embora distorça o espectro do sinal de voz nos vales, reduz o ruído audível.

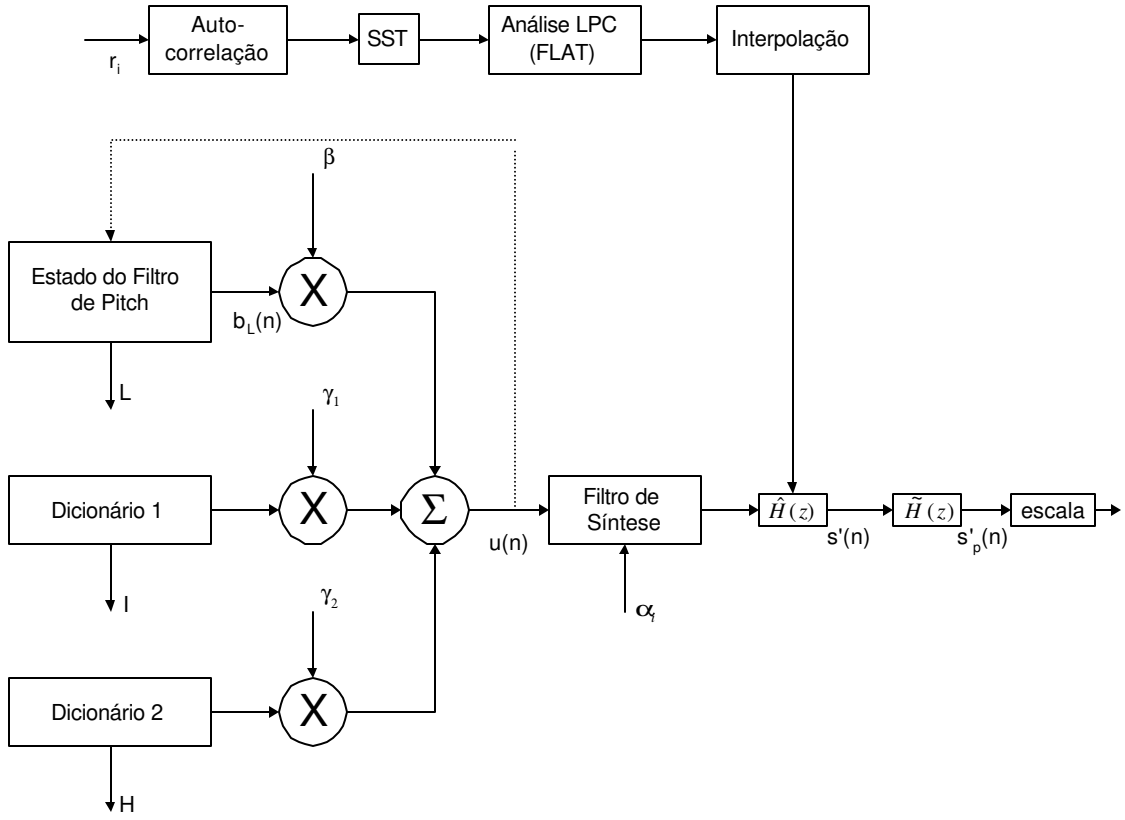

Figura 2.6: Diagrama em blocos do decodificador VSELP.

Com certa freqüência, o espectro do LPC para sons sonoros mostra um decaimento espectral ao redor de 6 dB/oitava. Para compensar este decaimento é introduzido como parte integrante do filtro um termo (fixo) no numerador. Toda informação necessária para atenuar dinamicamente os vales encontra-se no filtro de síntese LPC (α_i representa os coeficientes LPC interpolados para o subquadro).

O pós-filtro adaptativo empregado no VSELP é :

$$\hat{H}(z) = \frac{1 - \sum_{i=1}^{p} \eta_i z^{-i}}{1 - \sum_{i=1}^{p} \alpha_i \lambda^i z^{-i}} \tag{2.66}$$

onde $\lambda = 0{,}8$.

O espectro devido ao denominador, anulando-se os coeficientes η_i do numerador, é ilustrado na figura 2.7. O efeito de se multiplicar por λ^i é o de deslocar radialmente todos os pólos para a origem. O polinômio do numerador na equação é uma versão do denominador cujo espectro foi suavizado. Para se determinar os coeficientes do numerador η_i, a função de autocorrelação da resposta ao impulso do filtro só-pólos correspondente ao polinômio do denominador é calculada para atrasos de 0 a 10. A seqüência de autocorrelação é submetida à técnica de expansão de largura de banda por meio da multiplicação por uma janela binomial como foi feito no codificador. Em seguida os coeficientes autoregressivos são novamente calculados por meio da recursão de Levinson-Durbin.

O filtro fixo de correção do decaimento espectral é :

$$\tilde{H}(z) = 1 - \mu z^{-1}, \tag{2.67}$$

onde $\mu = 0{,}4$.

O pós-filtro introduz ainda uma variação no ganho do sinal de saída que deve ser compensada por meio de um controle automático de ganho. Deve-se garantir um ganho aproximadamente unitário entre o sinal de saída e o de entrada do pós-filtro. O fator de escala, expresso pela raiz quadrada da razão entre a energia do sinal de entrada e a energia do sinal de saída,

$$S_f = \sqrt{\frac{\sum_{n=0}^{N-1} \hat{s}^2(n)}{\sum_{n=0}^{N-1} \hat{s}_p^2(n)}}, \tag{2.68}$$

é filtrado por um filtro passa-baixas:

$$S'_f(n) = (0{,}9875 S'_f(n-1)) + (0{,}0125 S_f). \tag{2.69}$$

A saída do pós filtro espectral, $S'_p(n)$ (Figura 2.6), é multiplicado por S'_f.

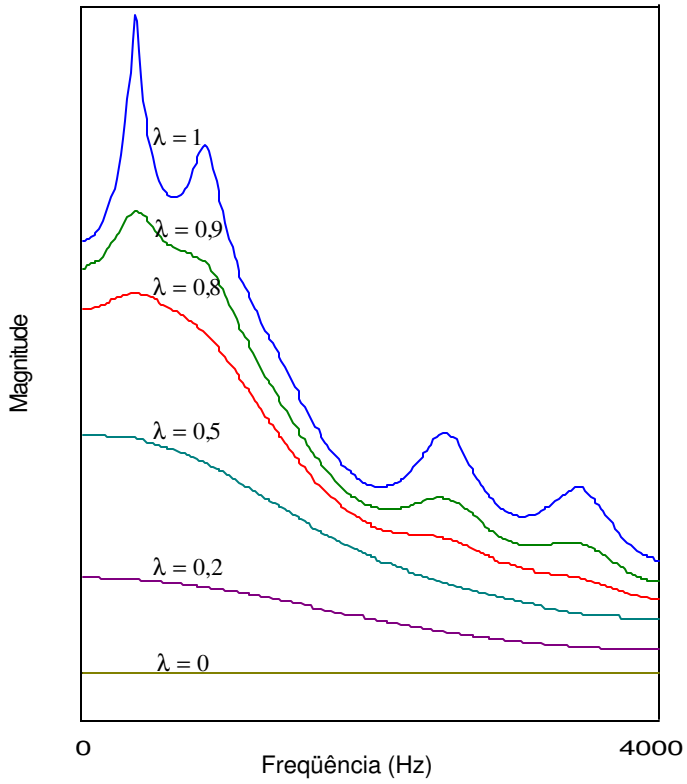

Figura 2.7: Magnitude espectral para pós-filtro só-polos para diferentes valores de λ.

Capítulo 3

O codificador *Enhanced Full Rate* - EFR

Neste capítulo é revisado o codificador de fala padronizado pela *Telecommunications Industry Association* (TIA/EIA) denominado *Enhanced Full Rate* para telefonia celular (padrão IS-641) [7].

3.1 Introdução

O codificador IS641 também é um codificador de fala da família CELP. Portanto, existem similaridades entre este e o VSELP. No EFR transmitem-se os pares de freqüências espectrais (LSF), enquanto que no VSELP transmitem-se os coeficientes de reflexão. Existem melhorias, também, no cálculo do período de *pitch* e na codificação do sinal de excitação.

O filtro preditor é calculado para um quadro, de 20 ms de duração, e seus coeficientes são transformados em LSF e transmitidos. A estimação de *pitch* é realizada pelos métodos de malha aberta e de malha fechada. Para a codificação do sinal de excitação é utilizada a técnica ACELP (*Algebraic Code-Excited Linear Predictive*).

O codificador EFR emprega a taxa 7,4 kbits/s para a fala e 5,6 kbits/s para a codificação de canal, tendo a taxa final de 13 kbits/s. O sinal sonoro pode ser amostrado a 8 kHz e quantizado a 16 bits, ou utilizar o sinal PCM de 8 bits. O sinal de saída do decodificador possui 16 bits.

A cada quadro de 20 ms é feita a análise LPC. Os coeficientes autoregressivos, calculados pelo algoritmo de Levinson-Durbin, são convertidos em pares espectrais de freqüências (LSF). Estes são quantizados usando quantização vetorial partida (*Split Vector Quantization* - SVQ), na qual os coeficientes são divididos em grupos e estes são quantizados separadamente. Os dicionários fixo e adaptativo são transmitidos a cada subquadro de 5 ms. Os parâmetros LPC são interpolados para serem utilizados a cada subquadro.

A estimativa de período de *pitch* é realizada através de dois métodos: o método de malha aberta e o método de malha fechada. A análise pelo método de malha aberta é realizada duas vezes por quadro. A partir desta estimativa é feita a análise em malha fechada a cada subquadro. O sinal residual, resultado da subtração do sinal estimado e da estimativa do *pitch* do sinal de fala, é utilizado para encontrar o vetor ótimo do dicionário fixo (algébrico). O ganho dos dicionários fixo e algébrico são quantizados juntos. A cada subquadro as memórias dos filtros são atualizadas para processamento do próximo segmento. Um diagrama em blocos do codificador pode ser visto na Figura 3.1. Cada bloco do codificador é explanado a seguir.

3.2 Pré-Processamento

O sinal de fala é pré-processado antes de iniciar o processo de codificação. Este pré-processamento consiste em duas funções combinadas em um único filtro: filtragem passa-altas e redução de faixa dinâmica (escala). Para evitar que ocorra *overflow*, quando o codificador é implementado em ponto-fixo, faz-se o abaixamento de escala dividindo-se o sinal de fala por fator de 2.

A filtragem passa-altas com freqüência de corte em 80 Hz (-3 dB) elimina freqüências indesejáveis como a freqüência de 60 Hz da rede elétrica. A atenuação na freqüência de 60 Hz é de –12 dB.

A resposta em freqüência resultante é representada na Figura 3.2.

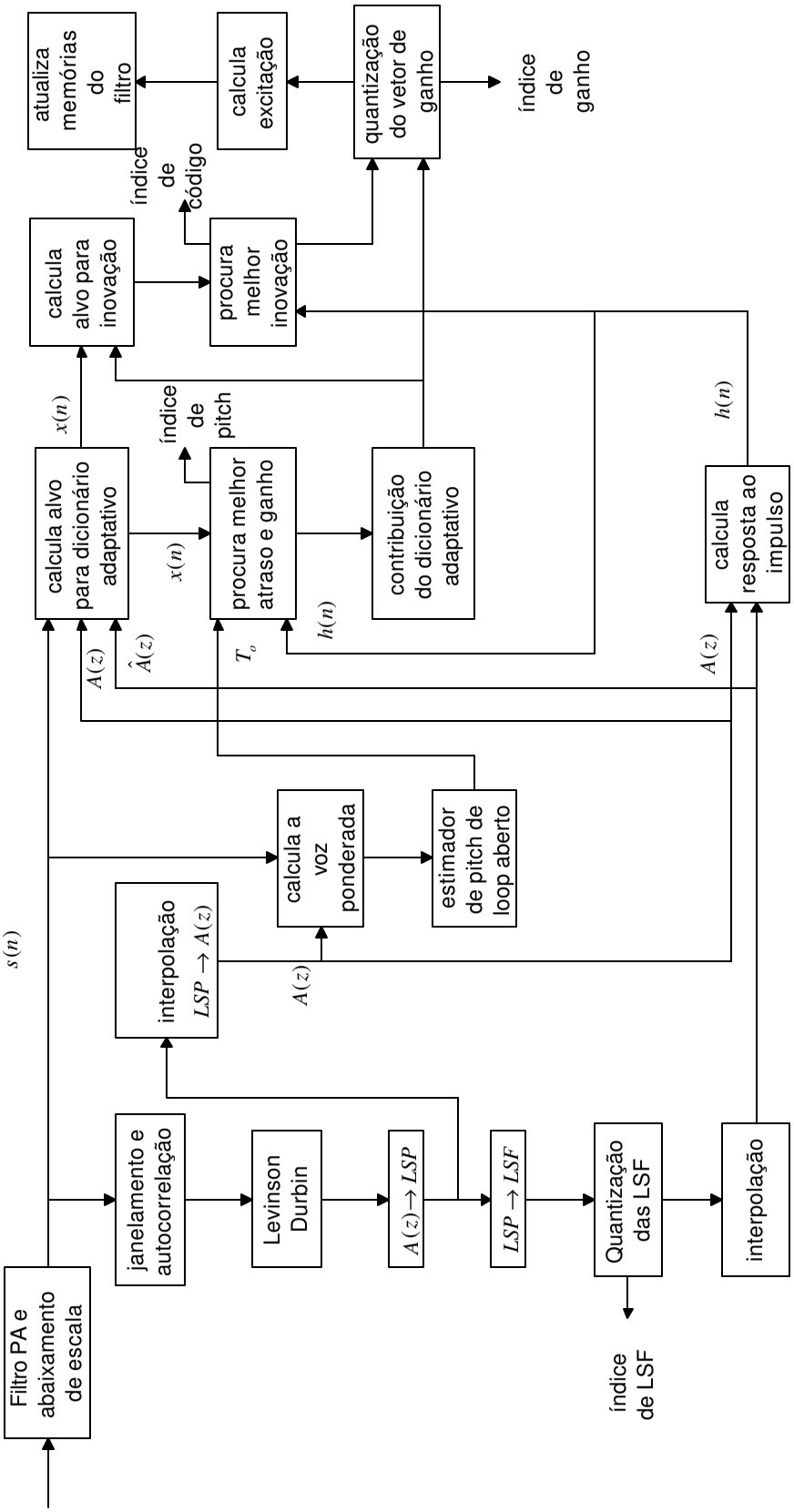

Figura 3.1: Diagrama em blocos do codificador EFR.

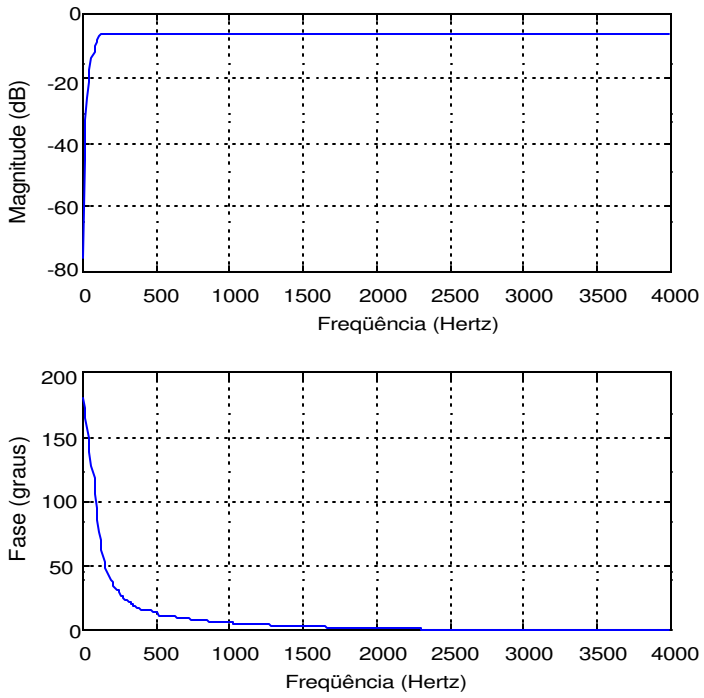

Figura 3.2: Resposta em magnitude e fase do filtro de pré-processamento.

3.3 Predição linear

Assim como no VSELP, a predição linear é realizada uma vez por quadro (160 ms). A obtenção dos coeficientes do filtro preditor para cada subquadro é realizada por interpolação tal como ocorre no VSELP. O sinal é filtrado por uma janela assimétrica de 30 ms de duração, a qual está centrada no quadro de 20 ms e é ilustrada na Figura 3.3. A duração do quadro é de 20 ms, portanto há um atraso de 5 ms (40 amostras) no sinal usado para o cálculo da autocorrelação.

A autocorrelação do sinal de fala janelado é calculada por:

$$r(k) = \sum_{n=k}^{239} s'(n)s'(n-k), \quad k = 0,...,10\,. \tag{3.1}$$

Lembrando que a modelagem autoregressiva tem tendência a subestimar a largura de bandas dos formantes, para compensar esta estimativa é realizada a expansão por meio de uma janela exponencial aplicada à função de autocorrelação. Esta janela é dada por:

$$w(i) = \exp\left[-\frac{1}{2}\left(\frac{2\pi f_0 i}{f_s}\right)^2\right], \tag{3.2}$$

para $i = 1,\ldots,10$. A freqüência $f_0 = 60\,\text{Hz}$ é a expansão de banda e $f_s = 8000\,\text{Hz}$ é a freqüência de amostragem.

Após esta ponderação dos valores de autocorrelação, são calculados os coeficientes do filtro preditor linear por meio do algoritmo Levinson-Durbin [19].

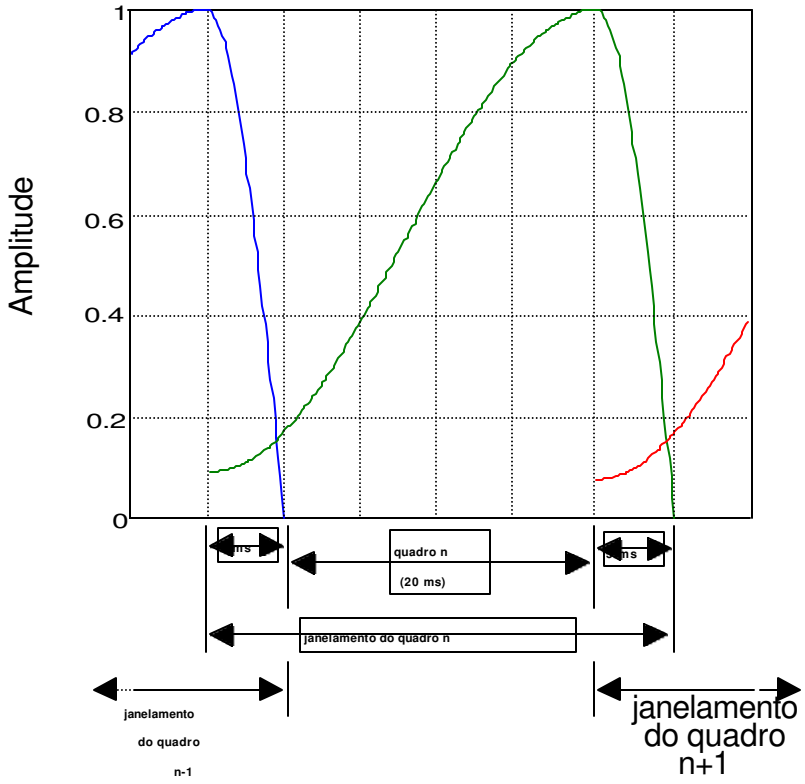

Figura 3.3 : Janela assimétrica.

3.4 Conversão para LSF

Os parâmetros transmitidos pelo codificador são os pares de freqüências espectrais (*LSF - line spectrum frequencies*) [20]. Eles têm informação dos formantes do sinal sonoro. São definidas como as raízes dos polinômios

$$f'_1(z) = A(z) + z^{-11}A(z^{-1}) \tag{3.3a}$$

$$f'_2(z) = A(z) - z^{-11}A(z^{-1}) \tag{3.3b}$$

onde $A(z) = 1 - \sum_{i=1}^{p} a_i z^{-i}$.

Os coeficientes do polinômio $f_1'(z)$ têm simetria par em relação ao coeficiente central e os do polinômio $f_2'(z)$ têm simetria ímpar. Suas raízes são complexas conjugadas e encontram-se sobre o círculo unitário e alternam-se entre si. O polinômio $f_1'(z)$ tem uma raiz localizada em z = -1 e $f_2'(z)$ tem uma raiz localizada em z = +1. Para eliminar esta redundância, calculam-se novos polinômios

$$G_1(z) = \frac{f'_1(z)}{1+z^{-1}}, \text{ e} \tag{3.4a}$$

$$G_2(z) = \frac{f'_2(z)}{1-z^{-1}} \ . \tag{3.4b}$$

Estas divisões são realizadas por adições e subtrações dos coeficientes de $f_1'(z)$ e $f_2'(z)$. Os polinômios $G_1(z)$ e $G_2(z)$ são simétricos, conforme (3.5) abaixo. Para o cálculo rápido das raízes, G_1(z) e G_2(z) são transformados em polinômios de Chebyshev [21][22]. Para tanto os polinômios são divididos por $e^{-j\omega P/2}$, onde P é a ordem do preditor (no caso 10). As raízes dos polinômios de Chebyshev estão localizadas no intervalo [-1, +1] no semicírculo superior. Os polinômios $G_1(z)$ e $G_2(z)$ se expressam por:

$$\begin{aligned} G_1(z) &= 1 + g_1(1)z^{-1} + \ldots + g_1(M_1)z^{-M1} + \ldots + g_1(1)z^{-(2M_1-1)} + z^{-2M_1} \\ G_2(z) &= 1 + g_2(1)z^{-1} + \ldots + g_2(M_1)z^{-M1} + \ldots + g_2(1)z^{-(2M_1-1)} + z^{-2M_1} \end{aligned} \tag{3.5}$$

onde $M_1 = \frac{P}{2}$. Fazendo-se a substituição $z = e^{j\omega}$, vem

$$G_1(e^{j\omega}) = e^{-j\omega M_1} G_1'(\omega) \tag{3.6a}$$

$$G_2(e^{j\omega}) = e^{-j\omega M_1} G_2'(\omega), \tag{3.6b}$$

onde

$$G_1'(e^{j\omega}) = 2\cos(M_1\omega) + 2g_1(1)\cos(M_1-1)\omega + \ldots + 2g_1(M_1-1)\cos\omega + g_1(M_1) \text{ e} \tag{3.7a}$$

$$G_2'(e^{j\omega}) = 2\cos(M_1\omega) + 2g_2(1)\cos(M_1-1)\omega + \ldots + 2g_2(M_1-1)\cos\omega + g_2(M_1) \tag{3.7b}$$

são polinômios de Chebyshev.

Consideremos o mapeamento $x = \cos\omega$ e $\cos m\omega = T_m(x)$, onde $T_m(x)$ é um polinômio Chebyshev em x de m-ésima ordem. Sabe-se que os polinômios de Chebyshev satisfazem a recursão:

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x), \tag{3.8}$$

com condições iniciais de $T_0(x) = 1 \text{ e } T_1(x) = x$. A série expandida em coseno pode ser expressa em termos de polinômios de Chebyshev:

$$G_1'(x) = 2T_{M_1}(x) + 2g_1(1)T_{M_1-1}(x) + \ldots + 2g_1(M_1 - 1)T_1(x) + g_1(M_1) \tag{3.9a}$$

$$G_2'(x) = 2T_{M_1}(x) + 2g_2(1)T_{M_1-1}(x) + \ldots + 2g_2(M_1 - 1)T_1(x) + g_2(M_1) \tag{3.9b}$$

As raízes de $G_1'(z) \text{ e } G_2'(z)$ são determinadas e correspondem às raízes LSF's dadas por $\omega_i = \arccos x_i$.

É utilizado o método da bipartição de Newton para encontrar os valores das raízes. Calculam-se os valores destas funções em 60 pontos igualmente espaçados entre [-1, +1] e verifica-se a mudança de sinal. A ocorrência de mudança de sinal significa que há uma raiz neste intervalo. Então o intervalo é dividido ao meio, verifica-se em qual dos segmentos se encontra a mudança de sinal e divide-se este segmento por 2. Esta partição é realizada por 4 vezes. Então faz-se a interpolação linear para melhorar a estimativa da raiz. Para o cálculo do valor da função usa-se a recursão dos polinômios de Chebyshev.

A relação entre as raízes q_i encontradas por este método e as LSF's é

$$f_i = \frac{f_s}{2\pi}\arccos(q_i) \quad i = 1,\ldots,10 \tag{3.10}$$

onde : f_i é a linha espectral de freqüências em Hz limitada ao intervalo [0, 4000];

f_s = 8000 é a freqüência de amostragem.

Os valores destas raízes variam em torno de um valor médio.

3.5 Quantização das LSF

As LSF's têm valores médios bem definidos. Para uma boa quantização, o número de

vetores do dicionário deve ser adequado. Mas quanto maior o número de vetores, maior o tempo de busca. Se o vetor alvo for particionado, pode-se utilizar dicionários menores com

grande número de combinações. Assim é utilizada a quantização vetorial partida (*Split Vector Quantization SVQ*). A Figura 3.6 ilustra o diagrama de blocos de quantização. A redução da faixa dinâmica dos valores favorece a qualidade do quantizador.

Primeiro retira-se do sinal seu valor médio, fazendo com que os valores das freqüências (LSF) variem em torno de zero. Utiliza-se um preditor MA de primeira ordem para calcular o resíduo (erro de quantização). O sinal predito é obtido a partir do sinal quantizado.

O filtro preditor é dado por

$$p_j(n) = \alpha_j \hat{r}_j(n-1), \tag{3.11}$$

onde : α_j é o coeficiente de predição do j-ésimo elemento do vetor LSF;

$r_j(n-1)$ é o resíduo quantizado do quadro anterior.

O resíduo do quadro corrente é dado por

$$r(n) = z(n) - p(n). \tag{3.12}$$

A técnica de quantização vetorial partida consiste em dividir o vetor de freqüências em três subvetores de tamanhos 3, 3 e 4. Cada subvetor é quantizado separadamente com 8, 9 e 9 bits respectivamente.

Para um vetor LSF f de entrada, e um vetor do dicionário de índice k, $\hat{f}^k$, a quantização é realizada buscando minimizar

$$E_k = \sum_{i=1}^{10} \left(f_i w_i - \hat{f}_i^k \; w_i \right)^2 \tag{3.13}$$

Os fatores de peso w_i , i = 1,...,10, são dados por

$$w_i = \begin{cases} 3{,}347 - \dfrac{1{,}547}{450} d_i, & para\ d_i < 450 \\ 1{,}8 - \dfrac{0{,}8}{1050}(d_i - 450), & caso\ \text{contrário} \end{cases} \tag{3.14}$$

onde $d_i = f_{i+1} - f_{i-1}$, com f_0 =0 e f_{11} = 4000 Hz.

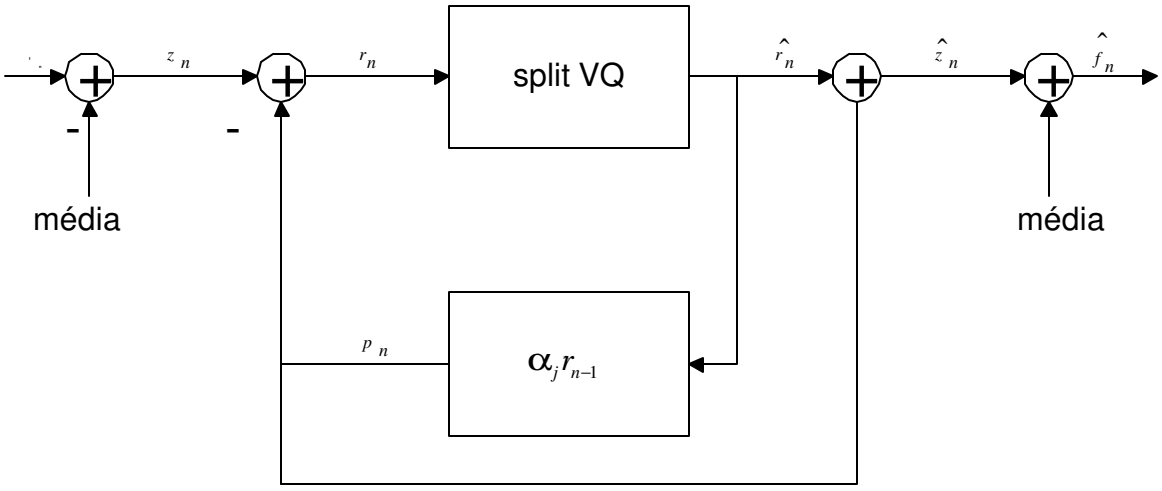

Figura 3.4 : diagrama em blocos do quantizador

3.6 Interpolação dos LSPs

As análises dos próximos blocos são feitas a cada subquadro para se ter um refinamento dos parâmetros do sinal. Neste bloco calculamos as freqüências referentes aos subquadros. São quatro subquadros com as freqüências dadas pela seguinte interpolação:

$$\begin{aligned} q_1^{(n)} &= 0{,}75q_4^{(n-1)} + 0{,}25q_4^{(n)} \\ q_2^{(n)} &= 0{,}50q_4^{(n-1)} + 0{,}50q_4^{(n)} \\ q_3^{(n)} &= 0{,}25q_4^{(n-1)} + 0{,}75q_4^{(n)} \\ q_4^{(n)} &= q_4^{(n)} \end{aligned} \tag{3.15}$$

onde $q_i^{(n)}$ e $q_i^{(n-1)}$ são os vetores LSP's do quadro atual e anterior, respectivamente.

A interpolação é realizada nos vetores LSP quantizados e não quantizados, originando respectivamente $\hat{A}(z)$ e $A(z)$, conforme a Fig. 3.1. Note que as freqüências estão sendo interpoladas, fazendo uma progressão do quadro anterior ao atual. No codificador anterior interpolava-se, de forma análoga a este, os coeficientes autoregressivos do preditor linear.

3.7 Filtro perceptual

Este filtro é utilizado para mascarar os sinais de erro pelos picos de formantes de forma semelhante ao utilizado no VSELP. O sinal é atenuado nas freqüências onde o erro é menos importante e amplificado nas freqüências onde o erro é mais importante.

O filtro é dado por

$$W(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)} \tag{3.16}$$

onde A(z) é o filtro preditor não quantizado (vide equação 1.1), $\gamma_1 = 0{,}94$ e $\gamma_2 = 0{,}6$. O sinal de fala $s(n)$ é filtrado resultando no sinal ponderado $s_w(n)$, que será utilizado no cálculo do sinal alvo no processo de otimização.

3.8 Resposta ao Impulso

A resposta ao impulso $h(n)$, do filtro de síntese perceptual, é computada a cada subquadro. Esta resposta ao impulso é necessária para a procura dos dicionários adaptativo e fixo. A resposta ao impulso $h(n)$ é computada filtrando os coeficientes do numerador $A(z/\gamma_1)$ através de dois filtros $\frac{1}{\hat{A}(z)}$ e $\frac{1}{A(z/\gamma_2)}$.

O filtro de síntese perceptual é dado por

$$H(z)W(z) = \frac{A(z/\gamma_1)}{\hat{A}(z)A(z/\gamma_2)} \tag{3.17}$$

3.9 Cálculo do sinal alvo

O sinal alvo é computado subtraindo a resposta a entrada zero do filtro de síntese perceptual do sinal de fala perceptual $s_w(n)$.

Neste codificador o sinal alvo é computado filtrando o sinal LP residual $r(n)$ através da combinação do filtro de síntese e do filtro perceptual. Depois de determinar a excitação do subquadro, os estados iniciais destes filtros são atualizados.

3.10 Cálculo do valor de *Pitch*

Para que a síntese do sinal no decodificador tenha boa qualidade é necessário que se transmita mais parâmetros que descrevam o sinal. Estes formarão o sinal de excitação para o filtro de síntese.

O cálculo de atraso de *pitch* é realizado em duas etapas. Primeiro, estima-se em malha aberta (*open-loop*) o valor do atraso, e depois, faz-se o refinamento em malha fechada (*closed-loop*). Esta técnica é utilizada para que o cálculo em malha fechada seja realizado de forma não exaustiva visando a redução do esforço computacional.

3.10.1 Análise em malha aberta

Esta análise é feita a partir do sinal de fala filtrado pelo filtro perceptual, sinal $s_w(n)$.

Para encontrar o período de *pitch,* calcula-se a correlação do sinal. Quando esta for máxima, tem-se a estimativa de *pitch*. Este cálculo é feito duas vezes por quadro, ou seja, a cada 10 ms.

Para períodos pequenos de *pitch* (voz feminina e infantil), um pequeno erro na estimativa do valor do período se acumula e torna-se perceptível, deteriorando a qualidade do sinal. Para minimizar este efeito a correlação:

$$C_k = \sum_{n=k}^{79} s_w(n) s_w(n-k) \tag{3.18}$$

é calculada em três faixas, k = 20...39, 40...79, e 80...143. Encontra-se um valor máximo para a correlação em cada faixa (cujo atraso respectivo é k_i) e normalizam-se estes valores dividindo-os por

$$\sqrt{\sum_{n=k_i}^{79} s_w^2(n-k_i)}\,, \tag{3.19}$$

onde i = 1,2,3, obtendo-se os máximos normalizados R_1, R_2 e R_3. O atraso ótimo T_{op} é selecionado entre as três correlações normalizadas. Nesta busca, os atrasos na menor faixa são favorecidos por meio da seguinte regra: o valor máximo normalizado R_i é selecionado se $R_i > 0{,}85R_{i+1}$. O valor de atraso ótimo em amostras, será $T_{op} = k_i$, com i vencedor.

3.10.2 Análise em malha fechada

Nesta análise os parâmetros encontrados são o atraso refinado e o ganho de *pitch*.

Com a estimativa inicial do valor de *pitch*, obtida pela análise em malha aberta, faz-se o refinamento pelo cálculo em malha fechada. A procura do atraso de *pitch* é feita buscando minimizar o erro quadrático médio entre o sinal original e o sinal sintetizado. É realizado maximizando a correlação

$$T_k = \frac{\sum_{n=0}^{39} x(n)\, y_k(n)}{\sqrt{\sum_{n=0}^{39} y_k(n)\, y_k(n)}} \tag{3.20}$$

onde $x(n)$ é o sinal alvo (sinal original menos a contribuição do filtro de síntese) e $y_k(n)$ é a excitação do subquadro anterior filtrada no atraso k (convolução da excitação do subquadro anterior com $h(n)$).

Através do cálculo da correlação normalizada acima, obtém-se o valor inteiro do atraso de *pitch*. Para o primeiro e terceiro subquadros é calculado para $T_{op} \pm 3$. Nos outros subquadros (segundo e quarto) a análise é feita em torno de valores inteiros do atraso de *pitch* do subquadro anterior. Nestes subquadros, segundo e quarto, a análise é realizada com resolução de 1/3 na faixa [T_1 – 5,66; T_1 + 4,66], onde T_1 é o valor inteiro do atraso de *pitch* do subquadro anterior.

Nos primeiro e terceiro subquadros é encontrado o atraso fracionário, com resolução de 1/3, na faixa [19,33; 84,66] e somente inteiro na faixa [85; 143].

O cálculo da parte fracionária do atraso de *pitch* é realizado interpolando a correlação na equação (3.20). É calculado na faixa de -2/3 a 2/3, com resolução de 1/3, em torno do valor inteiro.

O vetor v(n) do dicionário adaptativo, é resultado da interpolação do sinal de excitação do subquadro anterior observando a parte fracionária do *pitch* (fase).

Estas interpolações são realizadas por filtros FIR definidas no padrão como funções sinc janeladas por Hamming. Para a interpolação da equação (3.20) a função sinc é truncada em ±11, e para a interpolação do sinal de excitação $u(n)$, a função sinc é truncada em ±29.

O ganho de *pitch* é dado por

$$g_p = \frac{\sum_{n=0}^{39} x(n)\,y(n)}{\sqrt{\sum_{n=0}^{39} y(n)\,y(n)}}. \tag{3.21}$$

3.11 Dicionário Algébrico

O sinal de excitação é computado como sendo o sinal de fala menos as contribuições dos dicionários (fixo e adaptativo). Este sinal deve ser transmitido para ser feita a reconstrução do sinal. Assim

$$x_2(n) = x(n) - g_p\,y(n) \tag{3.22}$$

onde aqui $y(n) = v(n) * h(n)$ é o vetor do dicionário adaptativo filtrado e g_p é ganho do dicionário adaptativo (não quantizado).

Uma nova estrutura é utilizada (Figura 3.5), na qual a inovação é gerada a partir de um dicionário algébrico $\{a_k\}$ e uma matriz para dar forma ao sinal (*F*) [23]. O vetor excitação é dado por:

$$c_k = Fa_k \tag{3.23}$$

O dicionário algébrico é composto de vetores esparsos, apenas quatro valores diferentes de zero, com valores +1 ou -1. A matriz *F* realiza uma transformação no vetor de excitação no domínio da freqüência na qual suas energias são concentradas nas bandas de freqüências mais importantes. A matriz *F* é função do modelo LPC e melhora as regiões de formantes na fala reconstruída, geralmente obtida com pós-filtros. Esta matriz é uma matriz Toeplitz triangular inferior construída a partir da resposta ao impulso do filtro :

$$F(z) = (1 - \mu z^{-1})\frac{A(z/\gamma_1)}{A(z/\gamma_2)}, \tag{3.24}$$

onde $A(z)$ é o filtro LPC inverso, γ_1 e γ_2 são constantes, e μ é o fator que controla a inclinação espectral e varia a cada quadro de excitação.

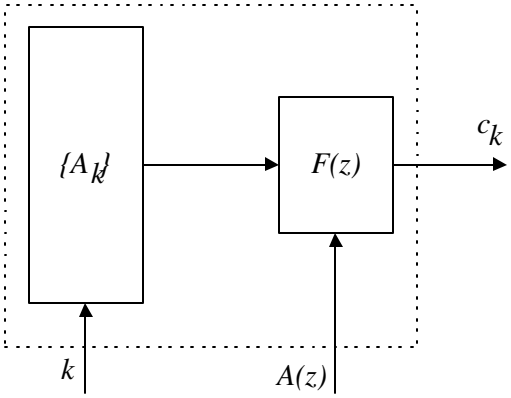

Figura 3.5 – Estrutura geradora do vetor de excitação

As 40 posições no subquadro são divididas em 4 faixas, as quais contem apenas um pulso, dadas pela Tabela 3.1.

Pulso	Posições
i_0	0, 5, 10, 15, 20, 25, 30, 35
i_1	1, 6, 11, 16, 21, 26, 31, 36
i_2	2, 7, 12, 17, 22, 27, 32, 37
i_3	3, 8, 13, 18, 23, 28, 33, 38 4, 9,14, 19, 24, 29, 34, 39

Tabela 3.1: Posições para os pulsos no dicionário algébrico.

O vetor c_k que representará o sinal de excitação deverá minimizar o erro quadrático

$$E = \left\| x_2 - gHc_k \right\|^2 . \tag{3.25}$$

O vetor ótimo do dicionário é determinado maximizando o termo:

$$T_k = \frac{(d^t c_k)^2}{c_k^t \Phi c_k} \tag{3.26}$$

sendo Φ uma matriz simétrica dada por

$$\Phi(i, j) = \sum_{n=j}^{39} h(n-i)h(n-j), \quad i = 0,...,39 \quad j = i,...,39, \tag{3.27}$$

e $d = H^t x_2$, onde H é definida como uma matriz convolução Toeplitz triangular inferior com diagonal $h(0)$ e diagonais inferiores $h(1)$, ..., $h(39)$.

$$H = \begin{bmatrix} h(0) & 0 & 0 & \cdots \\ h(1) & h(0) & 0 & \\ h(2) & h(1) & h(0) & \\ \vdots & & & \ddots \end{bmatrix} \tag{3.28}$$

O vetor d e a matriz Φ são computados antes da busca no dicionário.

A estrutura algébrica torna o procedimento de busca muito rápido devido ao vetor de inovações conter apenas 4 pulsos diferentes de zero. A correlação no numerador de T_k é dada por

$$C = \sum_{i=0}^{N_p-1} a_i d(m_i), \tag{3.29}$$

onde m_i é a posição do i-ésimo pulso, a_i é sua amplitude e N_p é o número de pulsos. A energia no denominador é dada por

$$E = \sum_{i=0}^{N_p-1} \Phi(m_i, m_i) + 2 \sum_{i=0}^{N_p-2} \sum_{j=i+1}^{N_p-1} a_i a_j \Phi(m_i, m_j). \tag{3.30}$$

Para simplificar o processo de busca, as amplitudes dos pulsos são predeterminadas. Isto é feito igualando a amplitude do pulso em uma certa posição ao sinal de $d(n)$ na mesma posição. É implementado da seguinte forma: primeiro o sinal $d(n)$ é decomposto em duas partes, seu valor absoluto $|d(n)|$ e seu sinal $sign[d(n)]$; em seguida, a matriz Φ é modificada para incluir a informação de sinal:

$$\Phi'(i, j) = sign[d(i)]sign[d(j)]\Phi(i, j), \quad i = 0,...,39, \quad j = i+1,...,39. \tag{3.31}$$

A correlação agora é dada por

$$C = \sum_{i=0}^{N_p-1} | d(m_i) | \tag{3.32}$$

e a energia por

$$E = \sum_{i=0}^{N_p-1} \Phi'(m_i, m_i) + 2 \sum_{i=0}^{N_p-2} \sum_{j=i+1}^{N_p-1} \Phi'(m_i, m_j). \tag{3.33}$$

A busca do melhor vetor é feita de forma sub-ótima não exaustiva, inserindo uma posição a cada vez.

3.12 Quantização dos ganhos

O ganho do dicionário adaptativo (ganho de *pitch*) e o ganho do dicionário algébrico são quantizados juntos usando-se um dicionário de 7 bits.

O ganho do dicionário fixo (algébrico) é calculado usando predição MA com coeficientes fixos. A predição de 4ª ordem é realizada a partir do erro de predição para cálculo da energia da inovação. A energia predita é dada por

$$\tilde{E}(n) = \sum_{i=1}^{4} b_i \hat{R}(n-i) \tag{3.34}$$

Sendo *E(n)* a energia de inovação com média removida no subquadro *n* dada por (expressa em dB)

$$E(n) = 10\log\left(\frac{1}{N} g_c^2 \sum_{i=0}^{N-1} c^2(i)\right) - \overline{E}\,, \tag{3.35}$$

onde $N = 40$ é o tamanho do subquadro, $c(i)$ é a excitação do dicionário algébrico e $\overline{E} = 36$ dB é a média da energia de inovação de longo termo. A energia predita é utilizada para calcular o ganho predito do dicionário fixo g_c' como na equação acima (substituindo-se g_c por g_c' e *E(n)* por $\tilde{E}(n)$). A energia média de inovação de subquadro é encontrada por

$$E_s = 10\log\left(\frac{1}{N} \sum_{i=0}^{N-1} c^2(i)\right) \tag{3.36}$$

e o ganho predito é encontrado por

$$g'_c = 10^{0.05(\tilde{E}(n) + \overline{E} - E_s)} \tag{3.37}$$

O fator de correção entre o ganho g_c e o estimado g'_c é dado por

$$\gamma = \frac{g_c}{g'_c} \tag{3.38}$$

O erro de predição é dado por

$$R(n) = E(n) - \tilde{E}(n) = 20\log(\gamma) \tag{3.39}$$

O ganho de *pitch*, g_p, e o fator de correção γ são quantizados juntos utilizando um dicionário de 7 bits.

3.13 Atualização de memória

A atualização dos estados dos filtros de síntese e perceptual são necessários para o cálculo do sinal alvo do próximo subquadro.

Após os ganhos terem sido quantizados, o sinal de excitação no subquadro presente é dado por

$$u(n) = \hat{g}_p v(n) + \hat{g}_c c(n) \tag{3.40}$$

onde $\hat{g}_p$ e $\hat{g}_c$ são os ganhos quantizados dos dicionários, $v(n)$ é o vetor do dicionário adaptativo e $c(n)$ é o vetor do dicionário fixo. Os estados podem ser atualizados pela filtragem de 40 amostras do sinal $r(n)$ - $u(n)$ através dos filtros de síntese e perceptual (onde $r(n)$ é o resíduo de predição linear). Este método requer três filtragens. O padrão deste codificador sugere que se faça de uma maneira mais simples utilizando apenas uma filtragem.

O sinal de fala de síntese local ($\hat{s}(n)$), é computado filtrando o sinal de excitação pelo filtro de síntese. A saída do filtro devido à entrada $r(n) - u(n)$ equivale ao sinal de erro de saída $e(n) = s(n) - \hat{s}(n)$. Então os estados do filtro de síntese são dados por $e(n)$, $n=30,...,39$. Para atualizar os estados do filtro perceptual, o sinal $e(n)$ é filtrado por este filtro encontrando o erro perceptual $e_w(n)$. Este erro pode ser encontrado através de

$$e_w(n) = x(n) - \hat{g}_p y(n) - \hat{g}_c z(n), \tag{3.41}$$

onde $x(n)$ é o vetor alvo, $y(n)$ é o vetor do dicionário adaptativo filtrado e $z(n)$ é o vetor do dicionário fixo filtrado.

3.14 Decodificador

A decodificação é realizada a partir dos índices recebidos. Na Figura 3.6 temos ilustrado o diagrama em blocos do decodificador. O processo de decodificação é realizado como a seguir.

Os índices da LSF recebidos são utilizados para reconstruir o vetor LSF quantizado. A interpolação descrita em 3.6 é realizada para obter os quatro vetores LSF

correspondentes aos subquadros. Cada vetor LSF interpolado é convertido em coeficientes LPC, os quais são usados para reconstruir o sinal de fala no subquadro.

A cada subquadro é realizado :

1. Decodificação do vetor do dicionário adaptativo: o vetor do dicionário adaptativo $v(n)$ é encontrado interpolando a excitação passada $u(n)$ (no atraso de *pitch*), usando o filtro FIR citado na seção 3.10.2 para realizar esta mesma operação.
2. Decodificação do vetor algébrico: as posições e amplitudes dos pulsos de excitação são extraídas do índice recebido. De posse destes dados, encontra-se o dicionário algébrico $c(n)$ como descrito na seção 3.11.

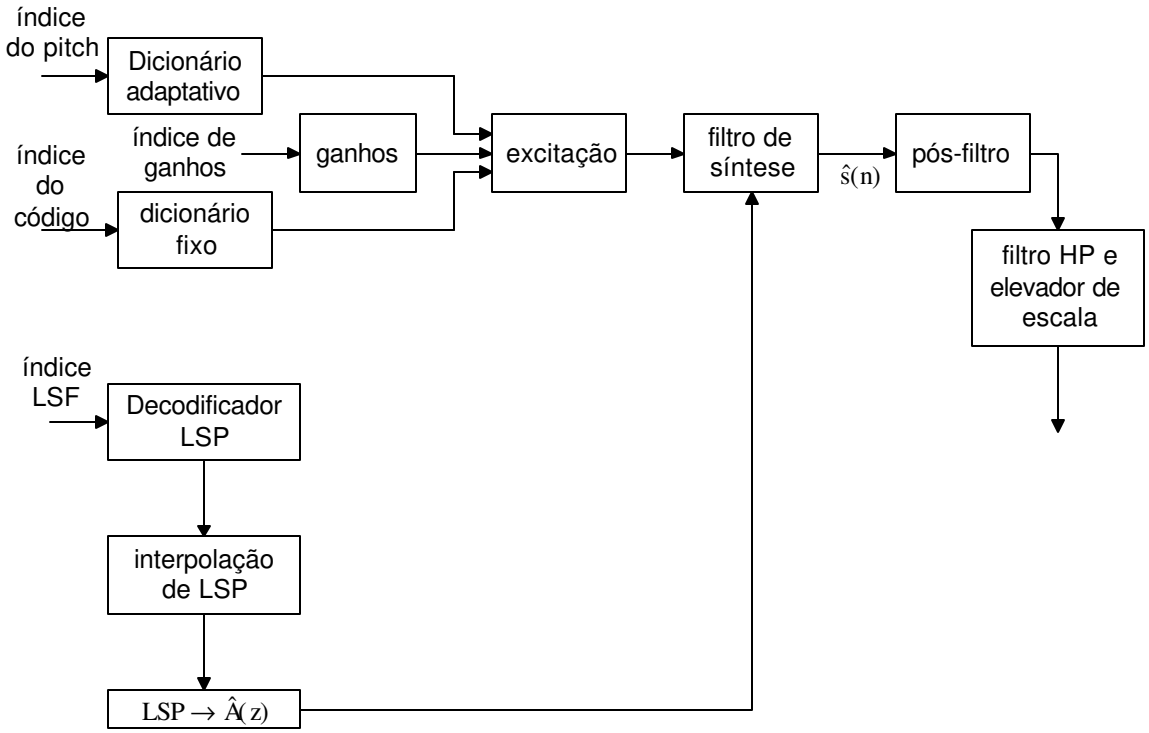

Figura 3.6 – Diagrama em blocos do decodificador EFR.

3. Decodificação dos ganhos: o índice nos fornece o ganho do dicionário adaptativo e o fator de correção $\hat{\gamma}$. O ganho do dicionário fixo é encontrado pelo método descrito na seção 3.12.
4. Reconstrução do sinal de fala: através da excitação total

$$u(n)\hat{g}_p v(n) + \hat{g}_c c(n). \tag{3.42}$$

Antes de sintetizar o sinal de fala, é feito um pós-processamento da excitação enfatizando a contribuição do vetor do dicionário adaptativo:

$$\hat{u}(n) = \begin{cases} u(n) + 0.5\hat{\beta}_c \hat{g}_p v(n), & \hat{g}_p > 0{,}5 \\ u(n), & \hat{g}_p \le 0{,}5 \end{cases} \tag{3.43}$$

onde $\hat{\beta}_c$ é o ganho de *pitch* decodificado e $\hat{g}_p$ é limitado por [0,0; 0,8]. Um controle adaptativo de ganho (AGC) é usado para compensar a diferença entre $u(n)$ e $\hat{u}(n)$. O fator de escalonamento de ganho é dado por :

$$\gamma = \begin{cases} \sqrt{\dfrac{\sum_{n=0}^{39} u^2(n)}{\sum_{n=0}^{39} \hat{u}^2(n)}}, & \hat{g}_p > 0{,}5 \\ 1{,}0 & \hat{g}_p \le 0{,}5. \end{cases} \tag{3.44}$$

O sinal de excitação torna-se

$$\hat{u}'(n) = \hat{u}(n)\gamma . \tag{3.45}$$

O sinal de fala reconstruído para um subquadro é

$$\hat{s}(n) = \hat{u}'(n) - \sum_{i=1}^{10} \hat{a}_i \hat{s}(n-i), \tag{3.46}$$

onde $\hat{a}_i$ são os coeficientes do filtro LPC interpolados.

O sinal de fala sintetizado $\hat{s}(n)$ é então passado por um pós-filtro adaptativo, o qual é descrito na próxima seção.

3.14.1 Pós-processamento

O pós-processamento consiste de três funções: pós-filtragem adaptativa, filtragem passa-altas e elevação da faixa dinâmica.

O pós-filtro adaptativo é formado por dois filtros em cascata : um pós-filtro de formantes e um filtro de compensação de inclinação espectral. O pós-filtro é atualizado a cada subquadro.

O pós-filtro de formantes é dado por:

$$H_f(z) = \frac{\hat{A}(z/\gamma_n)}{\hat{A}(z/\gamma_d)}, \tag{3.47}$$

onde os fatores γ_n e γ_d controlam a pós-filtragem de formante. Os valores dos fatores de pós-filtro adaptativo são $\gamma_n = 0{,}55$ e $\gamma_d = 0{,}7$.

O filtro $H_t(z)$ compensa a inclinação provocada pelo pós-filtro $H_f(z)$ e é dado por:

$$H_t(z) = (1 - \mu z^{-1}), \tag{3.48}$$

onde $\mu = \gamma_t k_1$ é o fator de inclinação. O fator $\gamma_t = 0{,}8$ e o fator k_1 é o primeiro coeficiente de reflexão calculado à partir da resposta ao impulso truncada $h_f(n)$ do filtro de formantes (3.47). Este fator k_1 é dado por:

$$k_1 = \frac{r_h(1)}{r_h(0)}; \tag{3.49}$$

sendo $r_h(0)$ e $r_h(1)$ os primeiros coeficientes da autocorrelação.

O processo de pós-filtragem é realizado da seguinte forma: primeiro o sinal de fala sintetizado $\hat{s}(n)$ é filtrado por $\hat{A}(z/\gamma_n)$ para produzir o sinal residual $\hat{r}(n)$. O sinal $\hat{r}(n)$ é filtrado pelo filtro de síntese $1/\hat{A}(z/\gamma_d)$. Então este sinal filtrado é passado pelo filtro de compensação de inclinação $H_t(z)$, resultando no sinal de fala pós-filtrado $s_f(n)$.

A diferença de ganho entre o sinal de fala sintetizado $\hat{s}(n)$ e o sinal de fala pós-filtrado $s_f(n)$ é compensada por um controle adaptativo de ganho (AGC). O fator de escala γ para cada subquadro é dado por:

$$\gamma = \sqrt{\frac{\sum_{n=0}^{39} \hat{s}^2(n)}{\sum_{n=0}^{39} s_f^2(n)}} \tag{3.50}$$

O sinal de fala reconstruído $s'(n)$ é encontrado como

$$s'(n) = \beta(n) s_f(n), \tag{3.51}$$

onde $\beta(n)$ é atualizado amostra a amostra pela equação

$$\beta(n) = \alpha\beta(n-1) + (1-\alpha)\gamma \tag{3.52}$$

onde α é um fator do AGC igual a 0,9.

Mais dois pós-processamentos são realizados: filtragem passa altas e elevação de escala do sinal. O filtro passa-altas, ilustrado na Figura 3.7, serve de precaução contra componentes indesejáveis de baixa freqüência. A freqüência de corte de – 3dB deste filtro é de 80 Hz e a atenuação em 60 Hz é de –10 dB. A elevação de escala consiste em multiplicar o sinal por fator de 2 para compensar a diminuição de escala feita no estágio de pré-processamento.

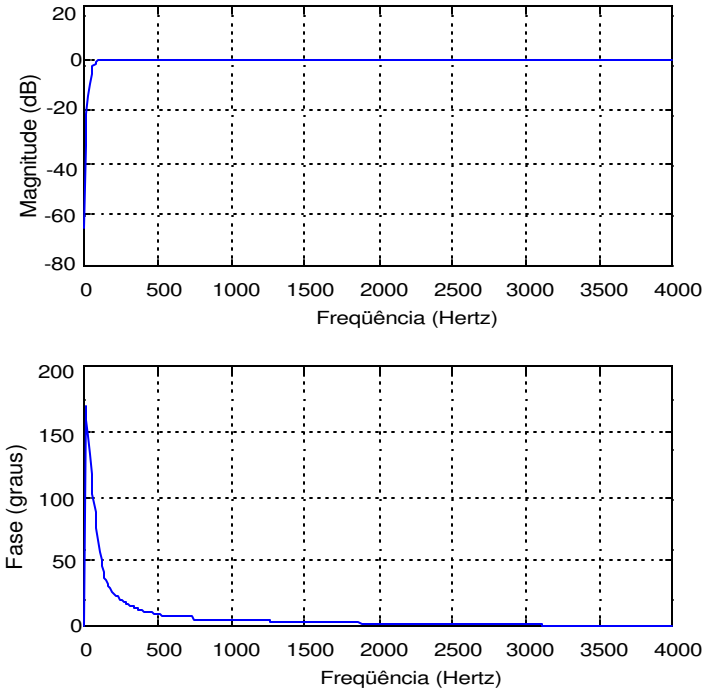

Figura 3.7: Resposta em magnitude e em fase do filtro de pós-processamento.

Capítulo 4

A Medida Perceptual de Qualidade de Voz

Este capítulo apresenta a medida PSQM - *Perceptual Speech Quality Measure*, utilizada como medida subjetiva na avaliação dos codificadores.

4.1 Introdução

Para se avaliar a qualidade de codificadores de fala são utilizadas medidas subjetivas e objetivas. A medida objetiva mais utilizada é a *Signal to Noise Ratio* (SNR). Este tipo de medida é freqüentemente utilizado por ser de fácil implementação e por ser de baixa complexidade computacional. Mas esta medida não traduz a qualidade perceptual do sinal. Portanto é ineficiente para os codificadores com baixas taxas de bits.

As medidas subjetivas são concebidas visando a avaliação da qualidade perceptual. Uma destas medidas é a MOS - *Mean Opinion Score.* Ela é realizada colhendo opinião de um grande número de pessoas, as quais geralmente não são profissionais na avaliação de qualidade de fala. Esta medida exige tempo e recursos para ser realizada, como avaliadores e locais apropriados, escassos no meio acadêmico.

Por estes motivos há uma procura por um método objetivo capaz de caracterizar a qualidade subjetiva do sinal de fala. Neste sentido a ITU-T - *International Telecommunication Union - Telecommunication Sector* comparou métodos computacionais que visam obter informação perceptual do sinal de fala e elegeu o método chamado de

PSQM - *Perceptual Speech Quality Measure*. Este estimador de qualidade subjetiva mostrou boa independência de línguas, de locutores ou de codificadores. A recomendação P.861 [10], que define o método PSQM, é uma ferramenta poderosa para estimar a qualidade subjetiva dos codificadores de fala.

Devido à característica de ter alta correlação com a medida MOS e ser indicado pela ITU-T, optamos por utilizar esta medida em nosso trabalho. Adicionalmente a medida PSQM mostra-se muito adequada em diversas outras situações: por exemplo, verificação de qualidade da qualidade de um sistema móvel celular em operação.

O algoritmo descrito a seguir foi implementado a partir das especificações da recomendação P-861 e validada pelos sinais de teste que acompanham a mesma.

4.2 A medida de qualidade PSQM

A medida baseia-se na comparação do sinal fonte com o sinal decodificado. O objetivo do método é simular a percepção sonora em situações reais. O método PSQM simula experimentos para julgar a qualidade dos codificadores de fala. O método PSQM busca representar a percepção humana e seus processos de julgamento; assim, entrada e saída com diferenças inaudíveis devem receber a mesma pontuação PSQM.

A Figura 4.1 ilustra o conceito básico da medida PSQM. Na Figura 4.2 temos o diagrama em blocos do algoritmo PSQM. Este algoritmo é detalhado a frente.

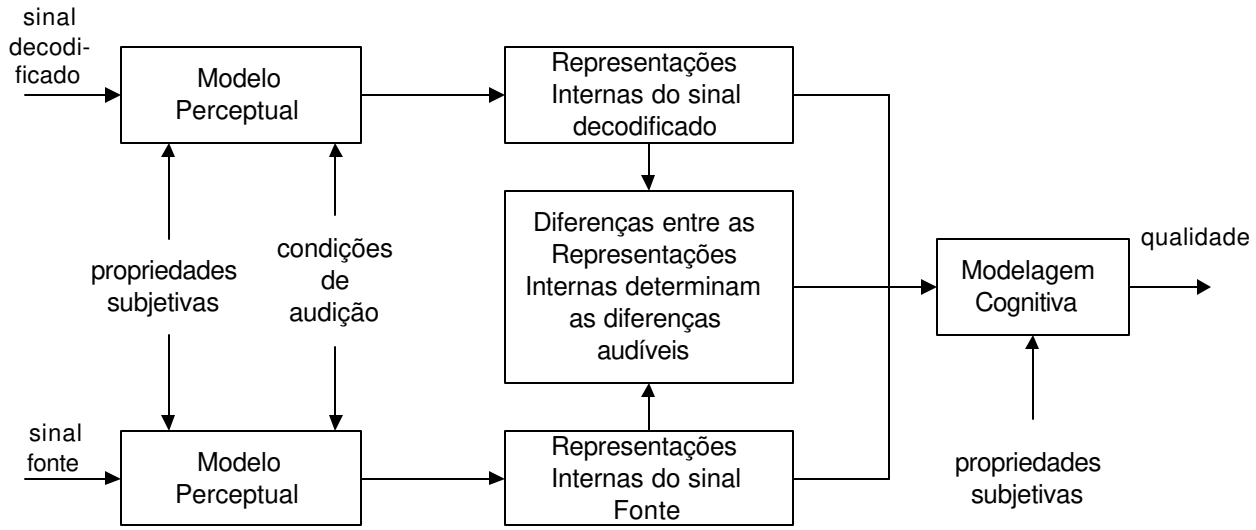

Figura 4.1 Conceito da medida PSQM

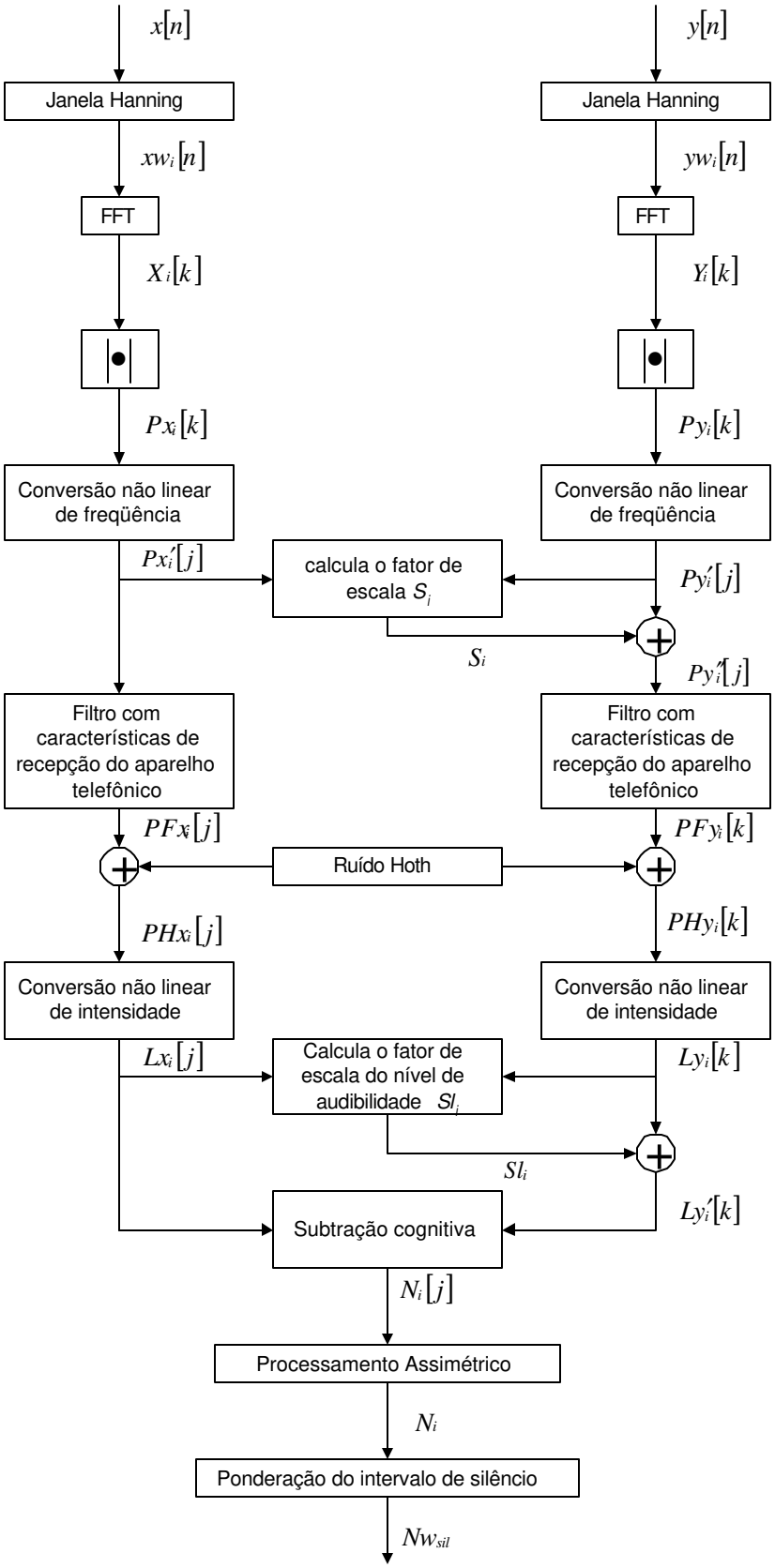

Figura 4.2 Diagrama em blocos do algoritmo da medida PSQM

Os sinais original e decodificado são mapeados em representações psicofísicas baseadas em estudos do aparelho auditivo. As diferenças entre estas representações são analisadas e a modelagem cognitiva é aplicada.

Como os sinais serão comparados, faz-se necessário alinhá-los no tempo e calibrar seus níveis devido ao ganho no codec (codificador e decodificador). Após estes ajustes iniciais os sinais estão prontos para o cálculo da medida PSQM ilustrada na Figura 4.2. Realiza-se um mapeamento do domínio do tempo para o domínio tempo-freqüência. Este mapeamento é implementado através de uma transformada de *Fourier* de curto-termo. Para isto, utiliza-se o janelamento de Hanning.

Na etapa seguinte, faz-se a conversão da escala linear em Hertz para a escala não linear de bandas críticas, o que resulta numa representação chamada de densidade perceptual espectral de potência. Esta densidade do sinal codificado é escalonada a cada quadro. Então realiza-se a filtragem dos sinais na banda telefônica e soma-se o ruído *Hoth* para ajustar às características de recepção do sinal do aparelho telefônico.

Na modelagem cognitiva o nível de audibilidade é escalonado e um processamento assimétrico é feito. O processamento assimétrico observa a ocorrência de quadros de silêncio, que não influem na qualidade do sinal, pois qualquer diferença entre os sinais durante um quadro de silêncio degrada a qualidade.

4.3 Iniciação

Por se tratar de medida comparativa entre sinais é necessário realizar alguns ajustes antes de se calcular o valor PSQM . São eles :

- alinhamento no tempo;
- escalonamento global para compensar o ganho do sistema;
- calibração global para fixar o volume do sinal de fala.

4.3.1 Alinhamento no tempo

O primeiro ajuste a ser realizado é alinhar no tempo os sinais fonte $x[n]$ e codificado $y_c[n]$. Os codificadores atrasam o sinal e se os sinais não estão alinhados, a PSQM

torna-se incoerente.

Se o atraso do sinal codificado em relação ao sinal fonte não é conhecido, então teoricamente, uma estimativa é dada pelo valor máximo da correlação cruzada entre os sinais.

A correlação cruzada é calculada no trecho que contém atividade sonora, ou seja, a elocução. Para determinar o trecho de voz ativa, faz-se a detecção de início e fim da elocução apenas para o sinal fonte, descartando o ruído existente no arquivo de fala. É considerada como primeira amostra da elocução no arquivo de fala aquela na qual sua magnitude somada às magnitude das quatro amostras anteriores totalizam 200 ou mais (está subentendido a faixa dinâmica do PCM linearizado, lei A). O mesmo conceito é utilizado para encontrar a amostra final, a qual é somada às magnitudes das quatro amostras sucessoras e deve totalizar 200 ou mais.

Após alinhar os sinais, deve-se compensar o ganho do codificador presente no sinal decodificado $y_c[n]$. O fator de escalonamento é dado por :

$$S_{global} = \sqrt{\frac{\sum_{início}^{fim} x^2[n]}{\sum_{início}^{fim} {y_c}^2[n]}} \tag{4.1}$$

O sinal decodificado $y_c[n]$ é multiplicado por S_{global}, resultando em $y[n]$.

Para garantir que a precisão da medida objetiva seja ótima, é necessário realizar uma calibração entre o nível de audição e o nível de audibilidade (*loudness*) comprimido. Os detalhes deste
escalonamento estão descritos na recomendação.

4.4 Mapeamento tempo-freqüência

Ambos os sinais, fonte $x[n]$ e decodificado $y[n]$, são janelados usando uma janela de Hanning, definida como segue:

$$w[n] = 0.5\left(1 - \cos\left(\frac{2\pi n}{N_f - 1}\right)\right) \quad para \;\; 0 \le n \le N_f\text{-}1 \tag{4.2}$$

onde N_f é o número de amostras por quadro. No nosso caso a freqüência de amostragem é de 16 kHz e N_f é 512.

Após o janelamento, calcula-se a *Fast Fourier Transforms* (*FFT*) do sinal e as densidades espectrais de potência, chamadas de $P_{x_i}[k]$ e $P_{y_i}[k]$, onde i é a contagem dos quadros.

4.5 Conversão e filtragem da escala de freqüência

A escala de bandas críticas é dividida em intervalos iguais. Para cada intervalo, um valor de densidade de potência perceptual é computada a partir das amostras da banda da densidade espectral de potência na escala em Hertz. As amostras da densidade de potência perceptual para a banda j no quadro i são dadas por:

$$P'_{x_i}[j] = S_p \frac{\Delta f_j}{\Delta z} \frac{1}{I_l[j] - I_f[j] + 1} \sum_{I_f[j]}^{I_l[j]} P_{x_i}[k]$$
$$P'_{y_i}[j] = S_p \frac{\Delta f_j}{\Delta z} \frac{1}{I_l[j] - I_f[j] + 1} \sum_{I_f[j]}^{I_l[j]} P_{y_i}[k] \qquad (4.3)$$

onde $I_f[j]$ e $I_l[j]$ são os índices da primeira e última amostra na escala em Hertz para banda j, respectivamente. O termo Δf_j é a largura de banda em Hertz da banda j, Δz é a largura de banda de cada sub-banda no domínio das bandas críticas, e S_p é o fator de calibração de potência definido na recomendação.

4.5.1 Escalonamento Global

Os sinais necessitam ser escalonados a cada quadro para compensar variações lentas de ganho. Apenas componentes de tempo-freqüência audíveis são considerados (acima do limiar absoluto para cada banda $P_o[j]$). As potências totais dos sinais fonte e decodificado no quadro i, P'_{x_i} e P'_{y_i}, são computadas usando a escala de freqüência convertida :

$$P'_{x_i} = \sum_{j=1}^{Nb} P'_{x_i}[j]$$
$$P'_{y_i} = \sum_{j=1}^{Nb} P'_{y_i}[j] \tag{4.4}$$

onde N_b é o número total de bandas.

Quando ambas as potências totais são maiores que 40 dB SPL, a potência do sinal decodificado é multiplicada pelo fator de escala S_i:

$$P''_{y_i}[j] = S_i P'_{y_i}[j], \tag{4.5}$$

onde

$$S_i = \frac{P'_{x_i}}{P'_{y_i}}. \tag{4.6}$$

Quando a potência total do sinal fonte ou do sinal decodificado está abaixo de 40 dB SPL, a potência do sinal decodificado para a banda j é multiplicada pelo fator de escala S_{av}, o qual é a média de todos os fatores S_i calculados anteriormente.

4.5.2 Filtragem na banda telefônica

As potências amostradas $P'_{x_i}[j]$ e $P''_{y_i}[j]$ precisam ser filtradas usando características de recepção apropriadas para um aparelho telefônico. Esta operação é realizada no domínio da freqüência:

$$P_{Fx_i}[j] = F[j]P'_{x_i}[j]$$
$$P_{Fy_i}[j] = F[j]P'_{y_i}[j], \tag{4.7}$$

onde $F[j]$ é a resposta em freqüência na banda j para as características de recepção do aparelho telefônico. Estes valores são definidos nas recomendações P.830 e P.861.

4.5.3 Ruído *Hoth*

Em uma conversação telefônica normal, há distorção do sinal de fala devido a ruído no ambiente de recepção. Este efeito é modelado através da adição de ruído *Hoth* aos sinais fonte e decodificado, conforme recomendação P.800:

$$P_{Hx_i}[j] = H[j] + P_{Fx_i}[j]$$
$$P_{Hy_i}[j] = H[j] + P_{Fy_i}[j], \tag{4.8}$$

onde $H[j]$ é a potência do ruído *Hoth* na banda j, seus valores são fornecidos na recomendação P.861.

4.6 Conversão Não Linear da Escala de Intensidade

A partir das densidades de potência perceptual são calculadas as densidades do nível de audibilidade comprimido. Esta operação visa compensar o efeito de audição subjetiva dependente da freqüência de um tom de estímulo. Usa-se a função de compressão:

$$L_{x_i}[j] = S_l \left(\frac{P_0[j]}{0{,}5} \right)^{\gamma} \left[\left(0{,}5 + 0{,}5 \frac{P_{Hx_i}[j]}{P_0[j]} \right)^{\gamma} - 1 \right] \tag{4.9a}$$

e

$$L_{y_i}[j] = S_l \left(\frac{P_0[j]}{0{,}5} \right)^{\gamma} \left[\left(0{,}5 + 0{,}5 \frac{P_{Hy_i}[j]}{P_0[j]} \right)^{\gamma} - 1 \right], \tag{4.9b}$$

onde $P_0[j]$ é o limiar interno de audibilidade, cujos valores são fornecidos na recomendação P.861.

O valor ótimo de γ foi obtido através de otimizações experimentais, sendo igual a 0,001, conforme a recomendação P.861.

O nível de audibilidade comprimido momentâneo (na unidade sones comprimidos) são computados pela somatória das densidades do nível de audibilidade comprimido conforme as equações:

$$L_{x_i} = \sum_{j=1}^{Nb} L_{x_i}[j] \Delta z$$
$$L_{y_i} = \sum_{j=1}^{Nb} L_{y_i}[j] \Delta z. \tag{4.10}$$

Os níveis de audibilidade comprimidos momentâneos são utilizados no modelamento cognitivo.

4.7 Modelagem Cognitiva

No método PSQM, todas as operações que não podem tratar isoladamente o sinal fonte ou o sinal decodificado são denominadas de operações cognitivas. São tratados quatro efeitos :

- escalonamento do nível de audibilidade ;
- ruído cognitivo interno;
- processamento assimétrico;
- processamento de intervalos de silêncio.

4.7.1 Escalonamento do Nível de Audibilidade

A densidade amostrada do nível de audibilidade comprimido do sinal codificado é escalonada, a cada quadro, com relação ao nível de audibilidade do sinal fonte:

$$L'_{y_i}[j] = S_{l_i} L_{y_i}[j], \tag{4.11}$$

onde o fator de escala S_{l_i} é calculado a partir dos níveis de audibilidade comprimidos momentâneos :

$$S_{l_i} = \frac{L_{x_i}}{L_{y_i}}. \tag{4.12}$$

Quando L_{x_i} ou L_{y_i} tiveram valores abaixo de 0,02 sones comprimidos, S_{l_i} é igualado a 1.

4.7.2 Densidade amostrada de perturbação de ruído

A densidade amostrada de perturbação de ruído $N_i[j]$ na banda *j* e no quadro *i* é calculada como a diferença absoluta entre L_{x_i} e L'_{y_i} :

$$N_i[j] = \left| L'_{y_i}[j] - L'_{x_i}[j] \right| - 0{,}01, \tag{4.13}$$

onde o fator 0,01 sones comprimidos representam o ruído interno cognitivo. Se devido ao fator de 0,01, $N_i[j]$ tornar-se negativo, então $N_i[j]$ é igualado a zero.

4.7.3 Processamento Assimétrico

Quando uma nova componente de tempo-freqüência é introduzida no sinal de fala, a

qualidade subjetiva torna-se mais degradada do que quando uma componente de mesmo volume é suprimida num processo de codificação. Esta assimetria tem maior ocorrência nos intervalos de silêncio. Se há ruído presente no sinal fonte, este pode ser suprimido, levando a uma melhoria na qualidade. Se não há ruído no sinal fonte durante o intervalo de silêncio, qualquer diferença entre o sinal fonte e o decodificado leva a um decréscimo da qualidade.

Este efeito de assimetria é quantificado por $C_i[j]$ que é incluso na perturbação de ruído no quadro *i*:

$$N_i = \sum_{j=1}^{Nb} N_i[j] C_i[j] \Delta z \,, \tag{4.14}$$

onde

$$C_i[j] = \left(\frac{P_{Hy_i}[j] + 1}{P_{Hx_i}[j] + 1} \right)^{0,2} \,, \tag{4.15}$$

sendo $P_{Hx_i}[j]$ e $P_{Hy_i}[j]$ as potências dos sinais fonte e decodificado, respectivamente, no quadro i e banda j. Quando $P_{Hx_i}[j]$ e $P_{Hy_i}[j]$ têm menos do que 20 dB acima do limiar absoluto de audibilidade na banda j, $C_i[j]$ é igualado a 1. O máximo valor de $C_i[j]$ deve ser limitado a 2,0.

4.7.4 Perturbação de ruído incluindo o processamento no intervalo de silêncio

Os intervalos de silêncio são considerados usando um fator de ponderação W_{sil}, que depende do contexto dos experimentos subjetivos. Quadros de silêncio são definidos como quadros para o qual o sinal fonte tem uma potência total Px_i' abaixo de 70 dB SPL. Uma vez determinado o fator de calibração global S_p, o limiar de silêncio é tomado como $Px_i' = 10^7$. Quadros com Px_i' menor que este valor são considerados silêncio.

Os níveis de audibilidade de ruído médio, N_{spav} e N_{silav}, podem ser calculados a partir dos quadros com voz ativa ou quadros de silêncio, respectivamente:

$$N_{spav} = \frac{1}{M_{sp}} \sum_{\substack{i \text{ de quadros} \\ \text{de fala ativa}}} N_i$$
$$N_{silav} = \frac{1}{M_{sil}} \sum_{\substack{i \text{ de quadros} \\ \text{de silêncio}}} N_i \qquad (4.16)$$

onde M_{sp} é o número de quadros de voz ativa e M_{sil} é o número de quadros de silêncio.

A influência dos intervalos de silêncio depende diretamente do comprimento destes intervalos. Se o sinal fonte não contém intervalos de silêncio, a influência é zero. Se o sinal fonte contém uma percentagem de quadros de silêncio, a influência é proporcional a esta percentagem. Não usando condições ambientais especiais, o valor de perturbação de ruído corrigido com o fator de peso W_{sil} para intervalos de silêncio é dado por:

. (4.17)

Na equação p_{sil} é a fração de quadros de silêncio, p_{sp} é a fração de quadros de voz ativa ($p_{sil} + p_{sp} = 1{,}0$), W_{sil} é o fator de ponderação para intervalos de silêncio, e $W_{sp} = \frac{1 - W_{sil}}{W_{sil}}$.

Para materiais com cerca de 50% de intervalos de silêncio é recomendado $W_{sil} = 0{,}2$.

A perturbação de ruído N_{Wsil} é finalmente o valor denominado de PSQM.

4.8 PSQM+

Estudos realizados [36] mostram que se o sinal decodificado contém pequenas distorções, a modelagem assimétrica (seção 4.7.3) melhora a correlação entre os resultados objetivos e subjetivos. Mas, se as distorções são muito grandes, a correlação entre o valor PSQM e a medida subjetiva torna-se pobre. Para distorções de grampeamento temporal, as componentes espectrais de potência $P_{Hy_i}[j]$ do sinal degradado são muito pequenas, levando a um pequeno valor de perturbação do ruído N_i. Isto, após calculada a média sobre todos os quadros *i*, fornece um valor PSQM bem pequeno. Este problema pode ser resolvido de várias maneiras, mas uma forma fácil e eficiente de compensar este efeito é introduzir um fator de escala adicional

$$S_i^+ = \min\left(\frac{Px_i' + 1}{Py_i' + 1}, 10000\right). \tag{4.18}$$

Este fator de escala é utilizado no cálculo da perturbação do ruído N_i em cada quadro *i*

$$N_i = \left(S_i^+\right)^{0,25} \sum_{j=1}^{N_b} N_i[j] C_i[j] \Delta z . \tag{4.19}$$

Quando a potência total do sinal decodificado e do fonte são aproximadamente as mesmas, o fator de escala S_i^+ tem seu valor em torno de 1,0 e o valor PSQM resultante é o mesmo obtido pela recomendação P.861. Maiores valores PSQM são encontrados na ocorrência de grampeamento temporal e menores valores PSQM são encontrados na ocorrência de distorções do nível de audibilidade, quando comparados aos valores PSQM encontrados pela P.861. Esta nova versão é chamada de PSQM+.

Capítulo 5

Refinamento da Modelagem Autoregressiva

Neste capítulo o método utilizado para aprimorar os codificadores VSELP e EFR será explanado.

5.1 Introdução

Na busca de melhorar a qualidade do sinal codificado optamos por melhorar a modelagem do trato vocal. Esta modelagem é do tipo autoregressiva (AR) para ambos os codificadores aqui explanados. Resultados apresentados na literatura especializada mostram que se a modelagem usar o tamanho da janela de análise igual ou múltiplo do período de *pitch* e se a posição desta janela for síncrona com a abertura da glote ocorrerá uma melhor estimação dos formantes [8][9]. Como a janela de análise varia de tamanho de acordo com o período de *pitch*, estas podem ter pequena duração (da ordem de 5 ms para *pitch* de 200 Hz). Portanto o método da covariância na modelagem AR é mais apropriado. Para garantir a estabilidade dos modelos AR's, é utilizado o método da covariância modificada [24]. Embora existam algoritmos eficazes para se encontrar os instantes de abertura ou fechamento da glote, neste trabalho optamos por realizar a busca da posição da janela de análise dentro do quadro utilizado pelo codificador original (padrão) de forma exaustiva. A medida utilizada para determinar a melhor posição e o melhor tamanho do quadro foi a relação sinal ruído (SNR) ponderada.

O trabalho envolve dois tipos de simulação:

- a primeira busca a maior SNR ponderada variando-se o tamanho e a posição da janela de análise dentro do quadro utilizado no codificador original (padrão). Esta busca é demorada, porém apresentou bons resultados.
- o segundo método utiliza um detector de *pitch* [25] para os quadros sonoros como estimativa da duração ótima, e busca-se apenas a posição da janela de análise, a fim de obter a maior SNR ponderada.

Não é nossa intenção a concepção de um codificador apropriado para execução em tempo real, mas a verificação da melhoria dos codificadores através de simulações por computador. A modificação mantém a mesma distribuição de bits do codificador original. Desta forma, não é necessária nenhuma modificação no decodificador, existindo uma compatibilidade total entre os codificadores original e refinado.

5.2 Refinamento da Modelagem Autoregressiva

Nesta seção detalharemos os dois métodos utilizados nas simulações. O primeiro método, que faz busca exaustiva da duração e posição da janela de análise para cálculo dos coeficientes LPC, será denominado de método **exaustivo** e seu algoritmo está definido na Tabela 5.1. O segundo método, no qual faz-se a estimação do *pitch*, será denominado de método **simplificado.**

O primeiro método procura a melhor posição e o melhor tamanho da janela de análise dentro do quadro usual dos codificadores. Assim, para cada quadro do codificador original temos a análise AR utilizando o melhor tamanho e posição sem alterar a taxa final de codificação. O menor quadro de análise utilizado foi de 32 amostras, corresponde a 4 ms. É recomendado usar o método da covariância para janelas de análise de curta duração (ao redor de 5 ms). Embora ambos os codificadores considerados utilizem 20 ms como duração do quadro, a janela de análise dos mesmos varia devido às diferenças na análise autoregressiva [6][7]. Para o VSELP, que utiliza o método da covariância, a janela de análise é de 170 amostras. Nas simulações, o tamanho da janela de análise utilizada é 32 a 170 amostras, e sua posição é procurada dentro da janela utilizada pelo codificador original. O período de 32 amostras corresponde a um *pitch* máximo de 250 Hz. Durações ainda menores da janela de análise comprometeriam a modelagem AR (matrizes mau

condicionadas). Para cada quadro do codificador VSELP original (160 amostras), o número total de variações é de 9730 (janelas de 32 a 170 amostras). O EFR utiliza o método da autocorrelação, por meio de Levinson-Durbin, o qual necessita realizar um janelamento do sinal de fala. Como é proposto utilizar o método da covariância, este janelamento é omitido. O tamanho desta janela de análise utilizada no codificador original é de 240 amostras. Então, para realizar a comparação, a janela de análise do método proposto varia de 32 a 240. Assim o número total de variações é de 21945.

```
Para quadro = 1 : Nq
    Para tamanho = pitch_min : pitch_max
        Para posição = 1 : janela_padrão – tamanho
            codifica sinal
            decodifica sinal
            calcula SNR
            Se SNR > SNR_max
                posição e tamanho ótimos
                SNR_max = SNR
```

Tabela 5.1 – Algoritmo do método exaustivo

O esforço computacional diminui com a inserção do detector de *pitch*. De posse do valor de *pitch*, o tamanho da janela é feito seu múltiplo e o número de posições procuradas é (170 – *pitch*) para o VSELP e (240 – *pitch*) para o EFR. Note que o número de variações diminui bastante, tornando a busca mais rápida. O algoritmo para este método está descrito na Tabela 5.2.

Para ambos os casos propostos, se o quadro for surdo, utiliza-se a janela de análise do codificador original.

```
Para quadro = 1 : Nq
    Calcula pitch
    Para posição = 1 : quadro_padrão – pitch
        codifica sinal
        decodifica sinal
        calcula SNR
        Se SNR > SNR_max
            posição ótima
            SNR_max = SNR
```

Tabela 5.2 – Algoritmo para o método simplificado.

5.3 Covariância Modificada

Para a análise LPC, utilizamos o método da covariância, pois o método da autocorrelação não permite trabalhar com intervalos pequenos de amostras. Mas o método da covariância é instável. Para estabilizá-lo, Dickinson propôs um método que se baseia nas energias residuais. Este método, chamado de **covariância modificada** [24]**,** é utilizado neste trabalho.

Seja $x_N = (x_1, \cdots, x_N)$ uma série temporal que resulta no modelo autoregressivo de ordem *p*

$$x_t - a_1 x_{t-1} - \cdots - a_p x_{t-p} = e_t \,, \tag{5.1}$$

onde e_t são variáveis aleatórias de média zero com variância σ^2 . É sabido que muitos estimadores de parâmetros, incluindo os métodos dos mínimos quadrados e o de probabilidade máxima, não asseguram a estabilidade do filtro de síntese.

No método da covariância, o erro quadrado mínimo expressa-se por:

$$E_j = \sum_{t=p+1}^{N} \left(x_t - a_1^j x_{t-1} - \cdots - a_j^j x_{t-j} \right)^2 , \tag{5.2}$$

onde $\{a_i^j\}$ são os coeficientes ótimos do modelo AR, $1 \le j \le p$.

Por outro lado, a analogia do problema com o método da autocorrelação permite a inferência a partir da recursão de Levinson-Durbin

$$E_j = \left(1 - k_j^2\right) E_{j-1} . \tag{5.3}$$

Ou ainda, a aplicação recursiva desta equação da ordem 1 a j

$$E_j = E_0 \left(1 - k_1^2\right) \cdots \left(1 - k_j^2\right) \tag{5.4}$$

Da equação (5.3), evidenciando-se k_j, encontra-se uma nova relação para k_j, a menos da ambigüidade de seu sinal,

$$k_j = -sign\left(a_j^j\right) \left(1 - \frac{E_j}{E_{j-1}}\right)^{1/2} . \tag{5.5}$$

Novamente a analogia com o método da correlação permite a dedução do sinal de k_j, por meio de $k_j = -a_j^j$, expresso na equação acima.

A iniciação é feita por

$$E_0 = \sum_{t=p+1}^{N} x_t^2 . \tag{5.6}$$

Como $E_{j-1} \geq E_j$ e os erros quadrados são grandezas positivas, segue

$$\left|k_j\right| < 1 , \tag{5.7}$$

condição que garante a estabilidade do filtro de síntese. Uma vez obtidos os $\{k_j\}$, calcula-se os $\{a_i^j\}$ pelo procedimento *step-up*. Resultados publicados na literatura especializada mostram que este método é semelhante ao da covariância, porém tem a estabilidade garantida. Em efeito, em nossas simulações não se verificou modelos instáveis. Note que o algoritmo determina os $\{a_i^j\}$ da mesma maneira que pelo método da covariância e emprega (5.2), (5.5) e (5.6) para determinar um novo conjunto de $\{k_j\}$.

5.4 Determinação do *Pitch*

A determinação do *pitch* é considerada uma das tarefas mais difíceis no processamento de fala. A complexidade da determinação de *pitch* vem da não estacionaridade da fala. Muitos

algoritmos foram apresentados, sendo que neste trabalho utilizamos o proposto por Medan et al. [25].

Devido à não estacionaridade do sinal, na estimativa de *pitch* são empregadas janelas de curta duração. Contudo, devido à faixa de possíveis valores de *pitch*, a janela de análise pode conter vários períodos, levando a uma média ou mesmo a um erro na determinação do *pitch*.

Um erro adicional que limita a exatidão do algoritmo é a discretização no tempo (quantização) na estimação de *pitch*, introduzida pela amostragem do sinal de fala. A estimação de *pitch*, expressa como um múltiplo inteiro do intervalo amostrado, contém um erro de quantização que pode levar a distorções audíveis.

O algoritmo proposto por Medan et al. utilizado neste trabalho supera muitas destas dificuldades introduzindo um modelo que permite quantificar o grau de similaridade entre dois intervalos de *pitch* adjacentes e não sobrepostos. O algoritmo oferece um esquema de implementação robusto, de alta resolução e eficiente, o qual é capaz de evitar distorções audíveis associadas à determinação de *pitch*.

5.4.1 Determinação do *Pitch*

A apresentação a seguir é feita no domínio contínuo do tempo, conforme a apresentação do artigo original [25].

Para cada instante t_0, nós definimos dois sinais $x_\tau(t,t_0)$ e $y_\tau(t,t_0)$ como sendo :

$$x_\tau(t,t_0) = s(t)w_\tau(t-t_0)$$
$$y_\tau(t,t_0) = s(t+\tau)w_\tau(t-t_0) \qquad (5.8)$$

onde $s(t)$ é o sinal de fala, e $w_\tau(t)$ é uma janela retangular de tamanho τ dado por

$$w_\tau(t) = 1 \quad \text{para } 0 \le t < \tau$$
$$w_\tau(t) = 0 \quad \text{caso contrário.} \qquad (5.9)$$

Pela equação (5.8) temos dois segmentos de fala adjacentes de duração τ segundos no intervalo $[t_0, t_0+\tau]$.

Considere um quadro de fala que inicia em $t = t_0$ e consiste de exatamente dois períodos de *pitch* de duração $\tau = T_0$, onde $x_{T_0}(t,t_0)$ é o primeiro período e $y_{T_0}(t,t_0)$ é o

segundo período, e T_0 é o período de *pitch* (em segundos) relativo ao instante de tempo $t = t_0$. Assumindo-se que a similaridade entre dois períodos sucessivos de *pitch* é alta, pode se assumir que um segmento é a versão de amplitude modulada do outro :

$$x_{T_0}(t, t_0) = a(t_0)\, y_{T_0}(t, t_0) + e(t, t_0) \tag{5.10}$$

onde $a(t_0)$ é um fator de modulação em amplitude (ganho), positivo e desconhecido, no instante t_0. O termo de erro $e(t,t_0)$ representa as outras dissimilaridades entre os dois períodos. A maximização da similaridade entre os dois segmentos, conduz à uma estimativa de intervalo de *pitch* (T_0). A minimização do erro quadrático leva ao problema de otimização :

$$T_0 = \underset{\tau, a(t_0)>0}{\arg\min} \left\{ J = \frac{\int_{t_0}^{t_0+\tau} \left[x_\tau(t, t_0) - a(t_0)\, y_\tau(t, t_0)\right]^2 dt}{\int_{t_0}^{t_0+\tau} \left[x_\tau(t, t_0)\right]^2 dt} \right\} \tag{5.11}$$

onde o erro é minimizado dentro do intervalo $[t_0, t_0 + \tau]$.

O termo de normalização do denominador é necessário para compensar o tamanho variável dos segmentos de fala e a distribuição de energia no intervalo de *pitch*. Na prática τ pode ser restrito dentro da faixa dos períodos de *pitch* esperados: $T_{0_{\min}} \le \tau \le T_{0_{\max}}$.

Para se determinar o valor ótimo do ganho $a(t_0)$, deriva-se a função custo J em relação a $a(t_0)$ e iguala-se a zero. Assim o ganho ótimo será :

$$a(t_0) = \frac{\langle x, y \rangle_\tau}{|y|_\tau^2}, \tag{5.12}$$

onde $\langle x, y \rangle_\tau$ é a média temporal do produto $x_\tau(t, t_0)$ e $y_\tau(t, t_0)$ no intervalo de tempo $[t_0, t_0 + \tau]$ dada por

$$\langle x, y \rangle_\tau = \int_{t_0}^{t_0+\tau} x_\tau(t, t_0)\, y_\tau(t, t_0) dt \tag{5.13}$$

e $|y|_\tau^2 = \langle y, y \rangle_\tau$ é a energia do segmento $y_\tau(t, t_0)$. Assim a função custo fica sendo :

$$J = 1 - \rho_\tau^2(x, y) \tag{5.14}$$

onde $\rho_\tau(x, y)$ é o coeficiente de correlação cruzada entre os segmentos *x* e *y*

$$\rho_\tau(x,y) = \frac{\langle x, y \rangle_\tau}{|x|_\tau \, |y|_\tau}, \tag{5.15}$$

o qual é restrito a ser positivo, já que $a(t_0)$ é assumido positivo. O período de *pitch* T_0 no instante t_0 pode ser calculado através do problema de maximização

$$T_0 = \arg\max_\tau \rho_\tau(x,y) \tag{5.16}$$

para $T_{0_{\min}} \le \tau \le T_{0_{\max}}$.

A minimização do erro quadrático médio na equação (5.11) eqüivale à maximização da correlação cruzada na equação (5.15).

A solução da equação (5.16) pode ser obtida usando técnicas digitais na qual o sinal $s(t)$ é amostrado uniformemente no intervalo de amostragem T. O período de *pitch* encontrado neste caso é de resolução finita, ditada pelo intervalo de amostragem, e é chamado de *pitch* inteiro.

A Figura 5.1 ilustra a busca do *pitch* inteiro. Duas janelas adjacentes de tamanho variável de n amostras cada uma, iniciando em $t = t_0$, são usadas para formar os vetores x_n e y_n. O coeficiente de correlação cruzada $\rho_n(x,y)$ é calculado no intervalo $N_{\min} \le n \le N_{\max}$. O *pitch* inteiro $\underline{N}$, corresponde ao valor de n no qual $\rho_n(x,y)$ é máximo. No exemplo $\underline{N} = n_2$.

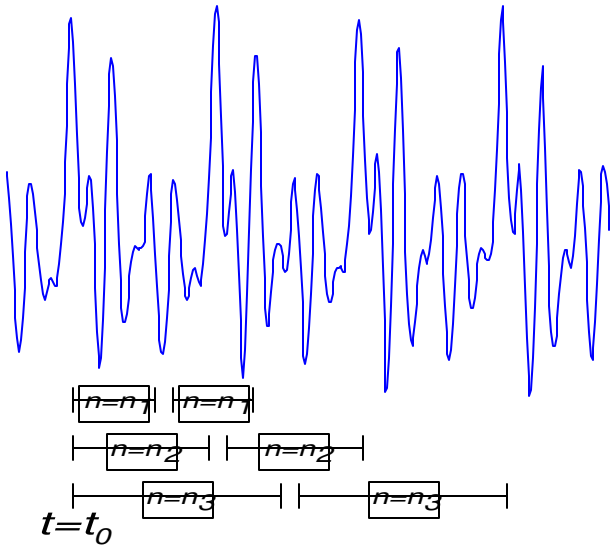

Figura 5.1 – Determinação do *pitch* inteiro.

Para evitarmos que a correlação calculada assuma valores altos para baixos valores de *pitch*, causando uma estimação errada de *pitch* o procedimento descrito a seguir foi adicionado por nós ao algoritmo acima explanado. A correlação cruzada é calculada para diversos segmentos no intervalo dado, de 32 a 85 amostras (250 a 94 Hz) no caso do VSELP e de 32 a 120 amostras (250 a 66 Hz) para o EFR. De posse destes valores de correlação cruzada, a média aritmética é calculada e este novo vetor é considerado o vetor de correlação cruzada do sinal.

Outra alteração realizada é a adição de uma janela exponencial ao vetor de correlação cruzada com fator de decaimento de 0,8 do máximo valor da faixa de *pitch*. Isto é feito para penalizar a estimação de *pitch* múltipla do valor real.

5.5 O Classificador Sonoro/Surdo

A análise do desempenho por meio da SNR segmentar é realizada para trechos sonoros, onde há a presença de *pitch*. O cálculo da SNR segmentar apenas para os trechos sonoros da fala é motivado pelo fato de que se busca um sincronismo da análise LPC com o *pitch*. Para isto, faz-se necessário determinar se o quadro em análise é sonoro ou surdo. A partir desta classificação, calcula-se a SNRseg para os quadros sonoros, que é maior que a SNRseg total.

O classificador utilizado é baseado nos seguintes parâmetros : o valor de correlação cruzada calculada para o detector de *pitch*, o valor RMS do quadro, a média de cruzamentos por zero e o coeficiente de autocorrelação de curto termo normalizado de atraso unitário. A seguir apresentamos o algoritmo do classificador sonoro/surdo. Se *iDecisão* = 0 então o trecho é surdo, caso contrário o trecho é sonoro.

Na Tabela 5.3 temos descrito o algoritmo usado para o classificador sonoro/surdo.

1. Inicializa limiares

   ref_cruz_zero = REF_CRUZ_ZERO

   ref_autocorrelação = REF_AUTOCOR

   ref_*pitch*=REF_*PITCH*

2. Se o quadro anterior é sonoro dividir ref_cruz_zero, ref_autocorrelação e ref_*pitch* por 2.
3. Estima o valor de *Pitch* (*pitch*).
4. Calcula o valor RMS do quadro (VRMS).
5. Calcula a taxa de cruzamento por zero (cruz_zero).
6. Calcula o coeficiente de autocorrelação (r_1).
7. Se VRMS > Noise_ref
   - se *pitch* > 0
     - se cruz_zero > ref_cruz_zero
       - iDecisão = 0
       - se r_1 > ref_autocorrelação
         - iDecisão = *pitch*
     - senão
       - iDecisão = 0
   - senão
     - iDecisão = 0
   - senão
     - iDecisão = 0

Tabela 5.3 – Algoritmo do classificador sonoro/surdo.

Capítulo 6

Resultados

Neste capítulo são apresentados e analisados os resultados das simulações realizadas com os codificadores VSELP e EFR. O codificador VSELP utilizado foi implementado a partir da documentação do padrão. O codificador EFR utilizado foi fornecido pela TIA juntamente com a documentação. A medida PSQM, utilizada para mensurar a qualidade dos sinais, foi implementada a partir da documentação da recomendação e validada por sinais de teste que acompanham a recomendação.

6.1 Medidas Utilizadas

É necessário avaliar a qualidade dos codificadores. Na busca de otimização, fez-se uso da medida objetiva relação sinal-ruído segmentar SNRseg. Esta medida é dada por

$$SNRseg = \frac{10}{L}\sum_{i=0}^{L-1}\log_{10}\left\{\frac{\sum_{n=0}^{N-1}s^2(iN+n)}{\sum_{n=0}^{N-1}\left(s(iN+n)-\hat{s}(iN+n)\right)^2}\right\} \qquad (6.1)$$

onde $s(n)$ é o sinal de voz original e $\hat{s}(n)$ é o sinal de voz decodificado. O valor N representa o tamanho do quadro analisado (160 amostras) e L é a quantidade total de quadros existentes na locução. Esta medida é mais adequada que a SNR, pois reflete a qualidade do sinal a cada quadro, observando a variação de energia quadro a quadro.

As medidas subjetivas tradicionalmente são realizadas por meio de testes através da audição dos sinais. A medida subjetiva normalmente utilizada é a MOS (*Mean Opinion Score*). Esta medida é de difícil implementação em trabalhos acadêmicos, pois demanda

tempo e necessita de grande quantidade de pessoas para realizar os testes. O método PSQM é feito por meio de cálculos numéricos e tem forte relação com os resultado obtidos pelo método MOS. Assim utilizamos o método PSQM para analisar a qualidade subjetiva dos métodos de refinamento propostos neste trabalho.

Não existe uma única função que transforma os valores PSQM em MOS. Os resultados publicados [27] mostram que isto depende do contexto dos testes, e até mesmo da língua. Uma relação do PSQM com o MOS está ilustrada na Figura 6.1. Observa-se que quanto menor o valor PSQM do sinal, melhor a qualidade do mesmo.

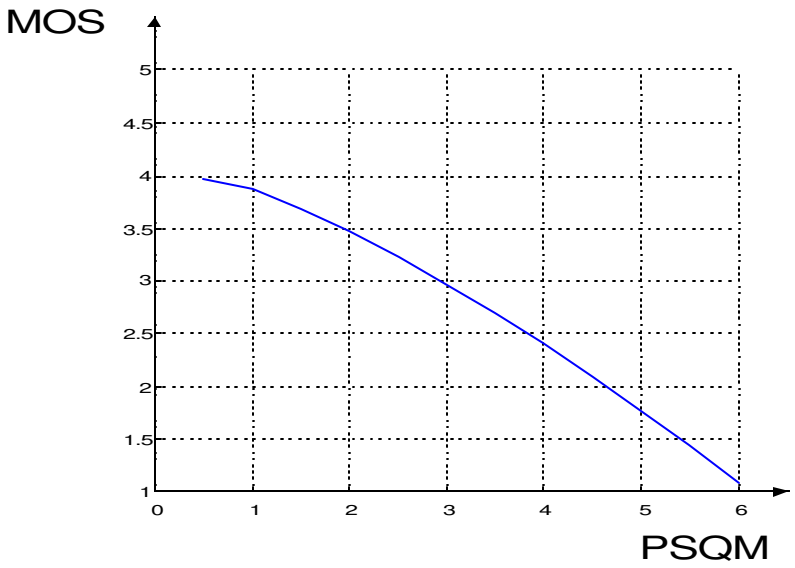

Figura 6.1 – Gráfico da relação MOS x PSQM

6.2 Sinais Utilizados

Foram utilizados os sinais de voz retirados do *Telephone Network Acoustic-Phonetic Continuous Speech Corpus* – NTIMIT [28]. Esta base de dados foi gerada com transmissão de voz em linhas telefônicas reais. O idioma no qual foi gravada é o Inglês. Os sinais estão amostrados a 16 kHz, porém os codificadores trabalham com sinais amostrados a 8 kHz. Necessita-se, então, dizimar o sinal. Para tal, usou-se de um filtro FIR passa-baixas projetado no MATLAB. O filtro com 97 taps tem sua freqüência de corte em torno de 3400 Hz e 64 dB de atenuação em 4000 Hz. A SNR entre o sinal original e o filtrado é de, no

mínimo, 30 dB, o que mostra que quase toda a energia do sinal está nas freqüências dentro da banda do canal telefônico.

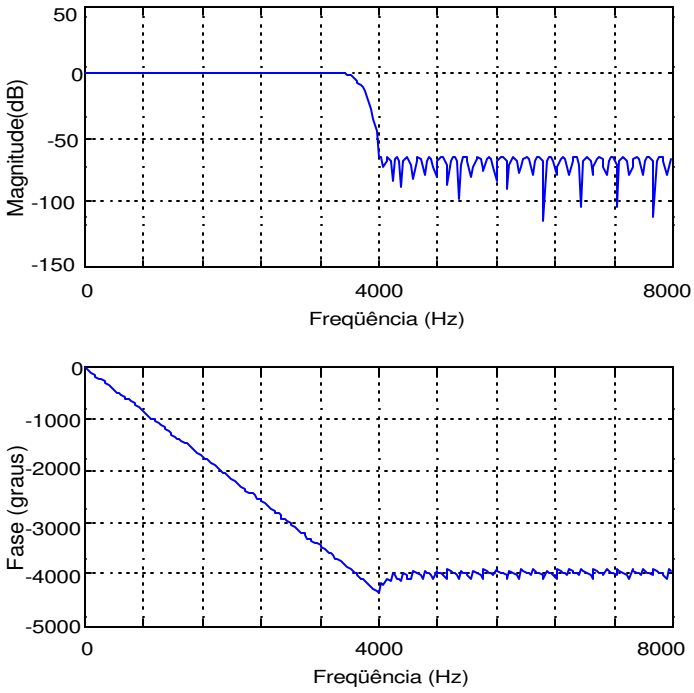

Figura 6.2 – Resposta em magnitude e fase do filtro FIR passa-baixas.

A medida PSQM em [10] trabalha com sinais de fala amostrados a 16 kHz. Torna-se, então, necessária a interpolação do sinal decodificado, a qual é realizada também com a utilização do filtro apresentado acima.

Na Tabela 6.1 temos as frases utilizadas neste trabalho. São três masculinas e três femininas.

6.3 Resultados obtidos para VSELP

Visando a comparação de desempenho, a SNRseg é calculada a partir do sinal $s(n)$ na saída do filtro Chebyshev do tipo II no codificador e do sinal sintetizado no decodificador $s'(n)$ antes do pós-filtro (vide Figuras 2.1 e 2.6). Temos na literatura [33] que o valor MOS para o codificador VSELP é de 3,5, o qual foi verificado usando-se a medida PSQM e fazendo a conversão pelo gráfico da figura 6.1.

Frase	Sexo	Duração	Conteúdo
DR1\FCJF0\SX307	Feminino	1,6 s	*The meeting is now adjourned.*
DR2\FAEM0\SX402	Feminino	2,8 s	*We'll serve rhubarb pie after Rachel's talk*
DR5\FBMH0\SX56	Feminino	2,5 s	*Academic aptitude guarantees your diploma*
DR3\MADC0\SX17	Masculino	1,7 s	*Carl lives in a lively home*
DR5\MHIT0\SX263	Masculino	2,0 s	*How oily do you like your salad dessing?*
DR8\MBSB0\SX363	Masculino	2,0 s	*The cow wandered from the farmland and became lost.*

Tabela 6.1 – Descrição dos sinais utilizados nas simulações.

Na Figura 6.3 temos a ilustração da variação da SNR em um quadro sonoro do sinal DR1\FAEM0\SX402. Esta Figura foi obtida para a busca simplificada. Para a janela de análise de tamanho igual a 39 nota-se que para a posição 73 ocorre o mínimo valor de SNR, 10,24dB e para a posição 66 ocorre o máximo valor de SNR, 19,75 dB. Assim no total temos a variação de 9,51 dB.

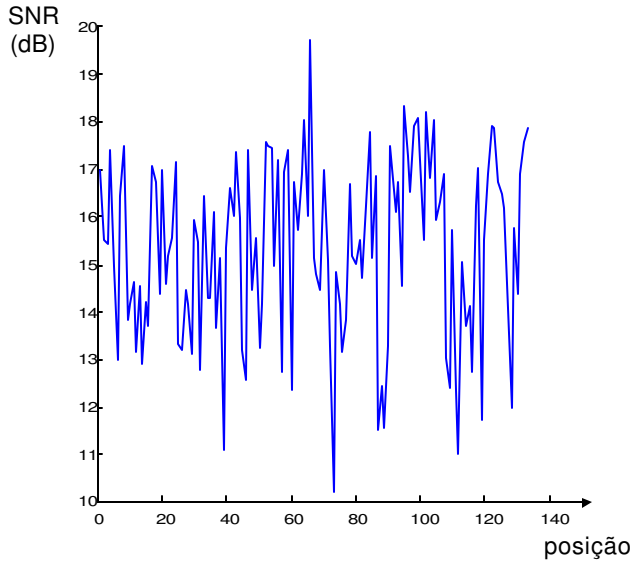

Figura 6.3 – Variação da SNR em um quadro de análise de tamanho 39.

Os resultados obtidos nas simulações, para os quadros sonoros, estão apresentados na Tabela 6.2. Para a busca exaustiva o ganho médio de SNRseg obtido é de 2,35 dB em relação ao codificador original. O melhor resultado obtido foi para a frase DR5\MHIT0\SX263 com ganho de 3,2 dB.

Na busca simplificada, baseada na estimação do período de *pitch*, o ganho médio de SNRseg foi de 1,45 dB e o melhor resultado também foi para a frase DR1\MHIT0\SX263 com 1,91 dB. Isto mostra que a escolha do tamanho do quadro tendo como base a estimação de *pitch* é um método efetivo, com esforço computacional reduzido.

Frase	Codificador Original SNRseg (dB)	Método Exaustivo SNRseg (dB)	Método Simplificado SNRseg (dB)
DR1\FCJF0\SX307	9,98	11,23	10,56
DR2\FAEM0\SX402	10,58	13,23	12,26
DR5\FBMH0\SX56	11,25	13,72	12,90
DR3\MADC0\SX17	10,60	12,42	11,82
DR5\MHIT0\SX263	10,71	13,94	12,62
DR8\MBSB0\SX363	9,97	12,65	11,68
Média	10,52	12,87	11,97

Tabela 6.2 – SNRseg para quadros sonoros do codificador VSELP original, utilizando o método exaustivo e utilizando o método simplificado

Para verificar a ocorrência de melhoria perceptual, utilizou-se a medida PSQM. A medida PSQM é calculada entre o sinal original e o sinal decodificado, aqui tanto para os trechos surdos quanto sonoros da fala. Os resultados estão na Tabela 6.3 e mostram que para o método simplificado não há melhoria na qualidade do sinal, mas para o método exaustivo há uma melhoria de 0,41 no valor PSQM.

Frase	Codificador Original PSQM	Método Exaustivo PSQM	Método Simplificado PSQM
DR1\FCJF0\SX307	1,09	0,98	1,08
DR2\FAEM0\SX402	2,33	1,81	2,23
DR5\FBMH0\SX56	1,63	1,42	1,79
DR3\MADC0\SX17	2,32	1,35	2,37
DR5\MHIT0\SX263	2,12	1,66	2,29
DR8\MBSB0\SX363	1,60	1,41	1,65
Média	1,85	1,44	1,90

Tabela 6.3 – PSQM para o codificador VSELP original, o método exaustivo e o método simplificado

6.3 Resultados obtidos para EFR

Os sinais utilizados para o cálculo da SNRseg são o sinal de voz filtrado no pré-processamento, $s(n)$ (codificador), e o sinal sintetizado $\hat{s}(n)$, antes do pós-filtro. Estes sinais podem ser visualizados nas Figuras 3.1 e 3.6. Este codificador tem melhor desempenho e seu valor MOS publicado [33] é 4. Nas simulações realizadas, obtemos o valor estimado pela Figura 6.1 de 3,78 do MOS.

Para ilustrar a variação da SNR ocorrida no EFR temos a Figura 6.4. A análise é realizada para o mesmo quadro do sinal que foi realizada no VSELP. Para a posição 105 ocorre o mínimo valor de SNR, 7,91 dB, e para a posição 187 ocorre o máximo valor de SNR, 18,10 dB. Assim, para o exemplo, no total temos a variação de 10,19 dB.

Na Tabela 6.4 temos os resultados da SNRseg dos quadros sonoros para o codificador EFR. Observe que a SNRseg média deste codificador original é 3,47 dB maior que a SNRseg média do codificador VSELP, comprovando que o EFR tem, de fato, melhor desempenho. Observa-se também que a qualidade subjetiva do codificador EFR é maior que a do VSELP.

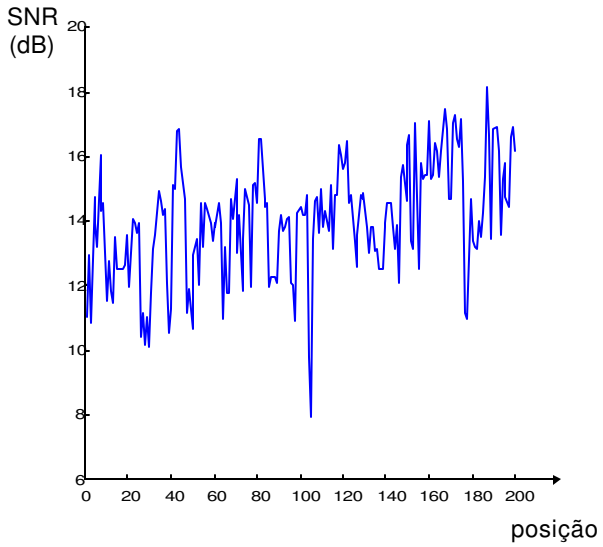

Figura 6.4 – Variação da SNR em um quadro de análise de tamanho 39.

Observando a melhoria ocorrida para o EFR, nota-se um ganho médio de 2,56 dB ocorrido para o método exaustivo. O melhor resultado ocorreu para a frase DR8\MBSB0\SX363 com o ganho de 3,53 dB.

Para o método simplificado o melhor resultado também foi o da frase DR8\MBSB0\SX363, com ganho de 2,35 dB. Neste método a diferença entre as SNRseg média é de 1,82 dB em relação ao codificador original.

Frase	Codificador Original SNRseg (dB)	Método Exaustivo SNRseg (dB)	Método Simplificado SNRseg (dB)
DR1\FCJF0\SX307	13,81	15,62	15,70
DR2\FAEM0\SX402	14,39	16,45	16,00
DR5\FBMH0\SX56	14,63	16,75	16,33
DR3\MADC0\SX17	13,88	16,00	15,24
DR5\MHIT0\SX263	13,88	16,57	15,85
DR8\MBSB0\SX363	13,36	16,89	15,71
Média	13,99	16,55	15,81

Tabela 6.4 – SNRseg para o codificador EFR original, o método exaustivo e o método simplificado

Analisando a qualidade subjetiva do sinal, notamos uma melhora na mesma para o método exaustivo, observada na Tabela 6.5. Para o caso do método simplificado, temos uma melhora de apenas 0,06 na medida PSQM. O ganho obtido para o método exaustivo é de 0,23.

Frase	Codificador Original PSQM	Método Exaustivo PSQM	Método Simplificado PSQM
DR1\FCJF0\SX307	0,82	0,42	0,82
DR2\FAEM0\SX402	1,41	1,34	1,38
DR5\FBMH0\SX56	1,06	1,04	1,02
DR3\MADC0\SX17	1,47	0,72	1,32
DR5\MHIT0\SX263	1,26	1,20	1,18
DR8\MBSB0\SX363	1,04	0,96	1,02
Média	1,18	0,95	1,12

Tabela 6.5 – PSQM para o codificador EFR original, o método exaustivo e o método simplificado

Capítulo 7

Conclusão

Este trabalho teve como motivação procurar melhorar a qualidade dos codificadores de fala utilizados na telefonia celular TDMA. Um caminho para se obter esta melhoria é por meio de refinamento da modelagem autoregressiva. Existem diversas opções para realizar este refinamento. Tomou-se como base estudos realizados que mostram melhoria na estimação de formantes quando a análise AR é realizada com janela de tamanho igual ao período de *pitch* e com posição síncrona à abertura ou fechamento da glote [33][34].

Definiram-se dois tipos de simulações: a primeira, busca exaustiva, na qual a posição e o tamanho da janela de análise são procurados dentro da janela definida no padrão; a segunda, utiliza um estimador de *pitch* para determinar o tamanho da janela de análise, e então, a posição é procurada de forma exaustiva. Embora tenham sido estudados algoritmos para encontrar os instantes de abertura ou fechamento da glote [33][34], estes não foram empregados nas simulações.

Após o estudo de alguns métodos para a extração de *pitch*, optou-se pelo trabalho de Medan et al [25]. Este método tem boa precisão. Devido aos resultados encontrados pelo método simplificado não terem tão bons quanto aos do método exaustivo, sugere-se em trabalhos futuros a seguinte alteração: após o cálculo do período de *pitch,* fazer uma busca do período ótimo para ± 3 valores em torno do valor encontrado.

Por utilizarmos blocos de análise curtos (da ordem de 30 amostras), o método da covariância é mais indicado para modelagem AR. Para garantir a estabilidade dos modelos, usou-se o método da covariância modificada sugerido por Dickinson [24].

Os codificadores utilizados são padronizados pela TIA. O VSELP foi implementado a partir do documento fornecido pela mesma [6]. No caso do EFR utilizou-se o código (em

linguagem C) fornecido pela TIA [7]. Foi implementada neste trabalho a medida PSQM, recomendada pela ITU-T [10], para medir a qualidade dos codificadores em uso na telefonia. Esta medida tem forte correlação com o MOS, indicando a qualidade subjetiva do sinal.

O codificador VSELP transmite os coeficientes de reflexão e os índices de três dicionários, dois fixos e um adaptativo (*pitch*). Temos na literatura [33] que o valor MOS para o codificador VSELP é de 3,5, o qual foi verificado usando-se a medida PSQM e fazendo a conversão pelo gráfico da Figura 6.1. O codificador EFR realiza uma extração de *pitch* mais refinada, com precisão maior que o VSELP, pois extrai seu valor inteiro e fracionário. Os coeficientes transmitidos pelo EFR são os pares de freqüências espectrais, o atraso de *pitch* e o sinal de excitação. Este codificador tem melhor desempenho e seu valor MOS estimado é 4. Nas simulações realizadas, obtemos o valor estimado de 3,78 do MOS para o EFR. Para o refinamento proposto neste trabalho, obtivemos o valor de 3,90 do MOS para o EFR no método exaustivo. Para o VSELP também houve ganho, o valor MOS estimado foi de 3,70. Embora se tenha obtido um melhor desempenho, estes métodos não são recomendados para a utilização em tempo real, dada a elevada complexidade computacional.

Os resultados para a medida objetiva são os seguintes: houve ganho para ambos os métodos, sendo o ganho médio do método exaustivo de 2.35 dB na SNR segmentar para o VSELP e de 2,56 dB para o EFR. No método simplificado, obtivemos ganhos médios de 1,45 dB (VSELP) e de 1,82 dB (EFR). Para a medida subjetiva, observa-se que para o método simplificado praticamente não há ganho em relação ao codificador original. Com relação ao método exaustivo, há ganho médio de 0,41 na PSQM para o VSELP e de 0,23 na PSQM para o EFR. Portanto, o método de fato introduz melhorias observadas tanto com medidas objetivas quanto subjetivas. O uso de um algoritmo de estimação de *pitch* reduz bastante o esforço computacional, porém compromete os ganhos de qualidade.

Bibliografia

[1] I.H.S.P. Fantini e L.G.P. Meloni, "Enhanced VSELP coding by a refined autoregressive modeling", XVII Simpósio Brasileiro de Telecomunicações, Vila Velha, ES, págs. 126-129, set. 1999.

[2] G. Fant, *Acoustic Theory of Speech Production*, Gravenhage, The Netherlands: Mounton and Co., 1960.

[3] L. M. Silva, "Codificação paramétrica da voz – um modelo LPC com excitação mista", Dissertação de Mestrado, Departamento de Engenharia Elétrica, Universidade de Brasília, 1989.

[4] M. R. Schroeder, B. Atal e J. Hall, "Optmizing digital speech coders by exploiting the masking properties of the ear", J. Acoust. Soc. Amer., vol. 66, pág. 1647, 1979.

[5] M. R. Schroeder e B. Atal, "Code-excited linear prediction (CELP): High quality speech at very low bit rates", Proc. ICASSP'85, IEEE Intern. Conf. Acoust., Speech, Signal Process., pág. 937-940, Tampa, FL, E.U.A., abril 1985

[6] TIA/EIA/IS-136.2, "800 MHz TDMA Cellular – Radio Interface – Mobile Station – Base Station Compatibility - Traffic Channels and FSK Control Channel", Telecommunication Industry Association, dez. 1994.

[7] TIA/EIA/IS-641, "TDMA Cellular PCS – Radio Interface – Enhanced Full-Rate Speech Codec", Telecommunication Industry Association, maio 1996.

[8] J. D. Markel e A H. Gray, *Linear Prediction of Speech*, Springer-Verlag, 1976.

[9] L. R. Rabiner e B. S. Atal, "LPC prediction error – analysis of its variation with the position of the frame analysis", IEEE Trans. Acoust., Speech, Signal Processing, vol. 25, no. 5, págs. 434-442, out 1977.

[10] ITU-T Recommendation P.861, " Objective Quality Measurement of Telephone Band (300-3400 Hz) Speech Codecs", ago. 1996.

[11] Y. Tohkura, F. Itakura e S. Hashimoto, "Spectral Smoothing Technique in PARCOR Speech Analysis-Synthesis", IEEE Trans. ASSP, vol. 26, no. 6, págs. 587-595, dez. 1978.

[12] A Cumani, "On a covariance-lattice algorithm for linear prediction", Proc. ICASSP'82, IEEE Int. Conf. Acoust., Speech, Signal Processing, págs. 651-654, maio 1982.

[13] A.V. Oppenheim, e R.W. Schafer, *Discrete-time signal processing*, Prentice Hall, Englewood Gliffs, NJ, 1989.

[14] A.J. Viterbi e J. K. Omura, *Principles of digital communication and coding*, McGraw-Hill, 1979.

[15] J. Makhoul, "New lattice methods for linear prediction", Proc. ICASSP'76, IEEE Int. Conf. Acoust. Speech, Signal Processing, Philadelphia, PA, abril 1976.

[16] J. Burg, "Maximum entropy spectral analysis", Ph.D. Dissertation, Geophysics Department, Stanford University, CA, maio 1975.

[17] Ramachandran, R. P. e Kabal, P., "Pitch prediction filters in speech coding", IEEE Trans. ASSP, vol ASSP-37, no. 4, págs 467-478, abril 1989.

[18] Y. Linde, A Buzo e R. M. Gray, "An algorithm for vector quantizer design", IEEE Trans. Commun., vol COM-28, no. 1, págs. 84-95, jan.1980.

[19] M.H. Hayes, "Statistical digital signal processing and modeling", John Wiley & Sons, 1996.

[20] F. Itakura, "Line Spectral representation of linear predictive coefficients of speech signals", J. Acoust. Soc. Amer., vol. 57, Suplemento no. 1, S35, 1975.

[21] P. Kabal e R. P. Ramachandran, "The computation of line Spectral Frequencies using Chebyshev Polynomials", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-34, no. 6, dez. 1986.

[22] S. Grassi, A Dufaux, M. Ansorge, and F. Pellandini, " Efficient algorithm to compute LSP parameters from 10^{th}-order LPC coefficients", Proc. ICASSP'97, IEEE Intern. Conf. Acoust., Speech, Signal Process., págs 1707-1710, 1997.

[23] C.Laflamme, J. P. Adoul, R. Salami, S. Morissete e P. Mabilleau, "16 kbps wideband speech coding technique based on algebraic CELP", Proc. ICASSP'91, IEEE Intern. Conf. Acoust., Speech, Signal Process., vol. 1, págs. 177-180, 1991.

[24] B. W. Dickinson, "Autoregressive estimation using energy ratios", IEEE Trans. Information Theory, vol. 24, no. 4, págs. 503-506, jul 1978.

[25] Y.Medan, E. Yair e D. Chazan, "Super resolution pitch determination of speech signals", IEEE Trans. Acoust., Speech, Signal Processing, vol. 39, no. 1, págs. 40-48, jan. 1991.

[26] W. Hess, *Pitch determination of speech signals*, Springer-Verlag, 1983.

[27] C. S. Kurashima, "Implementação de um pós-filtro adaptativo para a melhoria da qualidade perceptual de sinais de voz com ruído", Dissertação de Mestrado, Escola Politécnica da USP, 1999.

[28] C. Jankowski, A Kalyanswamy, S. Basson e J. Spitz, "NTIMIT: A phonetically balanced, continuous speech, telephone bandwidth speech database", Proc. IEEE'90, Int. Conf. on Acoust., Speech and Signal Processing, Albuquerque, abril 1990.

[29] A. S. Spanias, "Speech Coding: A tutorial review", Proc. IEEE, vol. 82, no. 10, out. 1994.

[30] J. Thyssen, H. Nielsen, e S. D. Hansen, "Non-linear short term prediction in speech coding", Proc. ICASSP'94, IEEE Intern. Conf. Acoust., Speech, Signal Process., págs. I185-I188, 1994.

[31] J. Thyssen, H. Nielsen, e S. D. Hansen, "Quantization of non-linear predictors in speech coding", Proc. ICASSP'95, IEEE Intern. Conf. Acoust., Speech, Signal Process., págs. 265-268, 1995.

[32] J.F.F.L.Dantas, "Codificação da Voz Utilizando o Modelo Multipulso Através de Análise Aprimorada do Sinal", Dissertação de Mestrado, Departamento de Engenharia Elétrica, Universidade de Brasília, jul. 1994.

[33] D.O'Shaughnessy, *Speech communications, humam and machine*, IEEE Press, 2000.

[34] E. Moulines, e R. Di Francesco, "Detection of the glottal closure by jumps in the statistical properties of the speech signal", Elsevier Science Publisher B.V., North-Holland, Speech Communication, vol. 9, págs. 401 – 418, 1990.

[35] I.A. Gerson e M. Jassuik, "Vector sum excited linear prediction (VSELP) speech coding at 8 kb/s", Proc. ICASSP'90, IEEE Int. Conf. Acoust., Speech and Signal Processing, Albuquerque, págs. 461-464, abril 1990.

[36] KPN, The Netherlands, "Improvement of the P.861 perceptual speech quality measure", ITU-T Contribution COM12-20-E, study period 1997–2000.

[37] Royal PTT, The Netherlands, "Correlation Between the PSQM and the subjective results of ITU-T 8 kbits/s 1993 speech codec test", ITU-T contribution COM 12-31, Geneva, set. 1994.

[38] I.H.S.P. Fantini e L.G.P. Meloni, "Otimização dos Codificadores VSELP e EFR por Refinamento da Modelagem Autoregressiva", XVIII Simpósio Brasileiro de Telecomunicações, Gramado, RS, set. 2000.