1. Introdução

Os codificadores de sinais de voz são sistemas capazes de representar sinais de voz com uma quantidade reduzida de bits, para fins de transmissão ou armazenagem.

Entre os muitos tipos de sistemas de codificação de sinais de voz, os codificadores de predição linear com excitação por dicionário de códigos (CELP) são os que possiblitam a melhor qualidade de voz para uma taxa de transmissão entre 4 kbits/s e 16 kbits/s., Contudo esses sistemas apresentam uma complexidade muito grande em função das buscas empreendidas no dicionário de códigos. Por esta razão, a maior parte do esforço de pesquisa em codificação de sinais de voz se concentra nos sistemas CELP e seus derivados.

O objetivo deste trabalho se refere à implementação de um codificador CELP, utilizando ferramentas modernas de treinamento de um dicionário de códigos conhecidas como técnicas de quantização vetorial. Além disso, o codificador CELP é também submetido a testes de medida de qualidade e de complexidade computacional.

No segundo capítulo, são introduzidos os princípios do processamento digital de sinais de voz, como as características básicas dos referidos sinais, a teoria da amostragem e as técnicas de quantização.

No terceiro capítulo, apresentamos os principais codificadores digitais de sinais de voz adotados nos principais sistemas de comunicações, classificando-os em codificadores de formato de onda, codificadores paramétricos e codificadores híbridos, descrevendo suas principais aplicações. Os codificadores de formato de onda como o PCM e o DPCM são sistemas bastante utilizados, com pouca complexidade e que apresentam uma taxa de transmissão na faixa entre 16 e 64 kbits/s. Os codificadores paramétricos como o LPC são sistemas de complexidade média e taxas de transmissão na faixa de 2.4 kbits/s. Os codificadores híbridos como o CELP são sistemas que misturam princípios dos codificadores de formato de onda com técnicas dos sistemas paramétricos, que apresentam grande complexidade computacional e taxas de transmissão na faixa de 4 a 16 kbits/s.

O quarto capítulo apresenta uma implementação de um sistema de codificação de predição linear (LPC), com as descrições dos diversos blocos que compõem um sistema dessa natureza. O funcionamento de um codificador LPC é descrito de forma a mostrar o processamento do sinal em cada bloco envolvido no sistema. Neste capítulo, é introduzido o problema da predição linear e demonstrada sua solução através do método da autocorrelação. Ainda é analisado o problema da detecção do *pitch*, que corresponde a algumas características dos sinais de voz, e implementado uma função para a detecção do *pitch*.

No quinto capítulo, apresentamos uma implementação de um codificador CELP, composto de diversos blocos, sendo que alguns destes são os mesmos

utilizados no codificador LPC. Além disso, descrevemos o processo de montagem de um dicionário de códigos através de dois métodos de quantização vetorial, o LBG modificado e uma nova técnica. Em seguida, descrevemos o funcionamento do codificador CELP implementado no que concerne a busca exaustiva pela excitação ótima e as várias etapas do processamento do sinal original.

No sexto e último capítulo, realizamos um conjunto de testes com vistas a avaliar o codificador CELP implementado no que se refere à qualidade da voz e à complexidade computacional. Finalmente, construimos gráficos ilustrando o comportamento no tocante à qualidade e à complexidade computacional do codificador implementado para dicionários de códigos e segmentos de voz de tamanhos diferentes.

2. Princípios de Processamento de Sinais de Voz

A codificação ou compressão de voz consiste na obtenção de maneira compacta de representações digitais de sinais de voz com o objetivo de transmissão ou armazenamento.

Os sinais de voz são limitados em banda a uma faixa entre 200 e 3400 Hz, e são amostrados em 8 kHz. Um codificador de voz deve representar estes sinais de voz com o menor número possível de bits, produzindo sinais de voz reconstituídos com qualidade satisfatória. Em geral, há um compromisso entre a taxa de transmissão do codificador e a qualidade do sinal de voz reconstituído.

2.1 As Propriedades Básicas da Voz

A voz é produzida a partir do ar forçado dos pulmões por meio das cordas vocais e do canal de voz humano (*vocal tract*). O canal de voz humano se estende da abertura das cordas vocais, também denominada glote, até a boca, que no homem comum chega a alcançar 17 centímetros de comprimento.

O canal vocal introduz correlações de curta duração (da ordem de 1ms) no sinal de voz, e podem ser interpretados como um filtro com ressonâncias chamadas de formantes. As freqüências destes formantes são controladas variando-se o formato das vias em questão. Por exemplo, movendo-se a posição da língua e dos lábios.

Uma parte importante dos codificadores de voz é a modelagem do canal de voz humano como um filtro de curtos períodos de duração, ou seja, costuma-se dividir um sinal de voz em segmentos menores. Como o formato deste filtro tem variação relativamente lenta, a função de transferência deste modelo de filtro precisa ser atualizada em perídos não muito curtos da ordem de 20ms.

O filtro do canal de voz humano é excitado pelo ar forçado dentro deste através das cordas vocais. Os sons de voz podem ser classificados em três classes dependendo do seu modo de excitação.

Sinais típicos de voz ou vozeados (*voiced sounds*) são produzidos quando as cordas vocais vibram abertas e fechadas, interrompendo ou não o fluxo de ar dos pulmões para o canal de voz humano e produzindo pulsos quase periódicos. A taxa de abertura e fechamento dá a característica (*pitch*) do som. Esta característica (*pitch*) pode ser ajustada variando-se o formato das cordas vocais e a pressão do ar nas cordas vocais. Estas classes de som possuem um alto grau de periodicidade do pitch, tipicamente da ordem de 2 a 10ms. Este comportamento de longa duração pode ser visto na figura 1, que mostra um segmento de voz amostrado a 8 kHz. Neste exemplo o período de pitch é de cerca de 8ms ou 64 amostras.

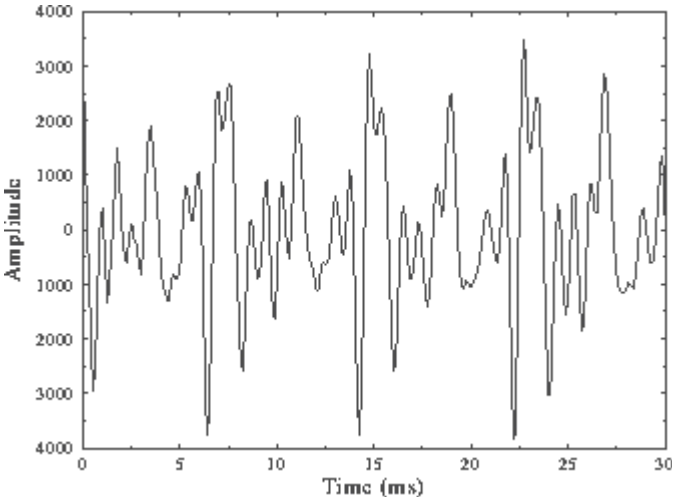

Figura 1 : Segmento típico de sinal de voz (*voiced*)

Sinais de voz não vozeados (*unvoiced*) são resultados de excitações semelhantes ao ruído branco gaussiano produzidas pela passagem do ar a grandes velocidades pelo canal de voz humano, quando a glote está aberta. Estes sons apresentam pouca periodicidade de longa duração como pode ser visto na figura 2, ainda que as correlações de curta duração estejam presentes.

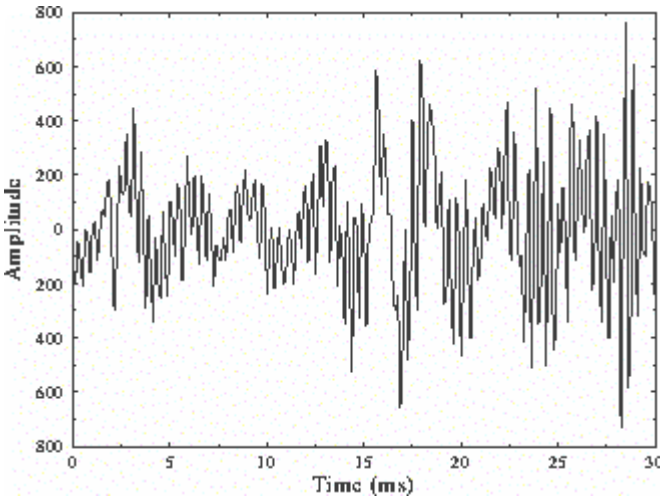

Figura 2 : Segmento típico de sinais de voz não vozeados

Sinais de voz do tipo "*plosive*" são resultado do fechamento do canal de voz humano que com o aumento da pressão do ar vindo dos pulmões são repentinamente liberados. Alguns tipos de sinais de voz podem ser classificados como qualquer um dos três tipos acima, mas na verdade são uma mistura.

Em geral, o formato do canal de voz e o seu modo de excitação mudam de forma lenta, e da mesma maneira os sinais de voz podem ser considerados aproximadamente quase estacionários desde que considerados em curtos períodos de tempo (20ms). É possível observar nas figuras 1 e 2 que os sinais de voz apresentam um certo grau de previsibilidade, devido às vibrações quase estacionárias das codas vocais e às ressonâncias do canal de voz humano. Os codificadores de voz exploram esta previsibilidade para reduzir a taxa de dados necessária a reproduzir o sinal de de voz com um bom nível de qualidade.

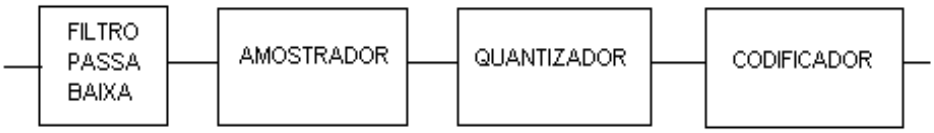

Figura 3: Sistema de codificação.

2.2 Teoria da Amostragem

Para se converter um sinal analógico em um sinal digital devemos limitar o sinal de interesse em banda, através do uso de um filtro passa-baixas. Este processo chamado de pré-filtragem é para satisfazer o critério de Nyquist. De acordo com Nyquist, a freqüência de amostragem deve ser maior que duas vezes a maior freqüência do sinal. Na prática taxas de amostragem mais altas são usadas para filtros não ideais. Em seguida, amostramos o sinal conforme mostrado na figura 4.

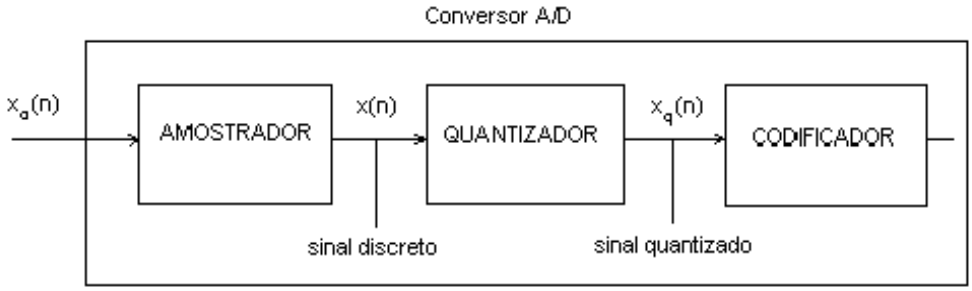

Figura 4: O processo de amostragem de um sinal.

Os padrões de sinais de áudio apresentam componentes em freqüências de até 20 kHz, e que são amostrados a 44.1 kHz no caso de CD e 48 kHz no caso de DAT. Os sinais de voz transmitidos por linhas telefônicas estão na faixa de 300-3400 Hz, sendo amostrados a uma taxa de 8 kHz.

2.3 Quantização

O processo de quantização é a conversão de um sinal discreto com valores contínuos em um sinal discreto com valores discretos, isto é, um processo de aproximação. O valor de cada amostra do sinal é selecionado a partir de um conjunto finito de valores possíveis.

A diferença entre a entrada não quantizada e a saída quantizada é chamada erro de quantização ou ruído de quantização, e é desejável que se consiga minimizar a magnitude deste erro. Para minimizar o erro de quantização várias técnicas de quantização podem ser usadas como por exemplo, quantização uniforme, logarítmica, não uniforme e vetorial.

O objetivo é fazer com que o tipo de quantização escolhida se adapte bem às estatísticas do sinal de voz de maneira que se consiga um desempenho ótimo,

com a maior qualidade e menor taxa de bits possíveis. Embora, na prática a eficiência do quantizador aumenta conforme a sua complexidade e custo.

2.3.1 Quantização Linear

Quantizadores lineares ou uniformes são aqueles em que a distância entre todos os níveis de reconstrução são iguais. Como este tipo de quantizador não altera o seu funcionamento de acordo com o tipo de sinal quantizado, geralmente não se obtém a melhor performance em termos perceptuais. No entanto, estes quantizadores são simples e de baixos custos para se implementar. Por exemplo, para se quantizar voz em sistemas telefônicos com qualidade se adota um quantizador linear com 13 bits.

2.3.2 Quantização Logarítmica

Sinais de voz podem ter uma faixa dinâmica maior que 60 dB, de forma que um grande número de níveis de reconstrução seria necessário se quiséssemos atingir uma boa qualidade usando quantização linear. No entanto, a resolução do quantizador é mais importante para partes de baixa amplitude do sinal que para altas amplitudes do sinal , o que nos leva a crer que um quantizador linear necessitaria de muitos níveis de reconstrução, e portanto, de mais largura de banda.

Podemos melhorar o esquema de quantização se conseguirmos aumentar a distância entre os níveis de reconstrução na medida em que a ampitude do sinal aumenta. Um método simples de se conseguir isto é aplicar o sinal a um compressor com características logarítmicas antes da quantização. Então, o sinal comprimido pode ser uniformemente quantizado. Na saída do sistema o sinal é passado por um expansor, cuja característica de transferência é a inversa do compressor. Esta técnica é conhecida por *companding*. As grandes vantagens são a simplicidade de implementação e boa performance.

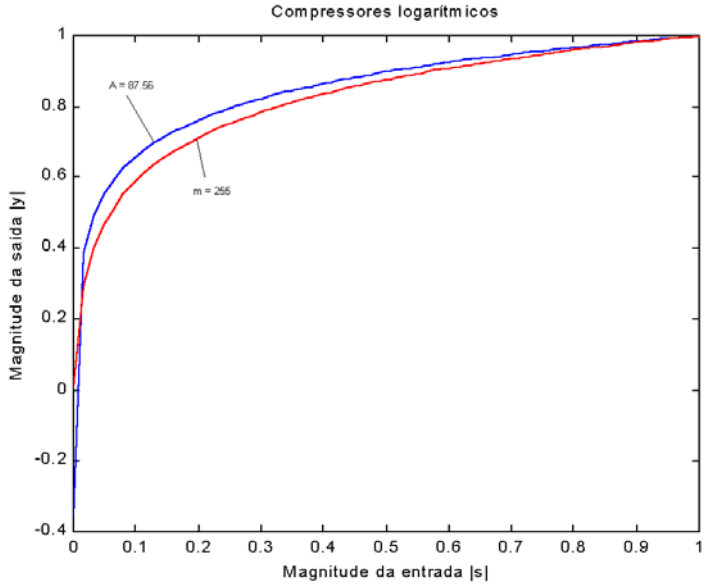

Figura 5 : compressores (a) μ-*law* e (b) *A-law.*

Duas características de *companding* muito conhecidas são o μ-*law* e o *A-law*. Estes dois tipos são bastante similares e as suas características de transferência são mostradas nas figuras acima. Para a maioria dos sistemas telefônicos A vale 87.56 e μ vale 255, como na figura 5. Usando esses esquemas é possível obter voz de qualidade usando um quantizador logarítmo de 8 bits.

2.3.3 Quantização não uniforme

O problema com a quantização uniforme é que na medida em que a amplitude do sinal diminui a relação sinal-ruído também diminui. Este problema é parcialmente resolvido pelo quantizador logarítmico, mas caso seja conhecida a função densidade de probabilidade (pdf) da entrada do sinal se podería relacionar a pdf aos níveis de reconstrução, resultando em uma minimização do erro de quantização.

Na prática uma estimativa da pdf pode ser usada para projetar os quantizadores. Isto pode ser obtido através de uma grande quantidade de informações a ser quantizada. Técnicas iterativas podem ser usadas para se obter os níveis de reconstrução a partir destas informações.

2.3.4 Quantização Vetorial

Nos métodos anteriores cada amostra era quantizada de forma independente das amostras vizinhas. A teoria da distorção de taxas afirma que esta não é a forma mais eficiente de se quantizar um sinal qualquer, mas quantizar as amostras em blocos de N amostras ou vetores. Este processo é apenas uma extensão dos métodos de quantização escalar vistos anteriormente.

Com a quantização escalar, a entrada é tratada como um número qualquer e arredondada para valores discretos previamente definidos. Por outro lado, com a quantização vetorial os blocos de N amostras são tratados como vetores N-dimensionais e quantizados para pontos pré-determinados no espaço N-dimensional.

A quantização vetorial sempre consegue melhor performance do que a escalar. No entanto, é mais sensível a erros de transmissão e geralmente envolve uma complexidade computacional bem maior do que a escalar.

3. Codificadoes de Sinais de Voz

Os codificadores de sinais de voz são usados para conferir uma maior eficiência na representação dos sinais de voz em sistemas digitais com largura de banda limitada, como por exemplo o sistema telefônico digital. Em geral, a voz humana é limitada numa faixa de freqüências de 200 a 3400 Hz e é amostrada em telefonia a 8 kHz. O codificador ideal deve representar os sinais com um número mínimo de bits, enquanto reproduz a voz original com a maior fidelidade possível. Na prática, sempre há um compromisso entre a taxa de bits do codificador e a qualidade da voz reproduzida.

Estes codificadores podem ser divididos em três classes: codificadores de formato de onda, codificadores paramétricos ou *vocoders* e codificadores híbridos. Os codificadores de formato de onda são usados a altas taxas de transmissão e produzem qualidade de voz muito boa. Os *vocoders* operam a taxas de bits bem pequenas, mas produzem em geral uma voz sintética. Os codificadores híbridos usam técnicas dos outros dois e produzem voz de boa qualidade a taxas de transmissão intermediárias. O gráfico da figura 5 mostra bem o compromisso que existe entre qualidade de voz e taxa de bits, além de situar as diferentes classes de codificadores de acordo com estes critérios.

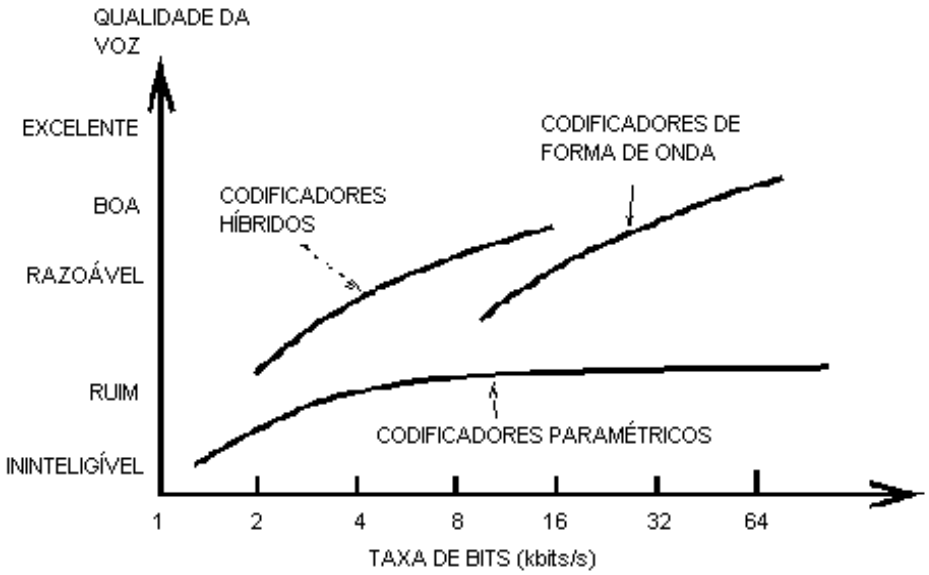

Figura 6 : Gráfico do compromisso qualidade versus taxa de bits.

3.1 Codificadores de formato de onda

Os codificadores de formato de onda procuram reproduzir o sinal de voz reconstruído cuja forma de onda seja a mais semelhante possível a do sinal original. Este codificadores não usam nenhum conhecimento prévio da natureza do sinal codificado, nem como o sinal foi gerado. Isto significa que estes dispositivos deveriam na teoria ser independentes do sinal e funcionar bem com quaisquer tipos de sinais.

Em geral, estes codificadores tem baixo nível de complexidade e produzem voz de alta qualidade a taxas de transmissão de aproximadamente 16 kbits/s. Quando se tenta diminuir a taxa de bits destes abaixo de 16 kbits/s, há uma degradação muito grande da qualidade da voz. A codificação de formas de onda pode ser realizada tanto no domínio do tempo quanto no domínio da freqüência. Os codificadores no domínio da frequênciia dividem o sinal em um número de componentes de freqüência e os codifica de forma independente, sendo que o número de bits usado pode variar de forma dinâmica.

3.1.1 PCM

A forma mais simples de codificação de formato de onda é a Pulse Code Modulation (PCM), que consiste simplesmente em amostrar e quantizar as formas de onda. A voz é limitada em banda na faixa de 4 kHz e amostrada a 8 kHz. Se usarmos quantização linear com 12 bits por amostra, consequimos produzir voz de boa qualidade a uma taxa de 96 kbits/s. Esta taxa de transmissão pode ser reduzida se usarmos quantização não uniforme nas amostras.

Em codificação de voz se usa sempre quantização logarítmica. Estes quantizadores produzem uma relação sinal ruído que é aproximadamente constante em uma faixa grande de níveis de entrada, e consegue produzir voz de excelente qualidade a uma taxa de 64 kbits/s. Estes quantizadores logarítmicos foram regulamentados na década de 60 e ainda são muito utilizados. Na América do Norte o padrão é o μ-*law* , enquanto que na Europa o padrão é o pouco diferente *A-law*. As vantagens destes quantizadores são a sua pequena complexidade e atraso, com alta qualidade de voz, enquanto que as desvantagens são a taxa de transmissão alta e suscetibilidade a erros nos canais de transmissão.

3.1.2 DPCM

Os codificadores do tipo Differential Pulse Code Modulation (DPCM) são codificadores de formato de onda que em vez de quantizar o sinal diretamente, como os codificadores PCM, quantizam a diferença entre o sinal de voz e uma estimativa feita do sinal de voz. Se esta estimativa ou previsão for eficaz, o erro do sinal entre as amostras previstas e as amostras reais terá uma variância menor que a das amostras originais, isto é, haverá uma maior eficiência no processo de quantização, resultando em menos bits que o sinal original. No decodificador, a diferença quantizada é somada ao sinal estimado de forma a resultar no sinal de voz reconstituído. Ao contrário de PCM, os codificadores DPCM, como o da figura 7, consideram a natureza do sinal a ser codificado, portanto, estes dispositivos não funcionam bem com outros tipos de sinal.

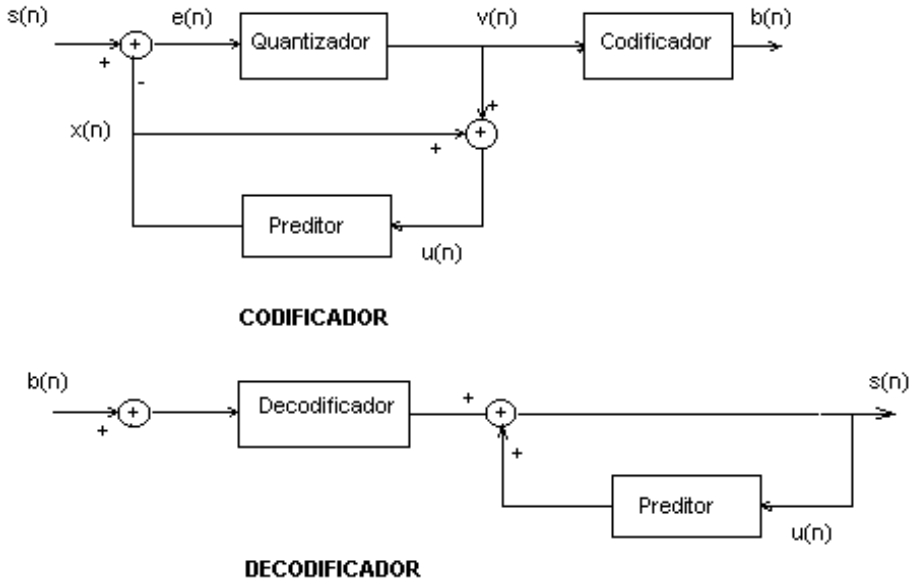

Figura 7: Sistema DPCM.

3.1.3 ADPCM

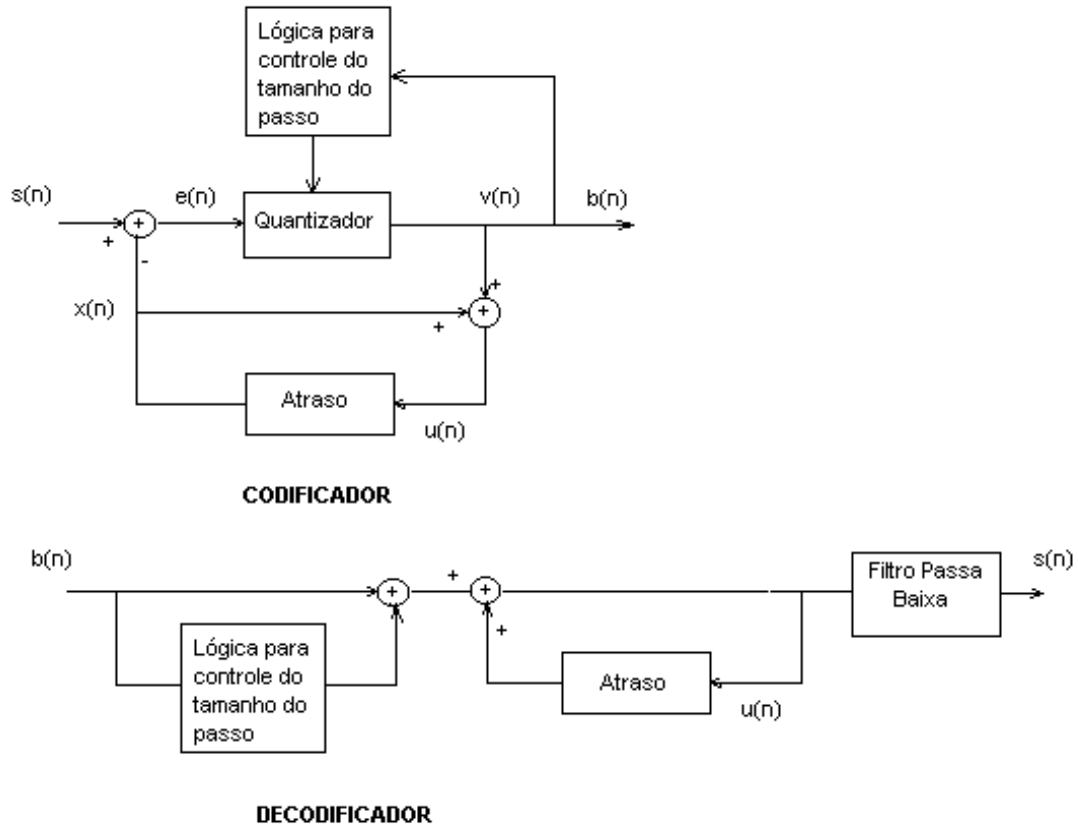

Figura 8: Sistema ADPCM.

Os codificadores Adaptive Differential Pulse Code Modulation (ADPCM), como o mostrado na figura 8, têm uma maior eficiência do que os DPCM, uma vez que o quantizador se adapta às estatísticas em mudança do resíduo estimado. Maiores ganhos podem ser conseguidos se o estimador conseguir se adaptar ao sinal de voz, o que poderia assegurar que o erro médio quadrático estaria sendo continuamente minimizado independentemente do sinal de voz.

Existem dois tipos de quantizadores e preditores adaptativos, chamados de *feedforward* e *feedbackward*. Com o *feedforward*, os níveis de reconstrução e os coeficientes de predição são calculados no transmissor, usando-se um bloco de voz. Estes coeficientes e níveis de reconstrução são quantizados e transmitidos ao receptor.

Para a adaptação *feedbackward* os níveis de reconstrução e os coeficientes do preditor são calculados a partir do sinal codificado. Como o sinal é conhecido pelo transmissor e receptor, não há necessidade de transmitir outro tipo de informação, de maneira que o preditor e o quantizador podem ser atualizados a cada amostra. Este tipo de estrutura pode produzir menores taxas de transmissão, contudo é mais sensível a erros que a técnica *feedforward*. A técnica ADPCM é bastante usada para voz codificada a taxas de transmissão médias. Existe um padrão CCITT para codificação telefônica que opera a uma taxa de 32 kbits/s, usando a técnica feed*backward* para ambos, quantizador e preditor. Neste caso o preditor tem dois pólos e seis zeros de forma que se consegue produzir boa qualidade para os sinais processados, inclusive aqueles que não são de voz.

3.1.4 Codificadores por Sub-Banda

A codificação por sub-bandas é a técnica mais simples no domínio da freqüência. Em cada codificador de sub-bandas, o sinal é passado por bancos de filtros. Em seguida, as sub-bandas são codificadas através de uma das formas descritas anteriormente no domínio do tempo. O número de bits atribuído a cada sub-banda pode ser variado de acordo com a importância perceptual da banda. No receptor as taxas de amostragem são aumentadas e as bandas são moduladas de volta às suas posições originais e somadas a fim de produzir a saída de voz.

A grande vantagem da codificação por sub-bandas, como na mostrada na figura 9, é que a quantização do ruído produzida em uma banda é confinada àquela banda. Isto evita que o erro de quantização se misture com outros componentes nas freqüências de cada banda. Isto significa que para cada banda podemos usar passos de quantização separados. Então bandas com baixas energias podem ter passos menores e portanto são preservadas no sinal de reconstrução. O confinamento do ruído de quantização também permite uma distribuição perceptual de bits. A codificação por sub-bandas é usada em transmissões de banda larga como por exemplo teleconferência.

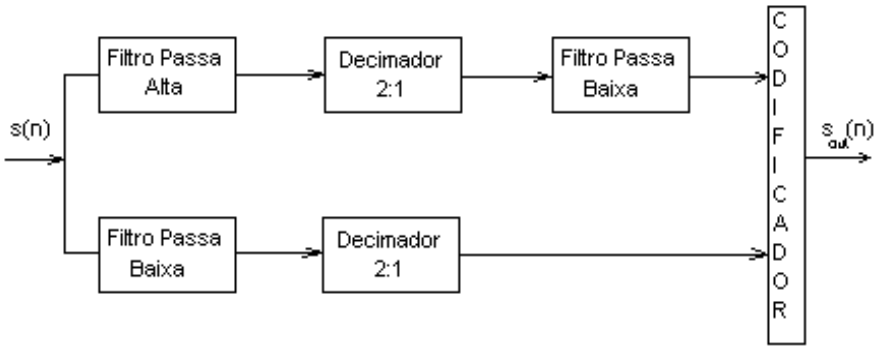

Figura 9: Um exemplo de codificação por sub-bandas.

3.2 Codificadores Paramétricos (*Vocoders*)

Os codificadores paramétricos ou *vocoders* operam usando um modelo baseado no canal de voz humano, extraindo do sinal sendo codificado os parâmetros deste modelo, que são transmitidos ao decodificador.

Os *vocoders* operam da seguinte maneira: o canal de voz humano é modelado por um filtro variante no tempo e excitado seja por uma fonte de ruído branco, que representa sinais do tipo não vozeado ou *unvoiced*, ou por um trem de pulsos separados por um período para sinais do tipo vozeado ou *voiced.* Desta forma, a informação a ser transmitida para o decodificador se constitui dos parâmetros do filtro, um *flag* ou aviso de voz vozeada ou não vozeada, o ganho desejado e o período de *pitch* para a voz. Estes dados são atualizados a cada 10-20 ms de modo a remediar a natureza não estacionária do sinal de voz.

Os parâmetros do modelo podem ser calculados pelo codificador através de diferentes maneiras, usando técnicas no domínio da freqüência ou do tempo. A informação também pode ser codificada para transmissão de várias maneiras diferentes. Os *vocoders* como o LPC operam a uma taxa de 2.4 kbits/s e produzem voz de baixa qualidade. A principal aplicação dos *vocoders* é em aplicações militares ou em aplicações que não requerem uma boa qualidade de voz, mas que precisam de taxas de tansmissão baixas de forma a permitir encriptação, por exemplo.

Os *vocoders* se baseiam em um modelo de produção de voz. Este modelo assume que a voz é produzida por um sistema linear, o canal de voz humano, alimentado por uma série de pulsos periódicos (*voiced*) ou ruído branco (*unvoiced*).

Caso a voz seja vozeada, a excitação consiste de uma série de pulsos periódicos, a distância ou o período entre estes pulsos é o período do *pitch*. Se a voz é do tipo não vozeada, a excitação é uma sequencia de ruído branco, correspondendo a um som característico do canal de voz humano.

Os *vocoders,* da mesma forma que os outros codificadores, buscam produzir sinais de voz que se parecem com os sinais originais. No transmissor, a voz é

analisada para determinar os parâmetros do modelo e a excitação. A informação é então transmitida ao receptor onde a voz é sintetizada. O resultado deste processo é que se consegue produzir voz inteligível a taxas de tansmissão baixas. No entanto, a voz sintetizada costuma ser pouco natural, fazendo com que o uso de *vocoders* seja restrito onde há exigência de baixas taxas de bits.

A baixa qualidade da voz processada por um *vocoder* é atribuída à natureza simples do modelo de produção de voz. O simples fato de considerarmos a voz vozeada ou não vozeada, não permite estágios intermediários. O ouvido humano é bastante sensível ao *pitch*, de modo que para voz do tipo vozeada, o *pitch* deve ser precisamente determinado e este problema ainda não foi resolvido satisfatoriamente. Com os *vocoders* há também o problema da sensibilidade aos erros no modelo, erros de cálculo dos parâmetros ou de transmissão.

3.2.1 Codificador Paramétrico de Canal

É o primeiro dos *vocoders*. Este codificador paramétrico usa em seu favor o fato de o ouvido humano ser insensível à fase de curta duração. Assim, para segmentos de voz de cerca de 20 ms, apenas a magnitude do espectro do sinal precisa ser considerada. O espectro é estimado com o uso de bancos de filtros. Obviamente, quanto mais filtros nesses bancos, melhor é a estimativa, mas também maior é a taxa de transmissão. A saída de cada um destes filtros é retificada e filtrada por um passa-baixas para se obter a envoltória do sinal. Em seguida o sinal é amostrado e transmitido.

As larguras de banda dos filtros usados nos bancos de filtros tendem a aumentar com a freqüência, uma vez que o ouvido humano responde de forma logarítmica no domínio da freqüência.O *vocoder* de canal, como o mostrado na figura 10, pode ser implementado com componentes digitais ou analógicos e produz uma qualidade de voz ruim a uma taxa de 2.4 kbits/s.

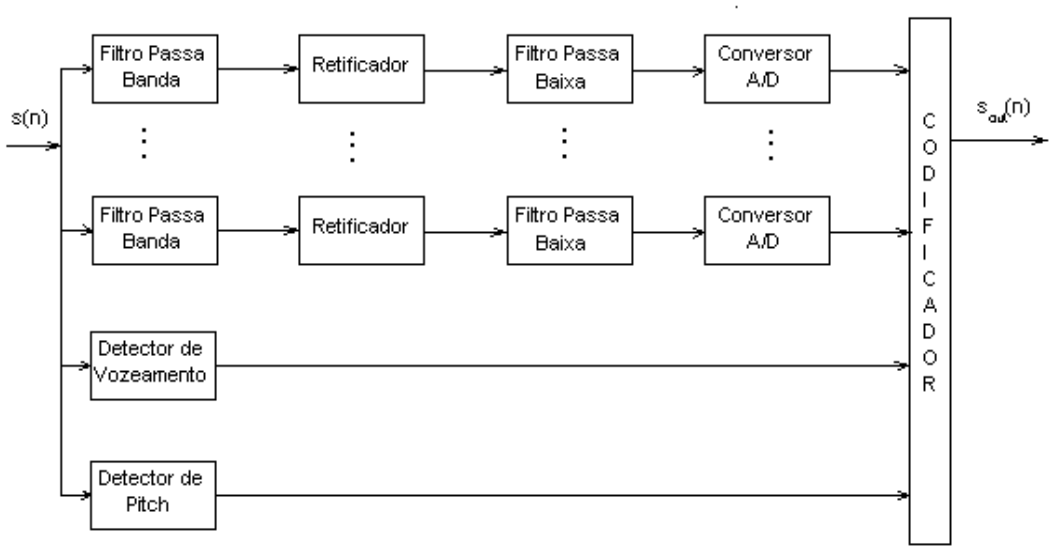

Figura 10: Codificador Paramétrico de Canal.

3.2.2 Codificador Paramétrico Homomórfico

Processamento de sinais do tipo homomórfico é uma classe geral de técnicas de processamento de sinais não lineares que podem tratar de forma eficaz sinais compostos. Este *vocoder* assume que o sinal de voz é uma convolução no tempo da resposta ao impulso do canal de voz e da função de excitação. Esta convolução é uma multiplicação no domínio da freqüência. Se o logaritmo do espectro for considerado a multiplicação se torna uma adição. Como o ouvido humano é insensível a fase do sinal, usa-se o logaritmo da magnitude do spectro: $\text{Log}(|S(e^{jw})|) = \text{Log}(|P(e^{jw})|) + \text{Log}(|V(e^{jw})|)$, onde $S(e^{jw})$ é o espectro do sinal de voz, $P(e^{jw})$ é o espectro da excitação e $V(e^{jw})$ é o espectro do canal de voz humano. Os coeficiente da análise *cepstral* para este sinal podem ser obtidos através da transformada inversa de Fourier do espectro logarítmico.

Em produção de sinais de voz, a resposta ao impulso do canal de voz humano é variada lentamente, enquanto a excitação varia rapidamente. Esta distinção é preservada nos coeficientes *cepstrais* e isto se manifesta como um deslocamento no tempo. Agora, as contribuições dos dois componentes de voz podem ser facilmente separadas se usando um filtro passa-baixas. Desta forma é relativamente simples determinar de forma precisa o *pitch*.

No *vocoder* homomórfico, os coeficientes *cepstrais* descrevendo o canal de voz humano são transmitidos juntamente com o *pitch* a uma taxa de aproximadamente 4 kbits/s.

3.2.3 Codificador Paramétrico de Predição Linear

O codificador de predição linear (LPC *vocoder*) usa o modelo de produção de voz descrito na figura 11, isto é, o mesmo dos outros *vocoders*. Este *vocoder* apenas difere dos outros no método usado para determinar o modelo do canal de voz humano. Este codificador assume que o canal de voz humano pode ser descrito ou modelado por um filtro de apenas pólos com resposta ao impulso infinita (filtro *all-pole* IIR).

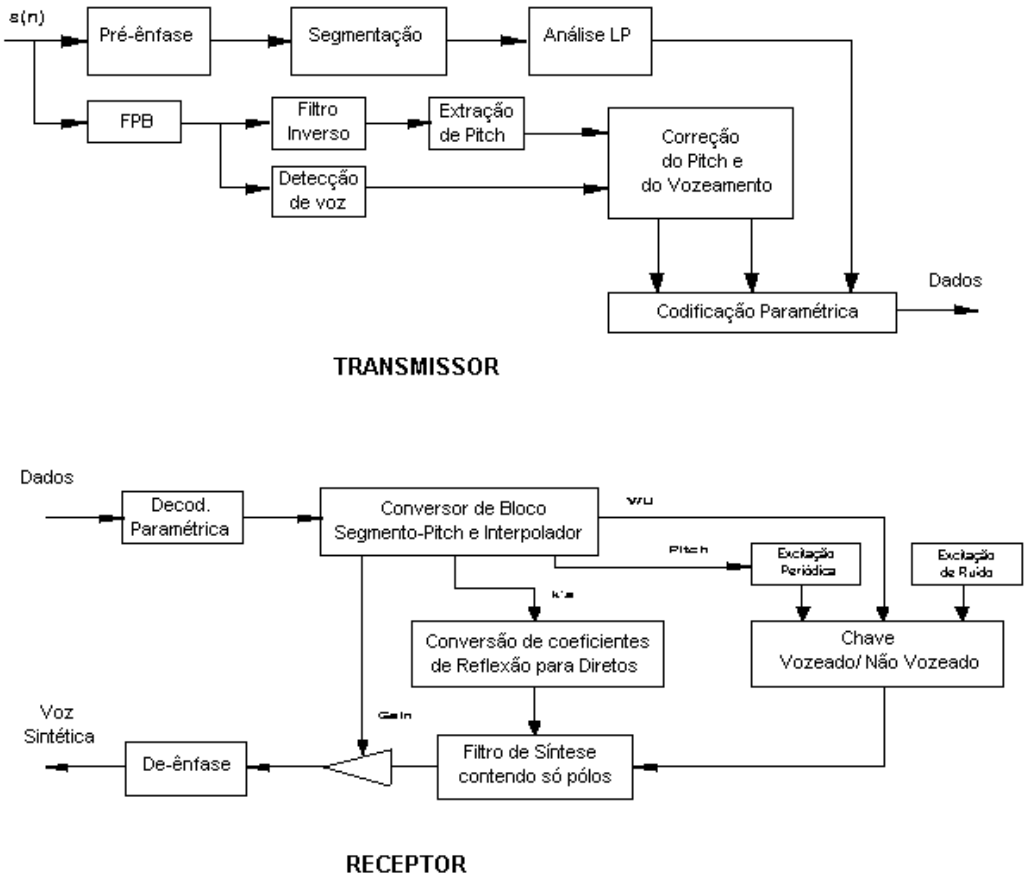

Figura 11 : Modelo de codificador LPC.

Em outras palavras, cada amostra de sinal de voz é interpretada como uma combinação linear das amostras anteriores. Os coeficientes deste filtro de apenas pólos são calculados pela análise LPC, de maneira a minimizar o erro médio quadrático entre a estimativa e a amostra verdadeira.

No codificador, um bloco de voz de cerca de 20 ms é armazenado e analisado para determinar os coeficientes do preditor. Estes coeficientes são quantizados e transmitidos ao receptor. Então, esta voz é passada por um filtro inverso ao filtro que modela o canal de voz humano, a fim de obter o erro ou resíduo de predição. O efeito prático do preditor é remover a correlação entre as amostras adjacentes, e isto faz com que seja mais fácil obter o período do *pitch* de voz, uma vez que a correlação entre as amostras fica mais perceptível em perídos de tempo mais longos. Portanto, uma decisão de melhor performance e qualidade pode ser tomada acerca do tipo de voz, vozeada ou não vozeada, usando o resíduo.

Os codificadores LPC são os mais usados entre os diversos vocoders existentes, uma vez que que o seu modelo de apenas pólos do canal de voz humano funciona de forma bastante eficiente. A taxa de transmissão é de cerca de 2.4 kbits/s e a qualidade de voz não é muito boa, com características um pouco sintéticas no som.

Na prática, os codificadores LPC transmitem os parâmetros calculados na análise LPC, o ganho, o *pitch* e um *flag* ou aviso de segmento vozeado ou não vozeado. Um algoritmo baseado em LPC muito utilizado é o LPC-10, que usa 10 coeficientes e é caracterizado por:

- Taxa de amostragem de 8 kHz;
- 180 amostras/quadro, 44.44 quadros/segundo;
- 2 coefientes são quantizados como log area ratios com 5 bits cada;
- 8 coeficientes de reflexão, sendo que o n° de bits diminui com o índice até 2 bits – 41 bits;
- 7 bits para o *pitch* e decisão do tipo de voz;
- 5 bits para o ganho;
- Total : 54 bits por quadro , aproximadamente 2400 bps.

3.3 Codificadores Híbridos

Os codificadores híbridos preenchem o espaço vazio deixado pelos codificadores de formato de onda e codificadores paramétricos. Como foi dito anteriormente, os codificadores de formato de onda são capazes de produzir boa qualidade de voz a taxas de transmissão de até 16 kbits/s, mas são muito limitados a taxas inferiores a estas. Por outro lado, os codificadores paramétricos produzem voz de baixa qualidade a taxas de 2.4 kbits/s e abaixo, mas não conseguem produzir voz de forma muito natural.

Apesar de existir outras formas de codificação híbrida, a mais usada é a codificação de análise por síntese (*Anaysis by Synthesis* - AbS) no domínio do tempo. Estes codificadores usam o mesmo modelo de filtro de predição linear do canal de voz humano usado nos LPC *vocoders*. No entanto, em vez de aplicarem um modelo de dois estados, vozeado ou não vozeado, para achar a excitação do filtro, o sinal de excitação é escolhido de forma que o sinal de voz reconstruído seja o mais parecido possível com a forma de onda do sinal original. Os codificadores de análise por síntese (AbS) foram introduzidos na forma que é conhecida atualmente *por Multi-Pulse Excited Coder* (MPE). Em seguida, foram introduzidos o *Regular Pulse Excited Coder* (RPE), o *Residual Excited Linear Predictive Coder* (RELP) e o *Code Excited Linear Predictive Coder* (CELP). Um modelo genérico de um sistema AbS é mostrado na figura 12.

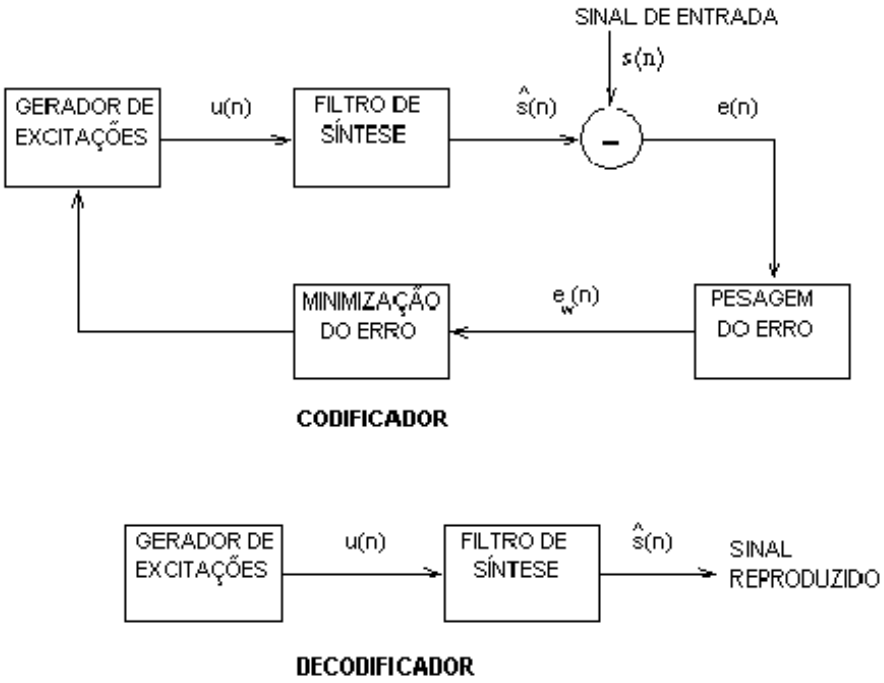

Figura 12 : Estrutura de um sistema de análise por síntese.

Os codificadores AbS dividem o sinal de voz de entrada de forma a codificá-lo em quadros, tipicamente da ordem de 20 ms. Os parâmetros de cada quadro são determinados para um filtro de síntese, e em seguida a excitação deste filtro é determinada. Isto é realizado achando-se o sinal de excitação que passado em um dado filtro de síntese minimiza o erro médio quadrático entre o sinal de voz de entrada e o sinal reconstituído. Por este motivo é dado o nome análise por síntese a estes codificadores, uma vez que o codificador analisa o sinal de entrada, sintetizando várias aproximações para este. Então para cada quadro, o codificador transmite informação representando os parâmetros do filtro de síntese e a excitação ao decodificador, enquanto no decodificador a dada excitação é aplicada a um filtro de síntese que produz o sinal reconstitúido. O filtro de síntese é geralmente um filtro de apenas pólos, de curta duração e linear da seguinte forma:

$$H(z) = \frac{1}{A(z)}$$

onde

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

é o filtro de predição de erro determinado pela minimização da energia do sinal residual produzido quando o segmento de voz original é passado por este filtro. A ordem *p* do filtro é tipicamente 10. Este filtro tem por objetivo modelar as correlações introduzidas no sinal de voz pela ação do canal de voz humano.

O filtro de síntese pode também incluir um filtro de *pitch* para modelar as periodicidades de longa duração presentes na voz do tipo vozeada. Alternativamente, estas periodicidades de longa duração podem ser exploradas

incluindo um *codebook* adaptativo, isto é um dicionário de códigos com um conjunto de seqüências gaussianas, no gerador de excitação, de forma que o sinal de excitação *u(n)* inclua uma componente da forma $G.u(n-\alpha)$, onde α é o período de *pitch* estimado. Geralmente, os codificadores MPE e RPE funcionam sem o filtro de *pitch*, ainda que a sua performance possa ser melhorada caso este filtro seja usado. Para os codificadores CELP, no entanto, o filtro de *pitch* é extremamente importante.

O bloco de pesagem do erro é usado para modelar a forma do espectro do sinal de erro, de maneira a diminuir a contribuição deste sinal. Isto é possível porque o sinal de erro nas regiões de freqüência onde a voz tem alta energia será mascarado pela voz. O filtro de pesagem enfatiza o ruído nas regiões de freqüência onde o conteúdo de voz é pequeno. Então, a minimização do erro pesado concentra a energia do sinal de erro em freqüências onde o sinal de voz tem altas energias. Desta maneira, o sinal de erro será ao menos parcialmente mascarado pelo sinal de voz, e sua importância relativa será reduzida. Este tipo de pesagem é usado a fim de produzir uma melhora significativa na qualidade do sinal reconstruído para este tipo de codificador.

A característica que distingüe os codificadore AbS é a maneira que o sinal de excitação *u(n)* aplicado ao filtro de síntese é escolhido. Conceitualmente, cada forma de onda possível é passada pelo filtro para se verificar que sinal de voz seria produzido por uma dada excitação. Esta excitação que produz o erro mínimo pesado entre o sinal original e o sinal reconstituído é escolhida pelo codificador e usada para excitar o filtro de síntese no decodificador. É justamente esta determinação através de malha fechada da excitação que permite os codificadores AbS produzir voz de qualidade a taxas de transmissão baixas. No entanto, a complexidade envolvida em aplicar todas as excitações pelo filtro de síntese é enorme. Portanto, alguns meios de reduzir esta complexidade, sem reduzir a qualidade de voz, devem ser achados.

3.3.1 Codificadores Multi-Pulso

Como indicado anteriormente, as diferenças entre os codificadores MPE, RPE e CELP surgem a partir da representação do sinal *u(n)* usado. Em codificadores multi-pulso, *u(n)* é dado por um número finito de pulsos não nulos para cada quadro de voz. As posições destes pulsos não nulos dentro do quadro, assim como suas amplitudes, devem ser determinadas pelo codificador e transmitidas ao decodificador.

Na teoria, seria possível encontrar os melhores valores para todas as posições e amplitudes de pulsos, mas isto não é prático em razão do aumento de complexidade envolvido. Na prática, algum método sub-ótimo de encontrar as posições e amplitudes dos pulsos deve ser usado. Tipicamente cerca de 4 pulsos a cada 5 ms são usados, e isto nos leva a uma boa qualidade de voz a 10 kbits/s.

3.3.2 Codificadores RPE

De forma similar ao codificador multi-pulso, o codificador RPE usa um determinado número de pulsos não nulos para produzir a excitação *u(n).* Entretanto, nos codificadores RPE os pulsos são regularmente espaçados com um intervalo fixo, e o codificador precisa apenas da posição do primeiro pulso e a amplitude de todos os pulsos.

Desta maneira, há uma necessidade de menos informação a ser transmitida acerca da posição dos pulsos, e para uma dada taxa de transmissão o codificador RPE é capaz de usar muito mais pulsos que o codificador MPE. Por exemplo, a uma taxa de 10 kbits/s cerca de 10 pulsos a cada 5 ms podem ser usados nos codificadores RPE, em comparação com 4 pulsos para os codificadores MPE. Isto permite que os codificadores RPE consigam qualidade de voz superior aos MPE, ainda que os codificadores RPE sejam mais complexos. Os codificadores RPE são usados no sistema europeu de telefonia móvel a uma taxa de 13 kbits/s .

3.3.3 Codificadores RELP

Quando um sinal de voz é passado por um preditor linear, a correlação entre as amostras é retirada. Então, se a predição é satisfatória, a saída do preditor deverá se aproximar do ruído branco, com uma forma espectral do tipo plana. As seções dos resíduos espectrais são muito semelhantes entre si. No entanto, o resíduo contém toda a informação da excitação, assim como a informação que o preditor linear omitiu. No codificador RELP, a idéia é que uma pequena porção deste ruído seja transmitido, e desta forma, idealmente todo o resíduo pode ser reconstruído no receptor.

No codificador RELP o resíduo é aplicado a um filtro passa-baixas com uma freqüência de corte de 1 kHz. A saída deste filtro é codificada usando uma das formas de codificação de formas de onda. No receptor, o resíduo é reconstruído copiando-se o resíduo de banda base para outras freqüências consideradas.

A aparência fina do espectro da voz, devido ao *pitch*, não é removida pelo preditor linear. Esta é uma das grandes desvantagens do codificador RELP. É improvável que a freqüência de corte dos filtros passa-baixas corresponda a algum harmônico da freqüência do *pitch*. Isto significa que quando o resíduo é reconstruído no receptor, a informação perceptual do *pitch* será incorreta para altas freqüências. Este problema pode ser aliviado, adaptando-se a freqüência de corte do filtro passa-baixas para a freqüência do *pitch*. Os codificadores RELP geralmente produzem qualidade de voz de boa qualidade a taxas de 9.6 kbits/s.

3.3.4 Codificadores CELP

Ainda que os codificadores MPE e RPE consigam produzir voz de qualidade a uma taxa de 10 kbits/s e acima, estes não são adequados a taxas muito abaixo destas. Isto se deve à grande quantidade de informação transmitida a respeito

das posições e amplitudes dos pulsos. Ao reduzir a taxa de bits usando menos pulsos, ou quantizando as suas amplitudes, a qualidade de voz se degrada rapidamente.

Atualmente, o algoritmo mais usado para produzir boa qualidade de voz a taxas inferiores a 10 kbits/s é o CELP. Esta técnica difere das codificações MPE e RPE em que o sinal de excitação é quantizado vetorialmente. A excitação é dada por uma entrada de um *codebook* ou dicionário de códigos que quantiza os sinais vetorialmente, e o ganho que controla a sua potência. Em geral, o índice do *codebook* é representado por 10 bits, produzindo um *codebook* de 1024 entradas, e o ganho é codificado com 5 bits. Desta maneira, a taxa necessária para transmitir informação de excitação é bastante reduzida, em torno de 15 bits comparado a 47 bits no RPE.

Originalmente o *codebook* usado em codificadores CELP continha seqüências gaussianas brancas, uma vez que se assumia que os preditores de curta e longa duração seriam capazes de remover toda a redundância do sinal, de forma a produzir um resíduo semelhante a um ruído aleatório. Também foi demonstrado que a função densidade de probabilidade (pdf) deste resíduo era quase gaussiana. Mais tarde, descobriu-se que o uso de um *codebook* para produzir excitação de curta e longa duração poderia produzir voz de alta qualidade.

No entanto, escolher a entrada do *codebook* a ser usada em um procedimento de análise por síntese significa que cada excitação tem de ser aplicada aos filtros de síntese a fim de verificar a semelhança do sinal reconstruído com o sinal original. Isto significa que a complexidade do codificador CELP original era muito grande para se implementar em tempo real. Desde 1985, muita pesquisa foi feita com o objetivo de reduzir esta complexidade, principalmente se alterando a estrutura do *codebook*.

O princípio de codificação CELP conseguiu produzir qualidade de voz satisfatória para comunicações a taxas entre 4.0 e 16 kbits/s, como por exemplo o padrão da CCITT que consiste de um codificador de 16 kbits/s e o do Departamento de Defesa dos E.U.A. que padronizou um codificador de 4.8 kbits/s.

O codificador do Departamento de Defesa divide a voz em quadros de 30 ms, sendo que cada um é sub-dividido em 4 sub-quadros de 7.5 ms. Para cada quadro o codificador calcula um conjunto de 10 coeficientes para o filtro de síntese de curta duração que modela o canal de voz humano. A excitação deste filtro é determinada para cada sub-quadro, e é dada pela soma das entradas escaladas de 2 *codebook*s. Um *codebook* adaptativo é usado para modelar as periodicidades de longa duração presentes na voz do tipo vozeada, e para cada sub-quadro um índice e um ganho são determinados para este *codebook*. Um *codebook* fixo de 512 códigos pseudo-aleatórios também é acessado para encontrar a entrada do *codebook*, e o multiplicador de ganho para esta entrada, que minimiza o erro entre os sinais reconstituído e original. No decodificador, as entrada escaladas dos 2 *codebook*s são passadas por um

filtro de síntese para produzir a voz. Finalmente, a voz é aplicada a um pós filtro a fim de melhorar a sua qualidade perceptual.

Os codificadores CELP e seus derivados são usados a taxas abaixo de 16 kbits/s, e em razão da determinação dos coeficientes do filtro por métodos adaptativos, estes introduzem grandes atrasos. O atraso de um codificador de voz é definido como o período de tempo que uma amostra de voz sai do codificador e chega ao decodificador, assumindo que o bit stream ou corrente de bits do codificador é alimentado diretamente ao decodificador. Para um codificador híbrido típico, o atraso é da ordem de 50 a 100ms, e um erro de tal magnitude pode causar problemas no que diz respeito ao erro e ruído no canal de comunicação.

O codificador CELP de pequeno atraso padronizado pelo CCITT tem qualidade de voz e especificações de erro e ruído no canal comparáveis ao codificador PCM de 32kbits/s, mas com uma taxa de transmissão de 16kbits/s. Estas especificações foram atendidas em um codificador CELP adaptativo do tipo *backward* desenvolvido pelo AT&T Bell Labs, e padronizado em 1992 como G.728.

Este codificador usa adaptação do tipo *backward* para calcular os coeficientes dos filtros, achados a partir da voz reconstituída. Isto significa que o codificador pode usar um quadro menor que o usual. e o padrão G728 usa um quadro de 5 amostras produzindo um atraso de menos de 2 ms. Um preditor de curta duração de ordem 50 é usado, e isto elimina a necessidade de preditores de longa duração. Então os 10 bits usados e que estão disponíveis para cada vetor de 5 amostras a 16kbits/s são usados para representar a excitação fixa do *codebook*. Destes 10 bits, 7 bits são usados para transmitir índice fixo do *codebook*, enquanto os outros 3 bits são usados para representar o ganho da excitação. A adaptação do tipo *backward* é usada para auxiliar a quantização do ganho de excitação, e no decodificador um pós filtro é usado para melhorar a qualidade perceptual da voz reconstruída. O resultado é um codificador com um atraso inferior a 2 ms, taxa de 16kbits/s, boa qualidade de voz e grande robustez a erros de canal.

A estrutura do codificador CELP, como o mostrado na figura 10, pode ser melhorada e usada a taxas inferiores a 4.8 kbits/s, classificando-se os segmentos de voz como por exemplo vozeado ou não vozeado. Desta maneira, os diferentes segmentos de voz são codificados separadamente com um codificador especial para cada tipo. Por exemplo, para segmentos de voz não vozeados o codificador não usará predição de longa duração, enquanto para voz do tipo vozeada esta predição será usada, mas o *codebook* fixo se torna menos importante. Estes codificadores conseguem boa qualidade de voz a 2.4 kbits/s.

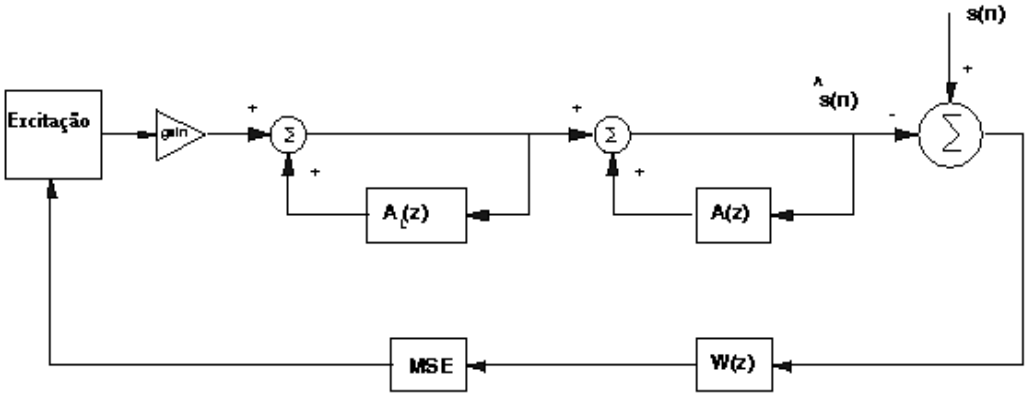

Figura 13 : Codificador CELP

4. Implementação de um Codificador LPC

4.1 Descrição do algoritmo LPC

O algoritmo implementado pelo alunos de codificação do grupo de processamento de sinais de voz é constituído de módulos básicos, contendo rotinas necessárias à implementação de um sistema de codificação e síntese LPC.

O sistema de codificação e síntese desenvolvido consiste de 11 blocos básicos, como mostrado na figura 14. Cada rotina é responsável por uma parte do processamento da voz original, sendo que uma outra rotina encarrega-se de realizar as chamadas de cada bloco de maneira a realizar o processamento da voz de maneira adequada.

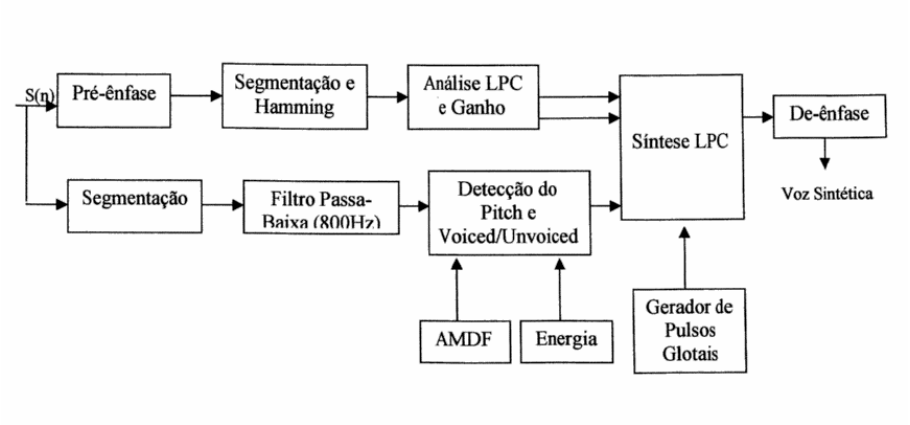

Figura 14. Sistema de codificação e síntese LPC.

Inicialmente, o sinal de voz amostrado é passado por um filtro de pré-ênfase. Em seguida, o sinal é dividido em segmentos cujo tamanho é escolhido pelo usuário, e aplicamos uma janela de *hamming* a cada segmento. Então, o sinal processado é entregue ao bloco que realiza a análise LPC e calcula o ganho.

Para obter o *pitch*, o sinal é primeiramente segmentado sem a aplicação da janela de *hamming* e aplicado a um filtro passa-baixa com freqüência de corte em 800 Hz. Em seguida, o sinal devidamente processado é entregue ao bloco de detecção do *pitch e* caracterização do tipo de voz (vozeado/não vozeado*)*, que com as informações provenientes dos blocos que calculam a energia e o AMDF do sinal, determina o pitch e o tipo de voz do segmento.

Os parâmetros do modelo de excitação usado na síntese LPC são: os coeficientes gerados na análise LPC, o ganho e o período de pitch. Estes parâmetros são usados no bloco de síntese LPC, que utiliza, como forma de excitação do filtro de síntese, pulsos obtidos no gerador de pulsos glotais para voz do tipo vozeada e ruído branco para voz não vozeada. Finalmente, a voz processada no sistema é reconstituída.

4.2 Filtro de Pré-Ênfase

A aplicação de um filtro de pré-ênfase ao sinal de voz original é necessária, uma vez que o filtro concentra a energia relativa ao espectro de alta freqüência do sinal, retirando a parte DC do sinal, introduzindo um zero perto de ω=0.

Além disso, há outras razões para se empregar o filtro de pré-ênfase. A primeira razão é a prevenção contra instabilidade numérica, sendo que os trabalhos nesta área focaram-se no método da autocorrelação. Assumindo que o sinal de voz é dominado por componentes em freqüências baixas, é bastante previsível que um modelo LPC de ordem elevada poderá resultar em uma matriz de autocorrelação contaminada. Chamamos de um sinal contaminado a um sinal que contém componentes indesejados ou que é dominado por ruído. Um filtro de primeira ordem deve ser capaz de aproximar o espectro do sinal ao espectro de um ruído branco. Outra razão é que o componente de fase mínima do sinal glotal pode ser modelado por um filtro simples de 2 pólos próximos a z=1. Então a característica dos lábios com o seu zero perto de z=1, tende a cancelar os efeitos espectrais de um dos pólos glotais. Introduzindo, um segundo zero próximo a z=1, as contribuições espectrais da laringe e dos lábios seriam eliminadas, fazendo com que a análise LPC correspondesse apenas ao canal de voz humano. No entanto, apesar do filtro de pré-ênfase, o espectro de predição linear não fica totalmente livre dos efeitos da laringe e dos lábios. Em geral, a pré-ênfase propicia aos primeiros formantes uma maior chance de influenciar a voz.

Desta maneira, para o projeto do filtro de pré-ênfase, usamos um filtro de ordem um(1) com o parâmetro μ variando de 0.9 a 1.0 ($0.9 \leq \mu < 1.0$), com a seguinte característica :

$$Hpe(z) = 1 - \mu z^{-1}$$

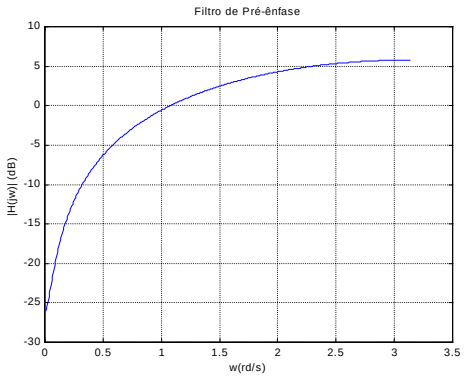

Figura. 15 Filtro de Pré-ênfase com μ=0.95.

4.3 Segmentação e Janelamento

Para aplicações de processamento de sinais é necessário selecionar uma porção do sinal que possa ser considerada estacionária para analisá-la. Os sinais de voz podem se considerados como estacionários quando analisados

em pequenos segmentos da ordem de 20 ms. Com isto, reduzem-se os efeitos do sinal de excitação na estimativa dos coeficientes do filtro de síntese, e obtêm-se uma melhor estimativa do espectro da voz.

O quadro ou segmento de análise, I, corresponde ao número de amostras que será usado para determinar os coeficientes da análise LPC. A razão I/L representa a taxa de superposição entre dois segmentos de análise adjacentes. Em nosso sistema usaremos uma taxa de 50% (I=L/2), como mostrado na figura 16.

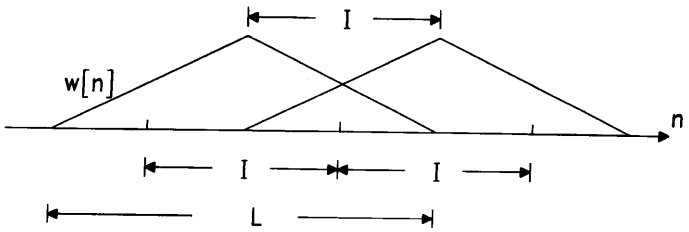

Figura 16. Superposição de dois segmentos de voz.

Outros conjuntos de parâmetros muito importante para a análise LPC são aqueles relacionados a aplicação das janelas. Estes parâmetros incluem o tipo e o tamanho do quadro de análise e da janela. São características espectrais desejáveis das janelas: uma largura de banda estreita no lóbulo principal e grande atenuação nos lóbulos laterais. Em geral, uma largura de banda estreita deve resolver os pequenos detalhes do sinal janelado, enquanto a grande atenuação nos lóbulos laterais deve evitar que o espectro do sinal seja corrompido pelo ruído de "*aliasing*".

No entanto existe um compromisso na escolha do tipo de janela. A janela retangular preserva as características temporais do sinal, mas acarreta um truncamento de maneira abrupta do sinal nas extremidades. As janelas de Hamming, Hanning, Blackman e Kaiser apresentam características de truncamento mais suaves nas extremidades, com uma maior distorção do sinal no domínio do tempo.

Usando o método da autocorrelação, a janela de nossa escolha, Hamming, é continuamente aplicada ao sinal de voz. Em geral, usamos janelas de Hamming ou de Hanning que possuem um caimento suave a fim de reduzir os efeitos de truncamento nas extremidades do segmento. As janelas com caimento mais suave costumam produzir melhores resultados do que janelas triangulares ou retangulares. A equação característica da janela de Hamming de ordem N é:

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi n}{N-1}\right), & n = 0,1,...,N-1 \\ 0, & \text{caso contrário} \end{cases}$$

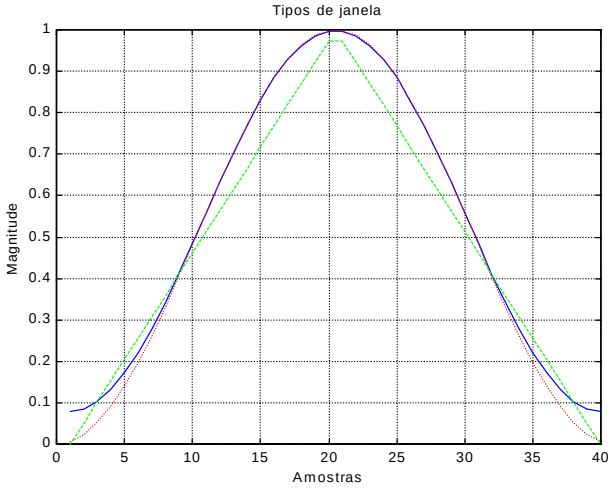

Figura 17. Tipos de janela: Hamming (azul), Hanning (verde) e triangular (vermelho).

4.4 Análise LPC

Na análise LPC, o modelo do preditor linear assume que o sinal de voz é um processo autoregressivo descrito pela seguinte equação:

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + gu(n) \ ,$$

onde s(n) é um sinal de voz produzido sinteticamente pelo modelo, u(n) é o sinal de excitação, a_i são os coeficientes de predição, p a ordem do preditor e g é o parâmetro de ganho que é usado para casar a energia da voz sintética com a da voz original.

O modelo de síntese pode ser representado da seguinte maneira:

$S(z) = \dfrac{g}{1 - A(z)} X(z)$, onde g representa o ganho e $A(z) = \sum_{i=1}^{p} a_i z^{-i}$.

$$e(n) = s(n-p) - \sum_{i=1}^{p} a_i s(n-p+i)$$

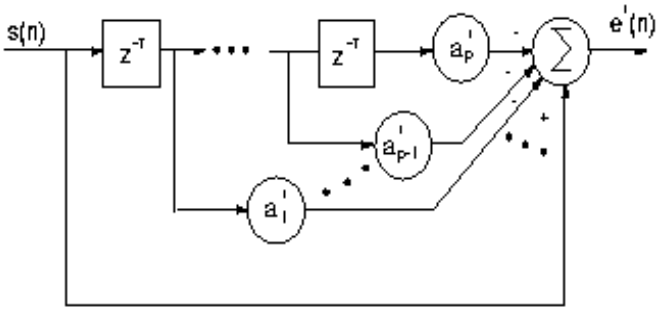

Figura 18. Realização de um filtro de análise de predição linear.

Na análise LPC de sinais de voz, os parâmetros dos modelos de excitação e do canal de voz humano são aproximados pelo sinal de entrada. As transformadas z da função de transferência do filtro que modela o canal de voz humano e da excitação são multiplicadas no domínio z. Do ponto de vista do domínio da freqüência, o canal de voz humano carrega a informação espectral sob a forma de uma envoltória.

Em um modelo LPC, o canal de voz humano é representado por um filtro de apenas pólos H(z). Como os sinais de voz são um processo não estacionário, H(z) deve ser um um filtro cujos coeficientes são variantes no tempo a cada quadro.

Como o canal de voz humano varia de forma lenta, os sinais de voz podem ser considerados como um processo estocástico cujas propriedades variam lentamente. Isto nos leva a assumir o sinal como estacionário em curtos períodos de tempo na análise LPC. Portanto, o sinal de voz pode ser considerado estacionário dentro de uma janela de N amostras, e podemos modelar a voz através de um filtro H(z), com os coeficientes de A(z) sendo obtidos através da análise de predição linear e atualizados a cada segmento do sinal.

Em um preditor linear de ordem p, a amostra atual da sequência de voz é estimada através de uma combinação linear de p amostras passadas.

Os parâmetros da predição linear são obtidos minimizando-se o erro médio quadrático da predição:

$\frac{\partial \varepsilon}{\partial a_i} = 0, i = 1,2,..., p$, onde $\varepsilon = E\left[(e(n))\right]^2 = E\left[(s(n) - \hat{s}(n))^2\right]$

A solução para esta minimização produz uma série de equações de matrizes Toeplitz, isto é para m=1,2,...p:

$$r_{ss}(m) - \sum_{i=1}^{p} a_i r_{ss}(m-i) = 0,$$

onde $r_{ss}(m) = E\left[s(n+m)s(n)\right]$ é a sequência de autocorrelação do segmento de voz.

A mesma equação sob a forma matricial é dada por :

$$\mathbf{R_x a} = \mathbf{r_x}$$

$$\mathbf{a} = \mathbf{R_x}^{-1}\mathbf{r_x}$$

$$\begin{bmatrix} r_s(0) & r_s(1) & r_s(2) & \Lambda & r_s(M-1) \\ r_s(1) & r_s(0) & r_s(1) & \Lambda & r_s(M-2) \\ r_s(2) & r_s(1) & r_s(0) & \Lambda & r_s(M-3) \\ \text{M} & \text{M} & \text{M} & & \text{M} \\ r_s(M-1) & r_s(M-2) & r_s(M-3) & \Lambda & r_s(0) \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \\ a(3) \\ \text{M} \\ a(M) \end{bmatrix} = \begin{bmatrix} r_s(1) \\ r_s(2) \\ r_s(3) \\ \text{M} \\ r_s(M) \end{bmatrix}$$

ou equivalentemente:

$$\begin{bmatrix} a(1) \\ a(2) \\ a(3) \\ \text{M} \\ a(M) \end{bmatrix} = \begin{bmatrix} r_s(0) & r_s(1) & r_s(2) & \Lambda & r_s(M-1) \\ r_s(1) & r_s(0) & r_s(1) & \Lambda & r_s(M-2) \\ r_s(2) & r_s(1) & r_s(0) & \Lambda & r_s(M-3) \\ \text{M} & \text{M} & \text{M} & & \text{M} \\ r_s(M-1) & r_s(M-2) & r_s(M-3) & \Lambda & r_s(0) \end{bmatrix}^{-1} \begin{bmatrix} r_s(1) \\ r_s(2) \\ r_s(3) \\ \text{M} \\ r_s(M) \end{bmatrix}$$

A seqüência de autocorrelação pode ser estimada através das N amostras do sinal de voz usando-se um estimador do tipo *polarizado* ou *não polarizado*. Os estimadores do tipo *polarizado* são sempre desejáveis, uma vez que geralmente produzem polinômios de fase mínima e por este motivo foram utilizados em nossa implementação do sistema LPC.

Estimador do tipo *polarizado:*
:

$$\hat{r}_{ss}(m) = \frac{1}{N-|m|} \sum_{i=0}^{N-|m|-1} s(n+|m|)s(n)$$

Estimador do tipo *não polarizado*:

$$\tilde{r}_{ss}(m) = \frac{1}{N} \sum_{i=0}^{N-|m|-1} s(n+|m|)s(n)$$

4.5 Ganho

No modelo LPC o ganho é usado para produzir sinal de voz sintético que tenha a mesma energia do sinal de voz original. O cálculo do ganho é realizado relacionando a energia na saída do filtro de análise LPC de cada segmento com a energia do segmento de sinal original. O ganho é uma função dos

coeficientes da função autocorrelação do segmento de voz analisado e dos coeficientes do filtro de análise.

$$g(n) = \left[r_x(0) - \sum_{k=1}^{p} a(k) r_x(k) \right]^{\frac{1}{2}}$$

4.6 AMDF

A função AMDF (*Average Magnitude Difference Function*) mostra a diferença média de um sinal deslocado sobre o próprio sinal e é definida para um segmento de sinal de voz estacionário:

$$AMDF(\tau) = \frac{1}{N} \sum_{j=1}^{sum\max} |s(j) - s(j+\tau)|$$

Então é formado um sinal com a diferença entre o sinal e o mesmo sinal adiantado e soma-se a diferença entre os valores das amostras. O valor da função AMDF é sempre zero para τ=0, e exibe pontos de mínimo, que correspondem ao período de pitch de um sinal de voz do tipo vozeado da vogal ‘e’, como mostrado na figura .

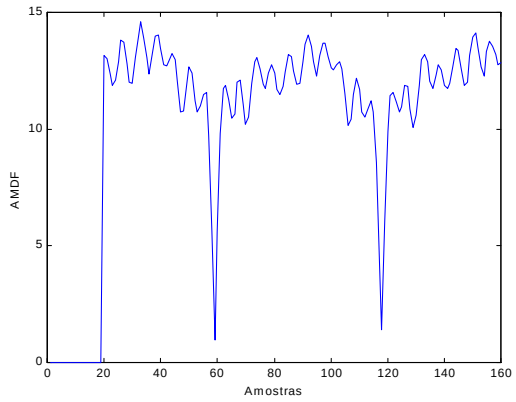

Figura 19. Função AMDF da vogal ‘e’.

4.7 Energia

A energia de um sinal de voz assumido como estacionário para cada um dos n segmentos deste sinal com N amostras pode ser expressa pela seguinte equação:

$$E(n) = \sum_{k=1}^{N} s^2(k)$$

4.8 Detecção de Pitch

O sinal diferença AMDF é formado pelo próprio sinal atrasado, e somando-se a magnitude da diferença entre os valores das amostras. O sinal diferença é

sempre zero para τ=0, e exibe pontos de mínimo nos atrasos correspondentes ao período de *pitch* dos sinais de voz do tipo vozeado.

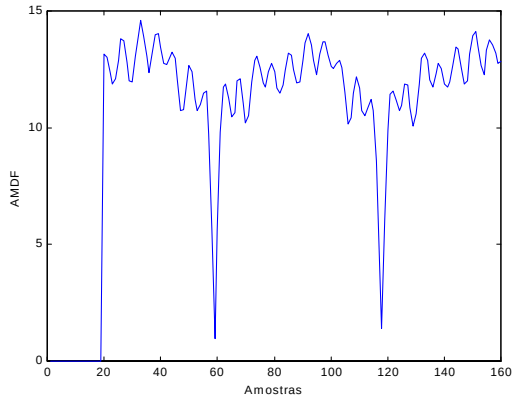

Figura 20. AMDF para segmentos de voz do tipo vozeada.

O período de pitch correto é determinado fazendo-se uma procura pelo primeiro ponto de mínimo absoluto. O ponto a que chamamos de mínimo absoluto é aquele de menor valor em todo o segmento de voz processado. Os pontos de mínimo locais derivados do segundo harmônico da freqüência fundamental de pitch são rejeitados quando a razão do valor máximo pelo valor mínimo é menor que o valor mínimo global. Os demais pontos de mínimo absoluto são rejeitados dando preferência a valores de mínimo de freqüências mais altas(ou valores menores de τ) quando os valores de mínimo estão próximos entre si (e perto do mínimo global). Se a razão do valor máximo pelo valor mínimo estiver abaixo de um limiar, então o quadro ou segmento é classificado como não vozeado. Na verdade, quando o AMDF para τ é zero, a razão do valor máximo pelo mínimo será sempre infinito. Para resolver este problema, o AMDF é calculado para os valores possíveis de τ (2.5ms - 10ms). Para detectar os quadros silenciosos, usamos o cálculo da energia. Se a energia do segmento é menor que um dado limiar, então o quadro é chamado não vozeado sem a necessidade de realizar a escolha dos picos de mínimo.

Durante a implementação, usamos tamanhos de segmento de 80 e 160 amostras, correspondendo a 10 e 20 ms, respectivamente. Na primeira parte do algoritmo, o AMDF do segmento é achado, descartando-se o fator 1/summax, uma vez que a razão entre os valores máximo e mínimo é um valor relativo. Escolhemos o valor de summax 40 e o valor máximo de τ=80 , o que corresponde a 10 ms ou o valor máximo do período do pitch esperado. Além do cálculo do AMDF, os valores máximo e mínimo do AMDF e a energia de cada segmento são também calculados no início do algoritmo e serão usados mais adiante.

A segunda parte do algoritmo corresponde a busca das estimativas do *pitch* e da decisão do tipo de sinal de voz do segmento de voz de interesse. Neste procedimento, o primeiro nulo correspondendo ao período de *pitch* é obtido e então a decisão a respeito da voz é tomada.

O processo da obtenção do período do *pitch* pode ser explicado de forma simples. Os pontos A, B, e C são candidatos a valores de mínimo correspondendo ao período de *pitch*. O ponto A correspondendo ao segundo harmônico da freqüência fundamental da freqüência de *pitch*, é rejeitado já que o valor do AMDF no ponto é maior que MIN + THR. Apesar dos pontos B e C serem mínimos, o ponto B é o escolhido como o período de *pitch*, em razão do valor do τ no ponto B ser menor que MIN+THR e o valor da amplitude no ponto B é menor que o valor no ponto C. Na fig. 21, o ponto C corresponde ao sub-harmônico da freqüência de *pitch* fundamental O limiar THR é uma função da diferença entre os valores máximos(MAX) e mínimos(MIN) do AMDF, dado por:

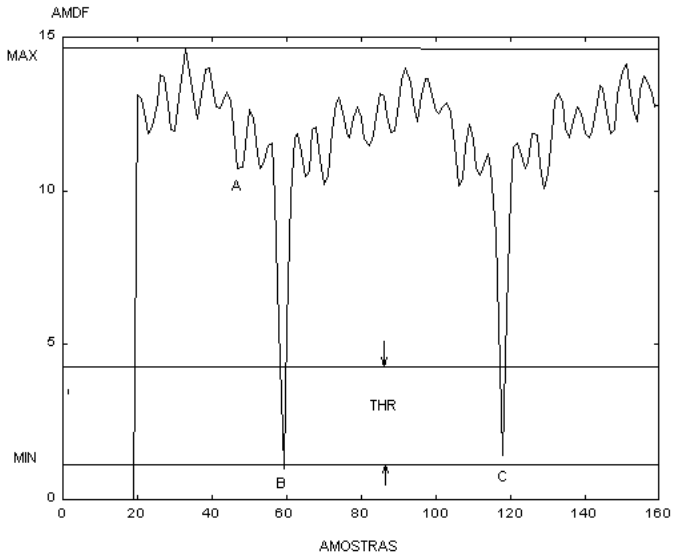

Figura. 21 Figura de ilustração da detecção do pitch

$THR = \frac{(MAX - MIN)}{NTHR}$, onde NTHR foi obtido experimentalmente como 8.

Depois de achar o período do pitch, a razão RAT é formada usando-se MAX e MINVALUE, que é o valor do AMDF no ponto do *pitch*. Podemos observar nos gráficos do AMDF que o valor do RAT é um forte indicador da característica de voz do segmento, isto é, se o segmento é vozeado ou não. Portanto, a nossa análise do tipo de voz é baseada principalmente no valor do parâmetro denominado RAT.

$$RAT = \frac{MAX}{MINVALUE}$$

Na análise de voz, a energia do segmento é comparada a um limiar de energia ENTHR, obtido experimentalmente. Em seguida, calculamos a energia do segmento e comparamos o resultado com o limiar de energia. Se o valor da energia do segmento for menor que o limiar que nos referimos, o segmento é classificado como não vozeado. Caso contrário, o RAT é comparado com um outro limiar que chamamos R1, escolhido como 2. Depois destas comparações, o segmento é classificado como não vozeado para o caso do RAT ser menor

que o limiar R1. Finalmente, para RAT maior que R1, o segmento é classificado como *vozeado*.

4.9 Gerador de Pulsos Glotais

A função do gerador de pulsos glotais é excitar o filtro de síntese quando o segmento do sinal que está sendo gerado é do tipo vozeado. Na verdade, quando excitamos um filtro de síntese, devemos colocar intervalos entre as excitações correspondentes ao período do pitch.

Entretanto, a experiência com o sistema de codificação e síntese LPC nos mostrou que as excitações feitas com um trem de impulsos produziam um som de qualidade muito ruim, devido às quedas abruptas da referida excitação. A solução para o problema do sinal de excitação foi a aplicação de sinais de excitação com características de decaimento mais suaves, como por exemplo, uma janela com transição suave ou um pulso glotal, que será definido em seguida.

Durante a implementação de um gerador de pulsos glotais, tivemos o cuidado de realizar uma análise sobre diversos sinais a fim de obter o número mínimo de amostras do período de pitch para um sinal de voz. O número obtido experimentalmente e usado como pior caso foi 20 amostras, o que nos permitia colocar um pulso glotal de até 20 amostras. Assumindo que o pulso ficava centrado na amostra equivalente ao período de pitch com 10 amostras para cada lado, evitamos uma superposição de dois pulsos glotais consecutivos. Na prática, usamos pulsos glotais com cerca de 11 amostras, que produziram a melhor qualidade de voz. Em seguida escrevemos a função que gera os pulsos glotais, assim como um gráfico descrevendo estes pulsos.

$$g(n)=\begin{cases}\dfrac{1}{2}\left[1-\cos\left(\dfrac{\pi n}{N1}\right)\right] & 0\le n\le N1\\ \cos\left(\dfrac{\pi(n-N1)}{2N2}\right) & N1\le n\le N1+N2\\ 0 & outros\end{cases}$$

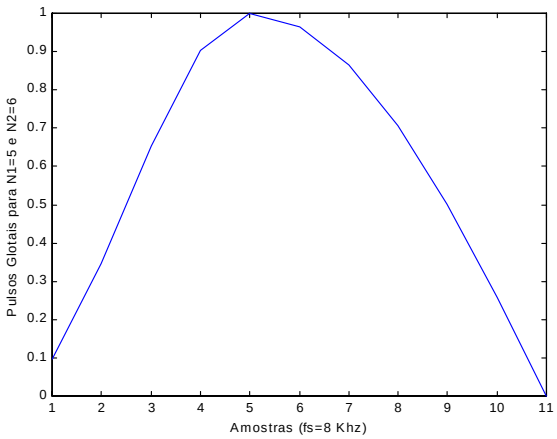

Figura 22. Um pulso glotal de 11 amostras.

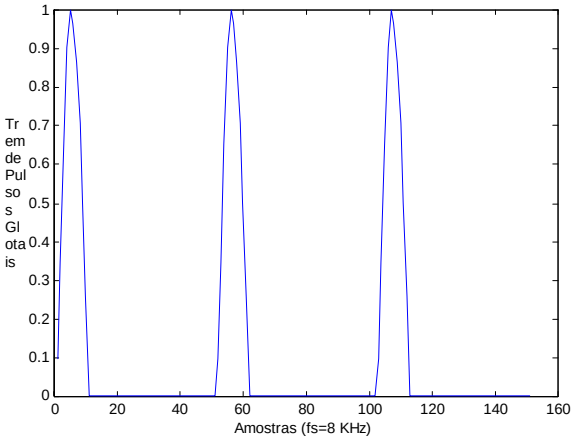

Figura 23. Trem de pulsos glotais.

4.10 Síntese LPC

O modelo linear de produção de voz LPC utiliza excitações de dois tipos: um trem de impulsos para voz do tipo vozeada e ruído aleatório para voz do tipo não vozeada. O canal de voz humano é modelado por uma função de transferência de apenas pólos. O modelo glotal é representado por um filtro passa-baixa de ordem 2 e o modelo dos lábios é representado por $L(z)=1- z^{-1}$. Finalmente, um fator de correção espectral é incluído para compensar os efeitos de baixa freqüência dos pólos. Na representação digital da voz, a correção espectral é omitida e o zero da função dos lábios é cancelado por um dos pólos glotais, reduzindo o sistema a um modelo de apenas pólos como o mostrado na figura 24.

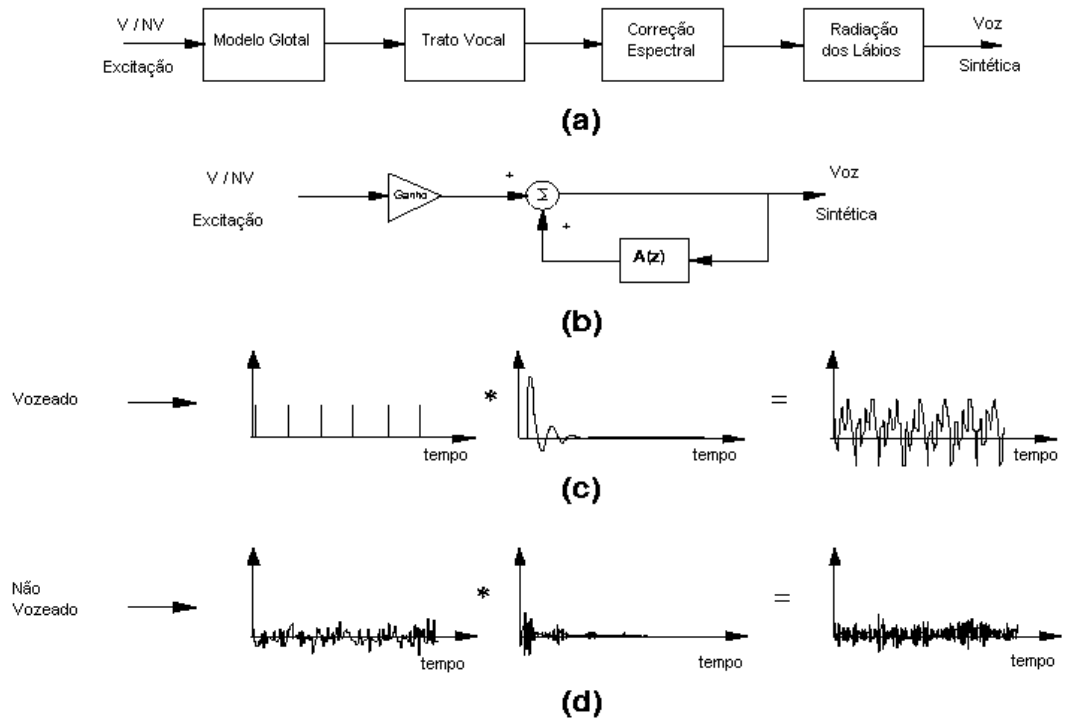

Figura 24. (a) Modelo de Produção de voz, (b) Modelo de apenas pólos do sistema, (c) Representação gráfica da produção de voz do tipo *vozeado,* (d) Representação gráfica da produção de voz do tipo não vozeado.

Na prática, nos sistema de codificação e decodificação (*codec)*, a síntese ou decodificação é baseada na obtenção de um conjunto de parâmetros, composto pelos coeficientes da análise de predição linear, o ganho e período de *pitch* de cada segmento do sinal de voz.

Estes parâmetros excitam um filtro de apenas pólos que representa um modelo matemático e acústico do canal de voz humano. Os primeiros parâmetros, os coeficientes da análise de predição linear são na verdade coeficientes de um filtro obtido no bloco que realiza a análise LPC. O segundo parâmetro, o ganho, é obtido na mesma rotina dos coeficientes LPC. E o terceiro parâmetro, o período do pitch, advém do bloco chamado detecção de *pitch* e contém na verdade duas informações importantes: o período de *pitch* em amostras e a característica do sinal de voz (vozeado/não vozeado).

O algoritmo implementado para sintetizar um sinal de voz a partir de um conjunto de parâmetros funciona da seguinte maneira. Primeiramente, a rotina de síntese recebe uma matriz contendo todos os coeficientes da análise LPC, onde cada segmento de voz corresponde a uma linha de coeficientes e o número de linhas é o número de segmentos do sinal original.

Em seguida, o algoritmo recebe um vetor contendo todos os períodos de *pitch* de todo o sinal de voz processado, onde cada segmento de voz equivale a um elemento deste vetor. Então, o primeiro passo do algoritmo é testar o vetor de *pitch*. Se o valor testado for igual a zero, significa que o segmento em questão é do tipo não vozeado, caso contrário o segmento em questão é do tipo vozeado.

A partir do conhecimento do tipo de voz de cada segmento processado, o filtro de síntese do sistema é excitado. Para um sinal do tipo não vozeado, o filtro de síntese é excitado com um sinal randômico equivalente a um ruído branco. Enquanto para um sinal do tipo vozeado, o filtro é excitado por um trem de pulsos espaçados pelo período de *pitch* de cada segmento.

Consequentemente, o sinal de saída do filtro de síntese é multiplicado pelo valor do ganho para cada segmento, contido em um vetor de ganhos. Finalmente, o algoritmo organiza os vários segmentos de voz processados na forma de um único segmento do sinal de voz reconstituído.

4.11 Filtro de De-Ênfase

O filtro de de-ênfase é usado para desfazer o efeito da pré-ênfase aplicada no início do processamento de voz do sinal. Teoricamente, a remoção do espectro natural da voz pelo filtro de pré-ênfase para voz do tipo vozeada é recolocada pelo filtro de de-ênfase, com o parâmetro μ=0.75, obtido experimentalmente. Na prática, a aplicação do filtro de de-ênfase a um sinal de voz reconstituído causa uma amplificação do sinal em questão.

$$Hde(z) = \frac{1}{1 - \mu z^{-1}}$$

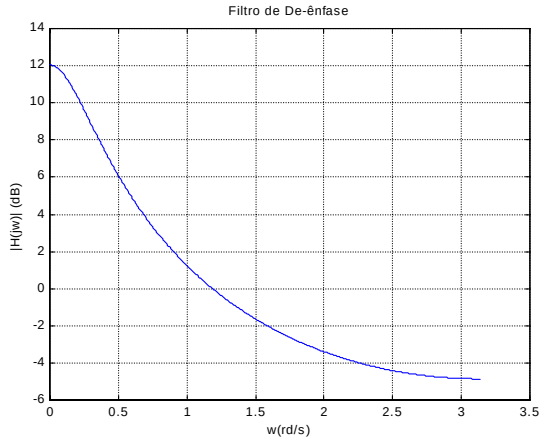

Figura 25. Filtro de de-ênfase.

4.12 Resultados

O sistema desenvolvido mostrou-se bastante flexível, em razão da divisão em diversos blocos inter-operáveis e robustos. A qualidade da voz reconstituída pelo sistema tem características sintéticas, no entanto possui um bom nível de inteligibilidade.

Durante a implementação destas rotinas em Matlab, procuramos escrever algoritmos robustos e flexíveis de modo que estas rotinas pudessem ser utilizadas por outros alunos, sem problemas de compatibilidade. Além disso, todas as rotinas ou funções desenvolvidas no Matlab contêm ajuda e uma explicação sucinta do que as mesmas realizam.

Com relação à complexidade do algoritmo LPC, podemos afirmar que o algoritmo apresenta uma complexidade média necessitando de alguns poucos minutos para processar um sinal de voz de cerca de 10 segundos. Este tempo de processamento é verificado em função da plataforma em que foi implementado o sistema, Matlab. Como o Matlab é uma linguagem interpretada, as suas implementações são muito mais complexas que em algoritmos escritos em C, por exemplo.

5. Implementação de um Codificador CELP

5.1 Introdução

O sistema CELP pertence a uma classe denominada sistemas de análise por síntese. Nestes sistemas os parâmetros do sistema como o ganho, os coeficientes do filtro de síntese, o pitch e a informação do tipo de voz (se vozeado ou não vozeado), são determinados pelo sistema de predição linear (LPC). Entretanto, a seqüência de excitação do filtro de síntese é mantida em um dicionário de códigos (*codebook*) e determinada por um processo de otimização em malha fechada, isto é, por um procedimento de análise por síntese, com mostrado na figura 26.

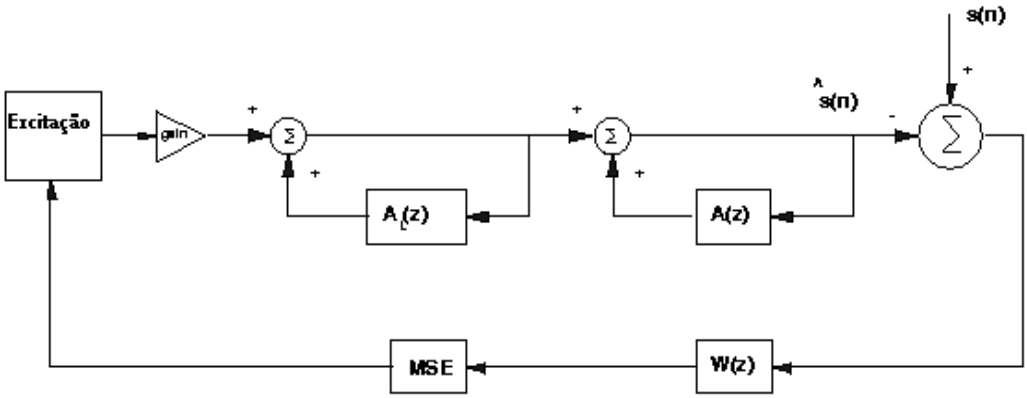

Figura 26. Um codificador de predição linear de análise por síntese.

A seqüência de excitação contida no dicionário de códigos é determinada por um processo de otimização que minimiza a diferença entre o sinal de voz original e o sinal sintetizado. Na realidade, chamamos esta diferença entre o sinal de voz original e o sinal sintetizado de erro. Em seguida, aplicamos um filtro perceptual de pesagem do erro, cuja função é otimizar o codificador para o bom entendimento da voz produzida.

De acordo com a figura 26, podemos descrever os principais blocos de um sistema de análise por síntese. Este sistema consiste em um filtro de predição linear de curto período, que representa a estrutura dos formantes da voz; um filtro de predição linear de período longo, que representa o *pitch* da voz; um filtro perceptual de pesagem W(z), que modela o erro de modo que o ruído de quantização seja mascarado pelos formantes de alta energia; um bloco chamado de erro médio quadrático pesado (MSE-*Mean Squared Error*), que minimiza o erro causado pela excitação; e um gerador de excitações, contidas no dicionário de código.

De um modo geral, os sistemas de análise por síntese são codificadores essencialmente híbridos, uma vez qe combinam as vantagens dos modelos baseados em *vocoders*, e as propriedades dos codificadores de formato de

onda. Além disso, as propriedades do sistema auditivo humano são exploradas através do uso da filtragem e pesagem perceptual.

5.2 Dicionário de Códigos (*Codebook*)

5.2.1 Introdução

O dicionário de códigos utilizado no sistema CELP contém um conjunto de excitações em formato de onda, que excitam o filtro de síntese do sistema. Estas excitações são sequências gaussianas de comprimento ajustável, obtidas através de um processo conhecido como quantização vetorial ou então através da geração de seqüências aleatórias com uma distribuição gaussiana. A excitação considerada ótima pelo sistema é escolhida de modo que o MSE pesado perceptualmente seja minimizado. Assim, o número de seqüências ou excitações usado no dicionário de códigos influencia de forma direta a complexidade computacional do sistema, uma vez que a seqüência ótima será selecionada após a aplicação de todas as entradas do *codebook*.

5.2.2 A Montagem do Dicionário de Códigos

A montagem de um dicionário de códigos usando quantização vetorial é um processo bastante complexo em termos computacionais. Primeiramente, é necessário um número de seqüências de voz para o treinamento do sistema. Estas seqüências de treinamento consistem de um conjunto de frases foneticamente balanceadas, isto é, são frases que possuem uma grande riqueza no que concerne o número de fones encontrados na língua portuguesa. Estas frases foneticamente balanceadas apresentam uma freqüência de ocorrência de fones que se aproxima de modo significativo daquela encontrada na língua falada. O número de frases balanceadas não deve ser inferior a 10 (dez), a fim de que o dicionário de códigos tenha seqüências bastante representativas no que se refere aos fones encontrados na língua portuguesa.

Para a construção das frases foneticamente balanceadas, consideramos as freqüências relativas esperadas F_i dos 37 fones do português falado e as freqüências relativas observadas f_i em uma lista de dez frases, onde N é o número de fones da lista e n_i é o número de vezes que o i-ésimo fone ocorreu na lista. Para que a lista de frases seja considerada foneticamente balanceada, é necessário que a freqüência relativa f_i se aproxime suficientemente bem de F_i. Através de um teste chamado χ^2, podemos determinar se a freqüência relativa desvia ou não de forma significativa de F_i. Então, usamos a distribuição com 36 graus de liberdade.

$$\chi^2 = \sum_{i=1}^{37} \frac{\left(n_i - NF_i\right)^2}{NF_i}$$

Em seguida, a obtenção de cada lista de frases foneticamente balanceada é feita da seguinte maneira. Escolhemos as frases, calculamos o número total N de fones da lista, calculamos o número de vezes que cada fone ocorreu no

total de dez frases, a freqüência relativa de cada fone no total das dez frases e finalmente calculamos χ^2 com a fórmula anterior.

Assim, calculamos a combinação das dez frases que resulta no menor χ^2. Caso este valor seja menor que um determinado limiar, escolhemos este conjunto como uma seqüência de treinamento. A seguir descreveremos os conjuntos de frases balanceadas que foram usados para o treinamento do sistema.

Lista 1:

A questão foi retomada no congresso.
Leila tem um lindo jardim.
O analfabetismo é a vergonha do país.
A casa foi vendida sem pressa.
Trabalhando com união rende muito mais.
Recebi nosso amigo para almoçar.
A justiça é a única vencedora.
Isso se resolverá de maneira tranqüila.
Os pesquisadores acreditam nessa teoria.
Sei que atingiremos o objetivo.

$\chi^2 = 11,85$
Nº total de fones = 259

Lista 2:

Nosso telefone quebrou.
Desculpe se magoei o velho.
Queremos discutir o orçamento.
Ela tem muita fome.
Uma índia andava na mata.
Zé, vá mais rápido!
Hoje, dormirei bem.
João deu pouco dinheiro.
Ainda são seis horas.
Ela saía discretamente.

$\chi^2 = 10,43$
Nº total de fones = 176

Lista 3:

Eu vi logo a Iôiô e o Léo.
Um homem não caminha sem um fim.
Vi Zé fazer essas viagens seis vezes.
O atabaque do Tito é coberto com pele de gato.
Ele lê no leito de palha.
Paira um ar de arara rara no Rio Real.
Foi muito difícil entender a canção.
Depois do almoço te encontro.

Esses são nossos times.
Procurei Maria na copa.

$\chi^2 = 12,12$
Nº total de fones = 229

Lista 4:

A sensibilidade indicará a escolha.
A Amazônia é a reserva ecológica do globo.
O ministério mudou demais com a eleição.
Novos rumos se abrem para a informática.
O capital de uma empresa depende da produção.
Se não fosse ela, tudo teria sido contido.
A principal personagem no filme é uma gueixa.
Receba seu jornal em sua casa.
A juventude tinha que revolucionar a escola.
A atriz terá quatro meses para ensaiar seu canto.

$\chi^2 = 13,46$
Nº total de fones = 315

5.2.3 Treinamento do Dicionário de Códigos

Para realizar o treinamento do dicionário de códigos todas as referidas seqüências de treinamento são aplicadas a um bloco de filtragem inversa. Este bloco consiste na utilização do filtro de síntese do sistema de maneira inversa, com o objetivo de se produzir as excitações consideradas "ideais" para excitar o filtro de síntese do sistema CELP. Após a aplicação do filtro inverso, obtivemos um conjunto de sinais a que chamamos de excitações ideais, como mostra a Fig. 27.

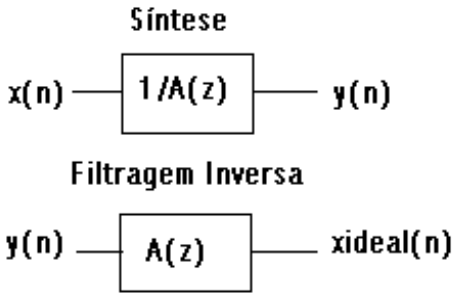

Fig. 27 O processo de filtragem inversa.

Após a obtenção do conjunto de excitações ideais, é iniciado o processo da montagem do *codebook* propriamente dito. Para a construção do dicionário de códigos, implementamos uma rotina em Matlab que calcula o centróide de um dado conjunto de n segmentos. Na verdade, o cálculo do centróide é feito para o conjunto de n segmentos de t amostras do conjunto de excitações consideradas ideais, como mostrado na figura 28.

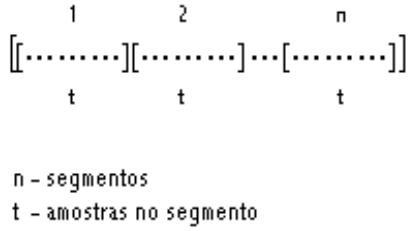

Fig 28 Segmentos contendo as excitações ideais.

O procedimento de obtenção do centróide é realizado da seguinte maneira. Primeiro, calculamos as distâncias euclideanas de cada segmento para com os demais segmentos de todo o conjunto de excitações ideais. Em seguida, alocamos os valores escalares destas distâncias em uma matriz quadrada n x n como mostrado a seguir:

$$\begin{bmatrix} 0 & d_{12} & d_{13} & \Lambda & d_{1n} \\ d_{12} & 0 & d_{23} & \Lambda & d_{2n} \\ d_{13} & d_{23} & 0 & \Lambda & d_{3n} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ d_{1n} & d_{2n} & d_{3n} & \Lambda & 0 \end{bmatrix}$$

Então, realizamos a soma dos valores das distâncias euclideanas de cada coluna ou linha. Como a matriz é simétrica, é indiferente realizar a soma das colunas ou das linhas. O primeiro centróide será aquele de menor valor na soma das colunas ou linhas e o índice correspondente à coluna ou linha de menor valor na soma será o índice do próprio centróide neste procedimento.

5.2.4 O Algoritmo LBG Modificado

O algoritmo LBG modificado consiste na repetição sucessiva do cálculo dos centróides, sendo que a cada centróide calculado o conjunto de vetores fornecido é dividido em duas regiões. Desta forma, teríamos no fim da aplicação deste algoritmo um número desejado de centróides obtidos a partir de conjuntos de vetores previamente separados em regiões.

Além disso, é necessário o fornecimento de um parâmetro ε que corresponde a um desvio ou erro do centróide para com os demais segmentos de voz. Este parâmetro é o responsável pela divisão dos segmento de voz em dois grupos distintos, o primeiro grupo compreendendo os segmentos situados entre o centróide e o centróide acrescido do parâmetro ε e o segundo grupo compreendendo a região restante. Como resultado deste processo conhecido como algoritmo de quantização LBG modificado, são obtidos o centróide e as duas regiões descritas acima.

Por outro lado, podemos interpretar o processo de construção de um dicionário de códigos como a divisão de uma dada região no espaço em pequenas

células com os seus respectivos centróides, como mostrado na figura 29. Na verdade, o processo de construção que foi descrito é chamado de quantização vetorial. Este processo parte de um conjunto de dimensão infinita de segmentos de voz, todos os sons da língua portuguesa, e tenta aproximar este conjunto para um conjunto finito de segmentos de voz, o dicionário de códigos, que possa representar de forma eficaz um universo enorme de possibilidades. Em outras palavras, realizando um treinamento computacional altamente complexo, procuramos fornecer um número de seqüências de sinais de voz que possam representar bem um conjunto infinito de ocorrências de sinais de voz encontrados na língua portuguesa. Assim, foram realizadas gravações de diversos conjunto de frases foneticamente balanceadas, e estas gravações foram fornecidas como treinamento para a rotina de montagem do dicionário de códigos.

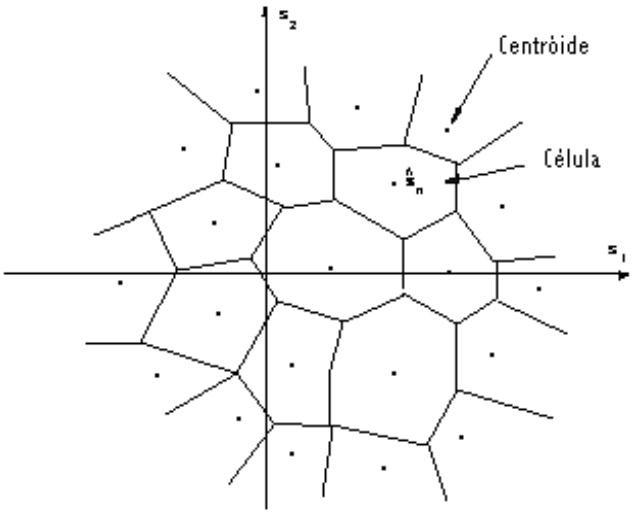

Fig 29 Esquema de células no processo de quantização vetorial.

Para a construção de um dicionário de códigos eficaz, implementamos uma rotina em Matlab que realiza o cálculo do centróides repetidamente, de maneira que possamos obter um dicionário com um número de entradas compatível com as necessidades do sistema de excitação por dicionário de códigos. O maior problema no treinamento do *codebook* é justamente o valor do parâmetro ε. O fornecimento deste parâmetro e do conjunto de treinamento deve resultar no cálculo de um centróide e na divisão de um conjunto de seqüências de treinamento em dois conjuntos, se possível do mesmo tamanho. O valor do parâmetro $\varepsilon = 0{,}01$ foi obtido experimentalmente. Ao longo dos muitos testes realizados para a obtenção de ε verificamos uma ocorrência enorme de divisões de um dado conjunto de treinamento em conjuntos de tamanhos bastante diferentes, como por exemplo, de um conjunto contendo apenas cerca de dez por cento (10%) das seqüências fornecidas no treinamento enquanto o outro conjunto apresentava cerca de noventa por cento (90%) das referidas seqüências, para a primeira iteração.

À medida que o algoritmo LBG modificado divide o conjunto de treinamento em regiões de tamanhos diferentes, as regiões divididas vão se tornando menores. De acordo com o tamanho do conjunto de treinamento inicial, o algoritmo pode chegar a divisões em regiões contendo matrizes vazias, interrompendo o processo. Então, é imperativo que sejam usados conjuntos de seqüências de treinamento grandes o suficiente, de maneira que após a divisão em regiões os conjuntos resultantes apresentem tamanhos adequados para o cálculos dos próximos centróides.

5.2.5 O Novo Algoritmo

O problema do algoritmo LBG descrito anteriormente está na divisão das regiões em tamanhos desiguais, a cada iteração do algoritmo. Para o cálculo dos centróide de uma maneira eficaz, seria necessário dividir as duas regiões resultantes de cada iteração em duas regiões iguais. Então, decidimos utilizar um novo algoritmo que consiste em dividir um dado conjunto de segmentos em regiões iguais a cada iteração do algoritmo.

O novo algoritmo funciona da seguinte maneira. Primeiro, o cálculo do centróide é realizado da mesma maneira que no algoritmo LBG modificado. No entanto, a divisão do conjunto de segmentos de voz de treinamento não utiliza o parâmetro ε como auxiliar na divisão em duas regiões. Essa divisão é feita calculando-se os dois vetores que estão mais distantes dos demais, isto é, os dois vetores que se situam nos extremos do espaço que compreende os vetores de treinamento. Como construímos uma matriz contendo as distâncias euclidianas de cada vetor para os demais, criamos um vetor contendo as distâncias em uma linha ou coluna da matriz de distâncias de maneira crescente e guardamos os índices contendo as posições dos vetores na matriz de distâncias para que possamos escolher os vetores que serão agrupados em cada uma das duas regiões resultantes. Então, escolhemos no vetor de distâncias em ordem crescente a primeira metade das distâncias que corresponde aos vetores que serão agrupados em uma das regiões. Essa primeira metade do vetor de distâncias em ordem crescente representa a metade dos vetores que estão mais próximos de um dos vetores de posição mais distante. A metade restante é agrupada na outra região. Assim, conseguimos obter um centróide e a divisão em duas regiões de mesmo tamanho, a partir de um dado conjunto de treinamento

Dessa maneira, aplicamos um método eficaz para quantizar vetorialmente um conjunto de seqüências de treinamento, resultando no cálculo do número desejado de centróides e mais importante, na divisão em regiões de tamanhos iguais para o cálculo desses centróides. O processo descrito anteriormente é repetido sucessivamente até que obtenhamos o número de regiões desejado e o centróide é calculado para cada uma dessas regiões. A diferença entre as duas técnicas utilizadas é que o LBG modificado resultava na divisão em regiões de tamanhos diferentes, e como conseqüência a escolha de centróides para regiões tão diferentes não representava adequadamente o conjunto inicial de treinamento. Por outro lado, o novo algoritmo se mostrou mais eficaz uma vez que dividia o chamado espaço de vetores de treinamento sempre em regiões de mesmo tamanho.

5.2.6 O Funcionamento de um Dicionário de Códigos

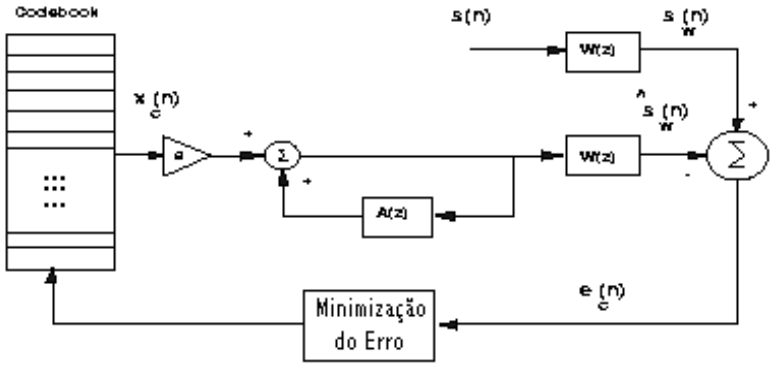

Figura 30. Diagrama em blocos de um codificador CELP.

O funcionamento de um dicionário de códigos é bastante simples, uma vez que este componente do sistema consiste em uma matriz contendo as seqüências obtidas no processo de construção do *codebook*.

O tamanho do dicionário de códigos influencia diretamente a complexidade do sistema e da mesma forma influencia o tempo de processamento de um dado sinal de voz. O número desejável de entradas no *codebook* se situa em torno de 1024 entradas, contudo dicionários de códigos com um número de entradas de 512, 256, 128, 64 e 32 também foram testados, apresentando resultados inferiores ao do *codebook* com 1024 entradas.

5.3 O Módulo de Procura e Otimização do Codificador CELP

Para a realização da procura pelo sinal ótimo de excitação para um dado segmento de voz processado, foi implementada a rotina de procura e otimização. Esta rotina tem por objetivo testar todas as seqüências de excitação contidas no dicionário de códigos e escolher a seqüência de excitação que produz o menor erro médio quadrático pesado perceptualmente .

Em linhas gerais, a rotina de procura e otimização recebe os parâmetros básicos necessários ao processamento do sinal de voz e realiza uma procura exaustiva no dicionário de códigos e em seguida realiza a otimização de cada segmento processado utilizando como ferramentas o filtro de pesagem perceptual e o cálculo do erro médio quadrático. Entre os chamados parâmetros básicos para a realização de um processamento de um sinal de voz usando o sistema de codificação e síntese CELP estão: os coeficientes da análise LPC, o número dos segmentos de voz a serem processados, o tamanho do segmento de voz a ser processado e um dicionário de códigos contendo as seqüência de excitações.

O funcionamento da função de procura e otimização é descrito pelas seguintes etapas. A princípio, a rotina aloca a memória necessária para o processamento

dos diversos vetores e matrizes a serem utilizados no processamento do sistema CELP. É necessário realizar este tipo de otimização na rotina em face da grande complexidade computacional de um sistema de processamento como este, onde há a necessidade de se empreender buscas exaustivas do codebook para muitos segmentos de voz, durante o processo. Com a implementação deste tipo de alocação de memória nas rotinas, pode-se reduzir de maneira substancial o tempo de processamento. Em seguida, realizamos para cada segmento a síntese usando o filtro FIR de apenas pólos com os coeficientes de predição linear obtidos na análise de predição linear.

Para cada segmento processado, a síntese é realizada para todas as seqüências de excitação do dicionário de códigos. Por esta razão é que se costuma dizer que o sistema CELP realiza uma procura exaustiva de modo a minimizar o erro médio quadrático, ou ainda, de modo a escolher a melhor excitação possível dado um conjunto finito de excitações.

Então, para todas as excitações do dicionário de códigos de cada segmento aplicamos o filtro perceptual de pesagem do erro. Após a pesagem perceptual, calculamos também para todas as excitações de cada segmento, o erro médio quadrático de cada síntese e construímos um vetor contendo os valores do erro médio quadrático para todas as excitações do dicionário de códigos, para cada segmento processado.

Assim, para cada segmento processado procuramos, no vetor contendo os valores do erro médio quadrático, o menor valor do erro e o índice correspondente a este valor minimizado. Na verdade, o índice deste valor minimizado também corresponde ao índice do dicionário de códigos para o qual temos a seqüência de excitação ótima. Desta forma, excitamos o filtro de síntese FIR utilizando os coeficientes de predição linear com a seqüência de excitação do dicionário de códigos correspondente ao índice do menor valor do erro médio quadrático. Finalmente, todo este processo descrito anteriormente é repetido para os muitos segmentos de voz do sinal que escolhemos processar e o sinal de voz sintetizado é reconstituído a partir dos segmentos individuais.

5.4 Cálculo do Erro Médio Quadrático

O cálculo do Erro Médio Quadrático (MSE) é realizado a fim de fornecer uma medida objetiva da qualidade do sinal sintetizado pelo filtro FIR com os coeficientes da análise LPC.

O MSE é obtido a partir do sinal diferença entre um segmento do sinal de voz original e um segmento do sinal sintetizado. Em seguida, este sinal diferença é elevado ao quadrado e é calculada a média das amostras contidas em cada segmento de L amostras, como mostrado na equação abaixo.

$$MSE = \sum_{n=0}^{N-1} d^2(n) = \sum_{n=0}^{N-1} (s_{pesado\_original} - s_{pesado\_sintetizado})^2$$

Os resultados calculados para cada segmento de voz são guardados em um vetor, que após o fim do processamento para um dado segmento, deverá conter os valores dos erros médios quadráticos calculados para cada seqüência de excitação do dicionário de códigos. O vetor contendo os resultados do cálculo do erro médio quadrático é fornecido a rotina de procura e otimização, que faz uma varredura neste vetor a fim de escolher o menor valor do erro médio quadrático, ou seja, o erro minimizado.

5.5 Filtro de Pesagem Perceptual

No sistema CELP, existe a necessidade de aplicarmos um filtro linear de pesagem perceptual no sinal de voz original e no sinal de voz sintetizado, a fim de atenuar as freqüências onde o sinal diferença é perceptualmente menos importante e amplificar as freqüências onde o sinal diferença é perceptualmente mais importante. A função de transferência deste filtro de pesagem perceptual é dada na equação abaixo.

$$w(z) = \frac{1 - \sum_{k=1}^{p} a_k z^{-k}}{1 - \sum_{k=1}^{p} a_k \alpha^k z^{-k}}$$

Os parâmetros a_k são os coeficientes da análise de predição linear e α é um parâmetro para controle da pesagem do erro como uma função da freqüência. A constante de pesagem α é dada pela fórmula abaixo, onde f_s é a freqüência de amostragem. Na prática, usamos α=0.9.

$$\alpha = e^{\frac{-2\pi.100}{f_s}}$$

Na rotina de implementação da filtragem descrita acima, os sinais original e sintetizado são aplicados ao filtro de pesagem perceptual antes da obtenção do sinal diferença entre estes dois sinais. Este procedimento é realizado de maneira a evitar instabilidade numérica no restante do processamento dos sinais de voz, fato que ocorre quando calculamos o sinal diferença antes da aplicação do filtro.

Podemos verificar na figura 31 que o filtro de pesagem perceptual realiza um tipo de mascaramento do sinal de erro, chamado sinal mascarado, com o sinal de voz original, chamado de sinal “mascarante”. O volume percebido do sinal de erro é determinado tanto pela potência total do sinal de erro ou sinal diferença e pela distribuição espectral deste sinal no que diz respeito ao sinal de voz original. Quando o espectro é plano, o ruído percebido se localiza nas regiões do espectro onde o sinal de voz têm baixa energia. Modelando o espectro do ruído de maneira que este seja proporcional ao espectro do sinal reduz o ruído percebido, e desta forma, melhora de uma maneira geral a qualidade da voz.

Fig. 31 O espectro da vogal “a” e a resposta em freqüência do filtro de pesagem de erro correspondente com α=0.8.

Uma importante propriedade do filtro de pesagem perceptual, W(z), é que os seus zeros cancelam os pólos do filtro de síntese LPC. Obtendo a resposta ao impulso do filtros em cascata, temos:

$$H_w(z) = H(z)W(z) = \frac{1}{1 - \sum_{k=1}^{p} a_k \alpha^k z^{-k}}$$

Podemos chamar a combinação dos filtros acima de filtro de síntese LPC com pesagem perceptual. Esta expressão pode ser aproximada por um filtro FIR de comprimento relativamente curto:

$$h_w(n) \approx 0$$

A expressão acima é chamada de resposta ao impulso pesada perceptualmente e o seu comprimento é tipicamente 20. As aproximações que foram realizadas no que concerne o filtro de pesagem perceptual e o filtro de síntese LPC resultam em uma economia computacional substancial no processo de análise por síntese.

Após a aplicação do filtro perceptual aos sinais original e sintetizado, obtemos o sinal diferença entre os dois. Então, construimos uma matriz com o número de linhas igual ao número de segmentos a serem processados do sinal de voz e o número de colunas igual ao tamanho do segmento do sinal de voz. Esta matriz nseg x L é a entrada para a rotina de cálculo do erro médio quadrático, descrita anteriormente.

5.6 Filtro de Pré-Ênfase

A aplicação de um filtro de pré-ênfase ao sinal de voz original é necessária, uma vez que o filtro concentra a energia relativa ao espectro de alta freqüência do sinal, retirando a parte DC do sinal, introduzindo um zero perto de ω=0.

Além disso, há outras razões para se empregar o filtro de pré-ênfase. A primeira razão é a prevenção contra instabilidade numérica, sendo que os trabalhos nesta área focaram-se no método da autocorrelação. Assumindo que o sinal de voz é dominado por componentes em freqüências baixas, é bastante previsível que um modelo LPC de ordem elevada poderá resultar em uma matriz de autocorrelação contaminada. Chamamos de um sinal contaminado a um sinal que contém componentes indesejados ou que é dominado por ruído. Um filtro de primeira ordem deve ser capaz de aproximar o espectro do sinal ao espectro do ruído branco. Outra razão é que o componente de fase mínima do sinal glotal pode ser modelado por um filtro simples de 2 pólos próximos a z=1. Então a característica dos lábios com o seu zero perto e z=1, tende a cancelar os efeitos espectrais de um dos pólos glotais. Introduzindo, um segundo zero próximo a z=1, as contribuições espectrais da laringe e dos lábios seriam eliminadas, fazendo com que a análise LPC correspondesse apenas ao canal de voz humano. No entanto, apesar do filtro de pré-ênfase, o espectro de predição linear não fica totalmente livre dos efeitos da laringe e dos lábios. Em geral, a pré-ênfase propicia aos primeiros formantes uma maior chance de influenciar a voz.

Com relação ao filtro de pré-ênfase, usamos um filtro de ordem um(1) com o parâmetro μ variando de 0.9 a 1.0 ($0.9 \leq \mu < 1.0$), com a seguinte característica :

$$Hpe(z) = 1 - \mu z^{-1}$$

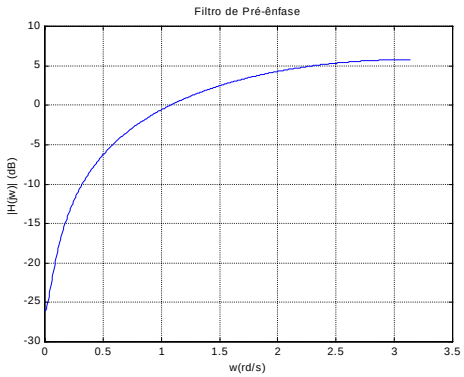

Figura 32. Filtro de Pré-ênfase com μ=0.95.

5.7 Segmentação e Janelamento

Para aplicações de processamento de sinais de voz é necessário selecionar uma porção do sinal que possa ser considerada estacionária para analisá-la. Os sinais de voz podem se considerados como estacionários quando analisados em pequenos segmentos da ordem de 20 ms. Com isto, reduzem-se os efeitos do sinal de excitação na estimativa dos coeficientes do filtro de síntese, e obtêm-se uma melhor estimativa do espectro da voz.

O quadro ou segmento de análise, I, corresponde ao número de amostras que será usado para determinar os coeficientes da análise LPC. A razão I/L representa a taxa de superposição entre dois segmentos de análise adjacentes. Em nosso sistema testamos uma taxa de 50% (I=L/2), com I=120 amostras e L=240 amostras, como mostrado na figura 33.

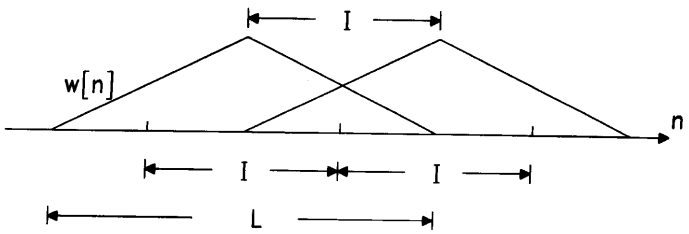

Figura 33. Superposição de dois segmentos de voz.

Contudo, não verificamos no sistema de codificação e síntese CELP o mesmo comportamento verificado no sistema LPC. Enquanto no sistema LPC a superposição descrita anteriormente tenha contribuído para uma melhora significativa da qualidade da voz processada, no sistema CELP a superposição não contribuiu com uma melhor qualidade de voz, mas somente com um aumento do tempo de processamento. Desta maneira, decidimos não usar superposição no sistema CELP de modo a economizar tempo de processamento em um sistema que realiza procuras exaustivas no dicionário de códigos. Os tamanhos de segmentos escolhidos para os testes deste sistema foram de 160 e 80 amostras por segmento, que correspondem a 20 e 10 ms para uma taxa de amostragem de 8000 Hz.

Outros conjuntos de parâmetros muito importantes para a análise LPC são aqueles relacionados a aplicação das janelas. Estes parâmetros incluem o tipo e o tamanho do quadro de análise e da janela. São características espectrais desejáveis das janelas: uma largura de banda estreita no lóbulo principal e grande atenuação nos lóbulos laterais. Em geral, uma largura de banda estreita deve resolver os pequenos detalhes do sinal janelado, enquanto a grande atenuação nos lóbulos laterais deve evitar que o espectro do sinal seja corrompido pelo ruído de "*aliasing*".

No entanto existe um compromisso na escolha do tipo de janela A janela retangular preserva as características temporais do sinal, mas acarreta um truncamento de maneira abrupta do sinal nas extremidades. As janelas de Hamming, Hanning, Blackman e Kaiser apresentam características de truncamento mais suaves, com uma maior distorção do sinal no domínio do tempo, mas apresentando um truncamento menos abrupto nas extremidades.

Usando o método da autocorrelação, a janela de nossa escolha, Hamming, é continuamente aplicada ao sinal de voz. Em geral, usamos janelas de Hamming ou de Hanning que possuem um caimento suave a fim de reduzir os efeitos de truncamento nas extremidades do segmento. As janelas com caimento mais suave costumam produzir melhores resultados do que janelas

triangulares ou retangulares. A equação característica da janela de Hamming de ordem N é:

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), & n = 0,1,...,N-1 \\ 0, & caso\ contrário \end{cases}$$

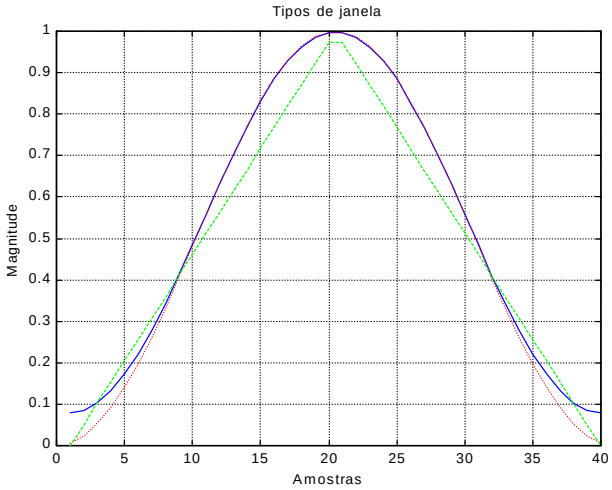

Figura 34. Tipos de janela: Hamming (azul), Hanning (vermelho) e triangular (verde).

5.8 Ganho

No modelo do codificador CELP, da mesma forma que no modelo LPC, o ganho é usado para reconstituir o sinal de voz processado com a mesma energia do sinal de voz original. O cálculo do ganho é realizado relacionando a energia na saída do filtro de análise LPC de cada segmento com a energia do segmento correspondente do sinal original. O ganho é uma função dos coeficientes da função autocorrelação do segmento de voz analisado e dos coeficientes do filtro de análise, como mostrado na equação abaixo.

$$g(n) = \left[r_x(0) - \sum_{k=1}^{p} a(k) r_x(k) \right]^{\frac{1}{2}}$$

5.9 Síntese CELP

A síntese dos sinais de voz no sistema CELP consiste em um filtro de síntese de apenas pólos caracterizado pelos coeficientes da análise de predição linear realizada a cada segmento de voz. A grande diferença do sistema CELP para o sistema LPC é no que concerne o tipo de excitação. No CELP as seqüências de excitação, em forma de onda, são mantidas no dicionário de códigos descrito anteriormente e após serem todas testadas a fim de se obter a minimização do erro médio quadrático, escolhemos então a excitação que gera o menor erro.

Assim, o sistema implementado escolhe as melhores excitações para os seus respectivos segmentos e ordena os segmentos sintetizados em um único sinal de voz sintetizado através da técnica de análise por síntese.

5.10 Filtro de De-Ênfase

O filtro de de-ênfase do sistema CELP, da mesma forma que o usado nos sistema LPC, é empregado para desfazer o efeito da pré-ênfase aplicada no início do processamento de voz do sinal. Teoricamente, a remoção do espectro natural da voz pelo filtro de pré-ênfase para voz do tipo vozeada é recolocada pelo filtro de de-ênfase, com o parâmetro μ=0.75, obtido experimentalmente. Na prática, a aplicação do filtro de de-ênfase a um sinal de voz reconstituído causa uma amplificação do sinal em questão. A função de transferência típica de um filtro de de-ênfase é mostrada na equação abaixo, e na figura 35 podemos observar a resposta em freqüência típica do filtro para μ=0.75.

$$Hde(z) = \frac{1}{1 - \mu z^{-1}}$$

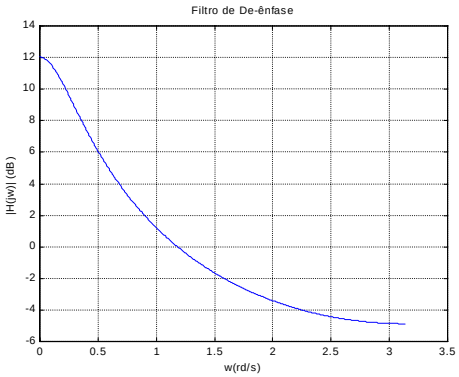

Figura 35. Filtro de de-ênfase com μ=0.75.

5.11 Resultados

Durante a fase de implementação do sistema de codificação e síntese CELP, verificamos as grandes possibilidades de se transmitir voz de alta qualidade a uma taxa de bits baixa. Contudo, este sistema também apresenta exigências enormes com relação a capacidade de processamento, uma vez que as buscas exaustivas no dicionário de códigos empreendidas pelo módulo de procura e otimização duravam cerca de alguns minutos de processamento para um sinal falado de apenas alguns segundos.

Após o término da implementação dos blocos do sistema, os primeiros testes foram realizados da seguinte forma. A partir da gravação de um determinado sinal falado, aplicamos o processo de filtragem inversa a esse conjunto de sinais a fim de gerar um conjunto de excitações consideradas ideais. Esse conjunto de excitações ideais foi colocado no sistema no lugar que seria do dicionário de códigos, para que pudéssemos testar o funcionamento e o desempenho do sistema implementado.

Então, realizamos o processamento do mesmo sinal que gerou as excitações a partir do processo de filtragem inversa. Desta maneira, reunimos as condições para que o sinal falado processado pelo sistema tivesse uma excelente qualidade, uma vez que tínhamos no conjunto de excitações todas as seqüências existentes no sinal processado e caso o sistema funcionasse como planejado teríamos o sinal reconstituído perfeitamente. O resultado foi que a voz processada pelo sistema CELP era de excelente qualidade, confirmando as nossas expectativas.

Em seguida, fizemos o mesmo processo de filtragem inversa para a lista número 1 das frases foneticamente balanceadas. Escolhemos este conjunto de frases em função do seu tamanho de cerca de 800 amostras, adequado para colocá-lo no lugar do codebook. Assim, gravamos outros sinais falados e os testamos no sistema com o codebook descrito logo acima. O resultado foi que não obtivemos sinais de fala de tão boa qualidade.

Constatamos que o dicionário de códigos é sem dúvidas, entre os componentes do sistema, o grande diferencial no que concerne a geração de sinais processados com alta ou baixa qualidade de voz. Entretanto, é muito difícil gerar um dicionário de códigos da maneira que idealizamos, já que o processo de montagem do *codebook* apresentava um problema com o parâmetro ε.

Solucionado o problema com o parâmetro ε do algoritmo de treinamento LBG modificado, construímos os dicionários de códigos para tamanhos de segmentos de voz de 10 e 20 ms e com os seguintes números de centróides: 32, 64, 128, 256, 512 e 1024. Estes dicionários de códigos foram obtidos a partir de um conjunto de seqüências de treinamento com cerca de 9000 entradas, que correspondem aos bancos de 1 a 4 devidamente preparados. A preparação destes bancos de treinamento consistia na eliminação das partes em silêncio e na fusão dos 4 bancos. Com relação a duração do processo descrito, o tempo de treinamento foi de cerca de 16 horas. Os testes envolvendo estes codebooks produziram voz reconstituída de qualidade superior ao do codificador LPC, conforme esperávamos.

Com o novo algoritmo implementado, construímos os dicionários de códigos empregando os mesmos critérios do algoritmo LBG modificado. No entanto, o conjunto de seqüências de treinamento continha exatas 8192 entradas, que correspondem aos bancos de 1 a 4 cortados após a amostra de número 8192. A razão para este número de entradas é que a rotina de implementação do novo algoritmo necessita de conjuntos de treinamento de potência de 2 para que não ocorram erros no processo de divisão em duas regiões iguais a cada iteração. O tempo de treinamento foi de cerca de 15 horas. O testes envolvendo estes codebooks produziram voz reconstituída de qualidade compatível com o método LBG modificado.

6. Análise e Resultados do Codificador CELP

O objetivo da análise do codificador CELP é avaliar os resultados obtidos através de testes de medida de qualidade objetiva como a relação sinal ruído (SNR) e o erro médio quadrático (MSE). Além destes testes, o desempenho computacional do codificador CELP também foi avaliado através do número de operações de ponto flutuante por segundo (flops).

As medidas de qualidade objetivas como a SNR procuram estimar a qualidade de um determinado codificador de voz, à medida que realizam uma comparação entre o sinal original e o sinal processado. Os métodos consistem na segmentação do sinal de voz em quadros de 10 a 30 ms de duração, e do cálculo de medidas de distorção ou distância para cada quadro. Entre estas medidas de distorção, encontra-se o erro do sinal, isto é, a diferença entre o sinal original e o sinal processado. O MSE serve tanto para avaliar a qualidade do sinal processado diretamente quanto indiretamente, uma vez que a SNR é obtida a partir do MSE.

6.1 Erro Médio Quadrático

O cálculo do Erro Médio Quadrático (MSE) é realizado a fim de fornecer uma medida de qualidade objetiva do sinal processado pelo codificador CELP. A idéia é tentar estimar a qualidade do sinal processado a partir da comparação com o sinal original.

O MSE é obtido a partir do sinal diferença entre um segmento do sinal de voz original e um segmento do sinal sintetizado. Em seguida, este sinal diferença é elevado ao quadrado e é calculada a média das amostras contidas em cada segmento de L amostras, como mostrado na equação abaixo.

$$MSE = \sum_{n=0}^{N-1} d^2(n) = \sum_{n=0}^{N-1} (s_{pesado\_original} - s_{pesado\_sintetizado})^2$$

6.2 Relação Sinal Ruído (SNR)

O SNR é a medida mais usada para sistemas de codificação analógicos e sistemas de codificação digitais usando formato de onda como por exemplo PCM e DPCM. A idéia do SNR é usar a energia do erro do sistema e compará-la com a energia do sinal de fala. O erro do sistema é a diferença entre o sinal de voz original e o sinal processado, como mostrado na equação seguinte.

$$SNR = 10\log_{10}\frac{E_{sinal}}{E_{erro}} = 10\log_{10}\frac{\sum_n s^2(n)}{\sum_n [s(n) - \hat{s}(n)]^2}$$

A medida do SNR representa o erro médio do sistema ao longo do tempo e da freqüência. A vantagem do SNR é a sua simplicidade. A desvantagem do SNR é que não foi projetado para sinais de voz, de maneira que atribui o mesmo

peso para todo o sinal de voz. Assumindo que a distorção por ruído ocorre em um faixa bastante larga com pouca flutuação de energia, o SNR deveria variar de segmento a segmento. Podemos encontrar valores muito altos de SNR em regiões de um sinal de fala com segmentos vozeados, uma vez que o ruído tem um efeito perceptual maior em segmentos de baixa energia, ou seja, segmentos não vozeados.

Uma medida de qualidade superior pode ser obtida se calcularmos o SNR segmento a segmento e tirarmos a média do resultado. Essa medida tomada a cada segmento chama-se SNR segmentado e é definido pela seguinte equação.

$$SNR_{seg} = \frac{1}{M}\sum_{j=0}^{M-1} 10\log_{10}\left[\sum_{n=1}^{l}\frac{s^2(n)}{\left[s(n)-\hat{s}(n)\right]^2}\right]$$

A segmentação do SNR permite que a medida de qualidade atribua pesos iguais a segmentos de natureza diferente do sinal de fala. Os problemas do SNR segmentado (SegSNR) aparecem quando processamos segmentos silenciosos e que resultam em resultados com valores negativos. Para evitar esse problema devemos eliminar os segmentos silenciosos antes de realizar o teste. Um outro método é fixar limiares de maneira a evitar segmentos silenciosos assim como evitar valores muito altos de SegSNR que não são percebidos pelos ouvintes. Colocando como limiares os valores de SegSNR 0 e 35 dB evitamos que a medida do teste seja tendenciosa para alguns segmentos que não contribuem para uma boa qualidade do sinal de voz como um todo.

6.3 Análise dos Resultados

Para a avaliar o codificador CELP implementado, realizamos diversos testes com a seguinte frase gravada: “Processamento de voz é muito fácil”. Estes testes consistem no processamento da referida frase, variando-se o tamanho do dicionário de códigos, tamanho do segmento de voz e o algoritmo de treinamento do dicionário de códigos. Ao final de cada medida, o sinal processado é submetido a uma análise subjetiva, usando-se a mesma escala de valores do teste MOS, mostrada abaixo.

MOS		
Pontuação	***Qualidade da Fala***	***Nível de esforço de entendimento***
5	Excelente	Sem esforço
4	Boa	Sem esforço apreciável
3	Razoável	Esforço moderado
2	Ruim	Esforço considerável
1	Insatisfatória	Muito esforço, não inteligível

Tabela 1. Escala de Pontuação do MOS.

A análise das tabelas 2 a 5 e dos gráficos 36 ao 47 mostra que o codificador CELP apresentou, como esperado, valores de SegSNR crescentes na medida que se aumentava o tamanho do dicionário de códigos. O MSE apresentou valores decrescentes, conforme esperado, na medida que se aumentava o tamanho do codebook. A complexidade computacional medida através do número de operações de ponto flutuante (flops) aumentava de forma aproximadamente linear para dicionários de códigos maiores. Além disto, a análise subjetiva também verificou uma melhor qualidade para sinais processados com maiores dicionários de código.

Com relação aos dicionários de código utilizados nos testes, aqueles obtidos através do algoritmo LBG modificado mostraram-se superiores no que se refere às medidas de SegSNR e MSE. Contudo, a análise subjetiva para dicionários do mesmo tamanho, empregando-se ambos os algoritmos, mostrou resultados idênticos.

Os sinais processados em segmentos de 10 ms mostraram-se, sem dúvida, superiores àqueles processados com segmentos de 20 ms. Em todas as medidas, os sinais processados em segmentos de 10 ms conseguiram resultados melhores, sendo que inclusive na análise subjetiva este comportamento foi verificado.

Resultados do Codificador CELP

Segmento de voz: 20ms

Tamanho do Codebook	*SegSNR(dB)*	*MSE*	*Flops*	*Análise Subjetiva*
1024	26,3433	3,7074E-04	2,0555E+09	3
512	25,7747	3,7070E-04	1,0185E+09	3
256	25,2498	3,7076E-04	4,9996E+08	3
128	25,1814	3,7074E-04	2,4070E+08	2
64	25,0893	3,7088E-04	1,2403E+08	2
32	23,4491	3,7159E-04	7,2176E+07	2

Tabela 2.

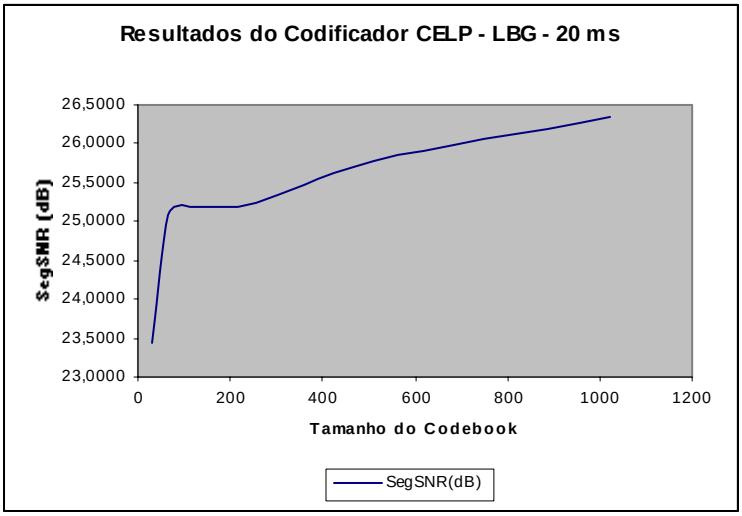

Figura 36.

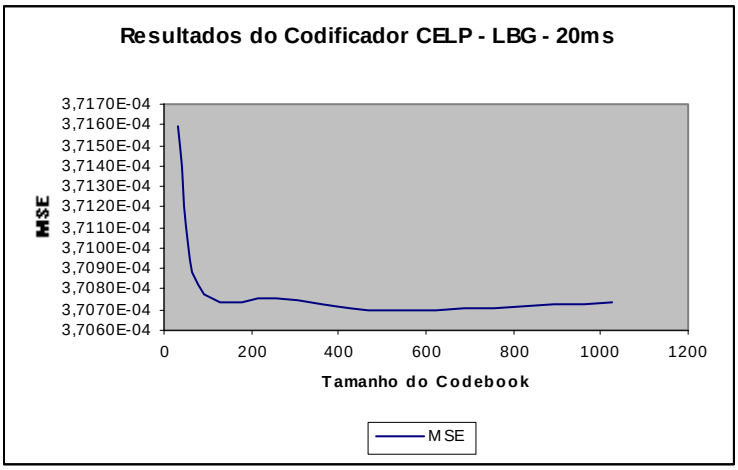

Figura 37.

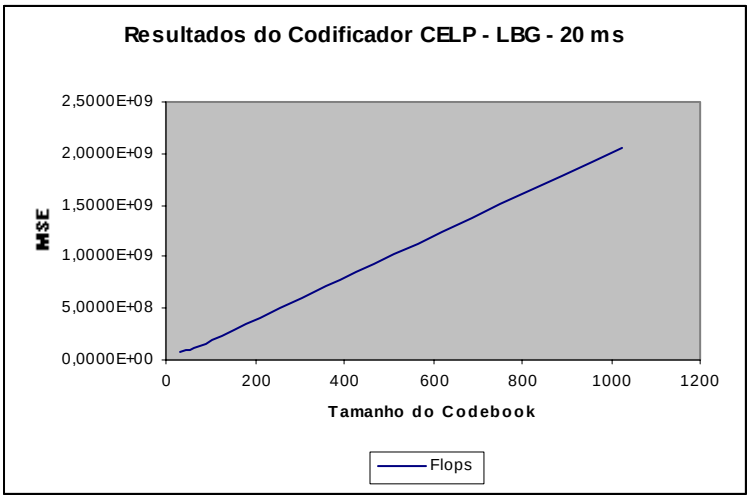

Figura 38.

Resultados do Codificador CELP

Segmento de voz: 10ms

Tamanho do Codebook	*SegSNR(dB)*	*MSE*	*Flops*	*Análise Subjetiva*
1024	21,6482	3,6473E-04	2,0395E+09	3
512	21,3469	3,6575E-04	9,9721E+08	3
256	20,8336	3,6682E-04	5,0208E+08	3
128	20,8336	3,6682E-04	2,5452E+08	3
64	20,8773	3,6682E-04	1,3724E+08	2
32	20,2465	3,6796E-04	7,2106E+07	2

Tabela 3.

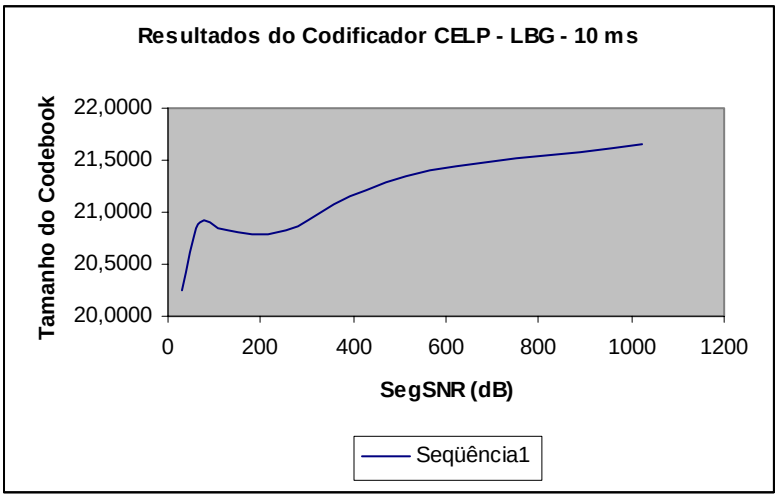

Figura 39.

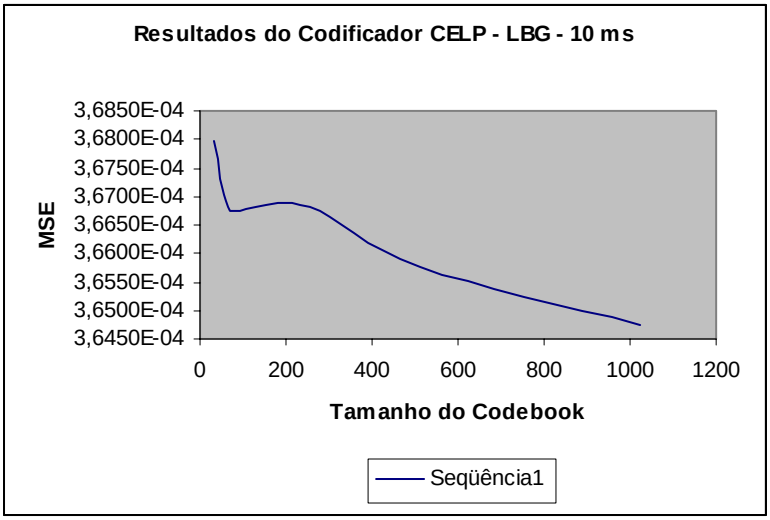

Figura 40.

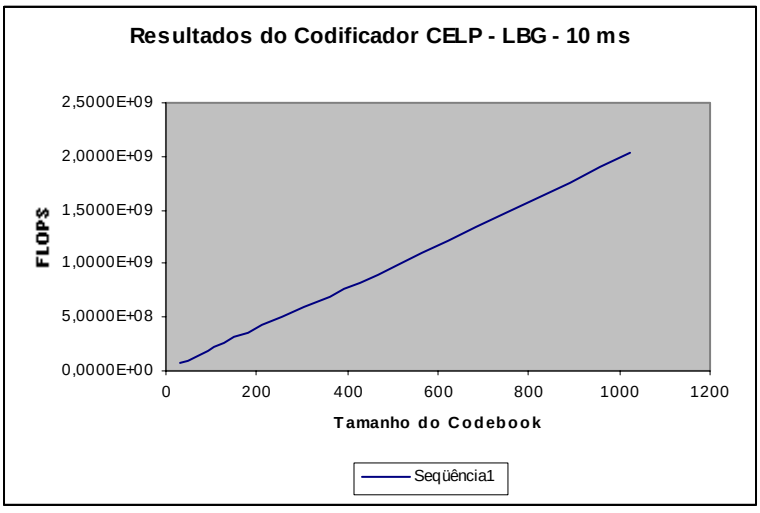

Figura 41.

Resultados do Codificador CELP

Segmento de voz: 20ms

Tamanho do Codebook	*SegSNR(dB)*	*MSE*	*Flops*	*Análise Subjetiva*
1024	25,7930	3,7095E-04	2,0813E+09	3
512	25,0435	3,7090E-04	1,0443E+09	3
256	24,2226	3,7149E-04	5,2586E+08	3
128	23,4991	3,7151E-04	2,6661E+08	2
64	23,2783	3,7128E-04	1,3698E+08	2
32	23,2348	3,7245E-04	7,2176E+07	2

Tabela 4.

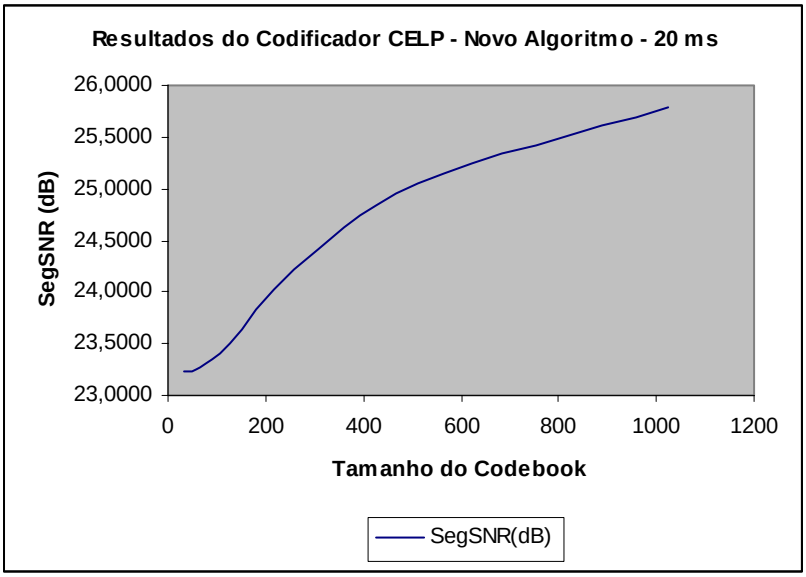

Figura 42.

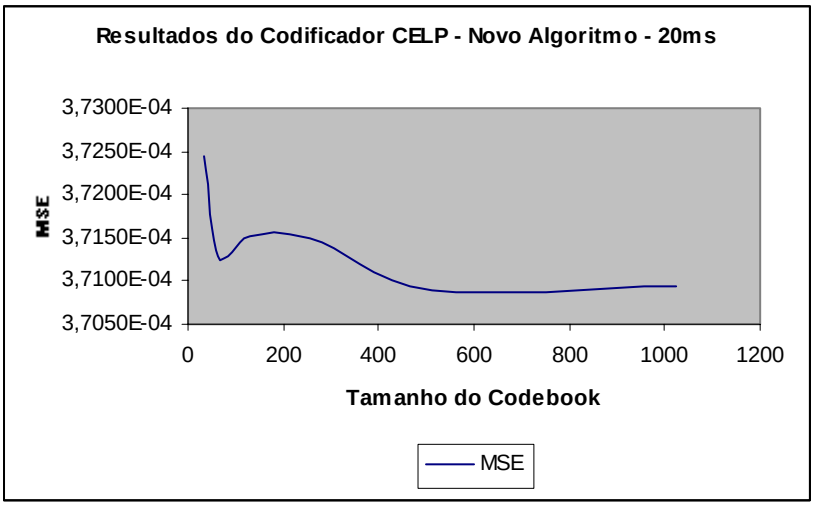

Figura 43.

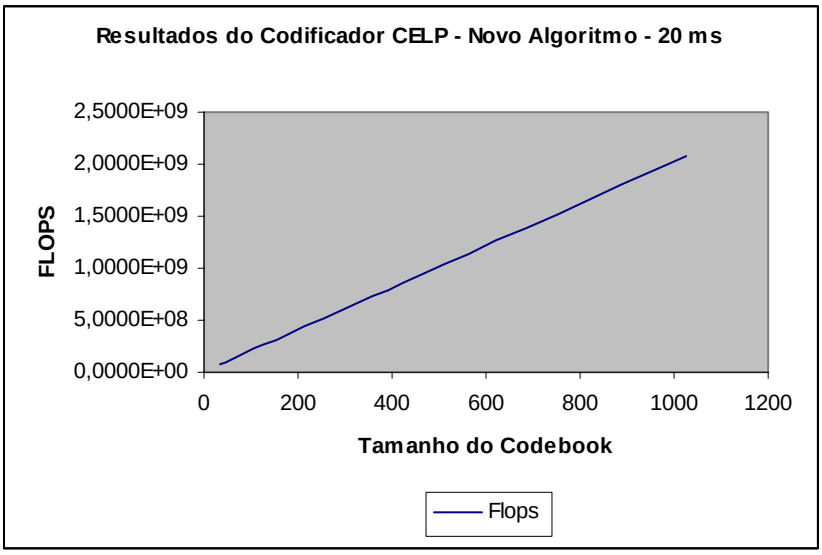

Figura 44.

Resultados do Codificador CELP

Segmento de voz: 10ms

Tamanho do Codebook	*SegSNR(dB)*	*MSE*	*Flops*	*Análise Subjetiva*
1024	21,7307	3,6519E-04	2,0813E+09	3
512	21,1234	3,6687E-04	1,0443E+09	3
256	20,5027	3,6764E-04	5,2586E+08	3
128	20,2996	3,6858E-04	2,6661E+08	3
64	20,2260	3,6973E-04	1,3698E+08	2
32	19,9106	3,6996E-04	7,2176E+07	2

Tabela 5.

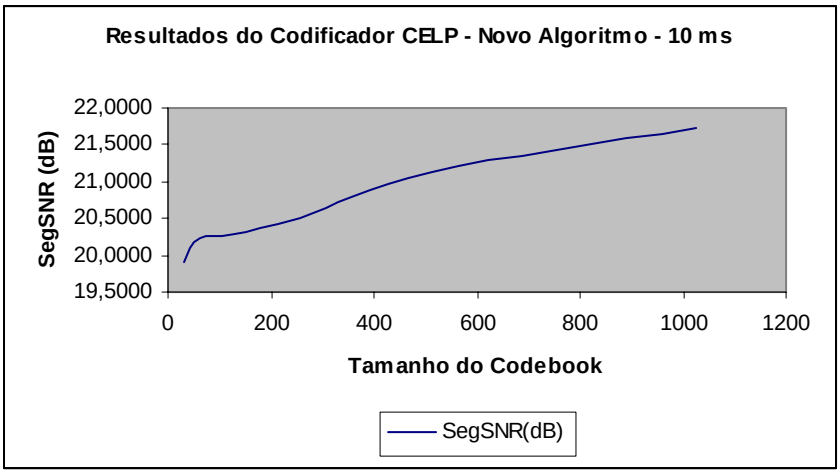

Figura 45.

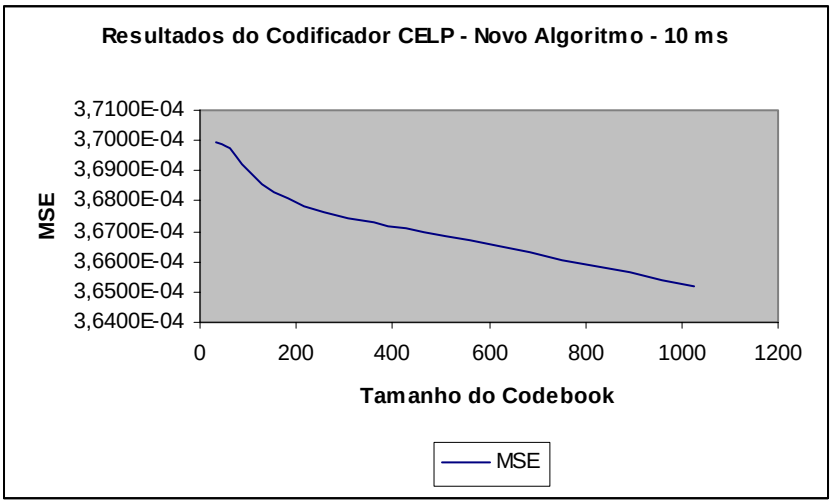

Figura 46.

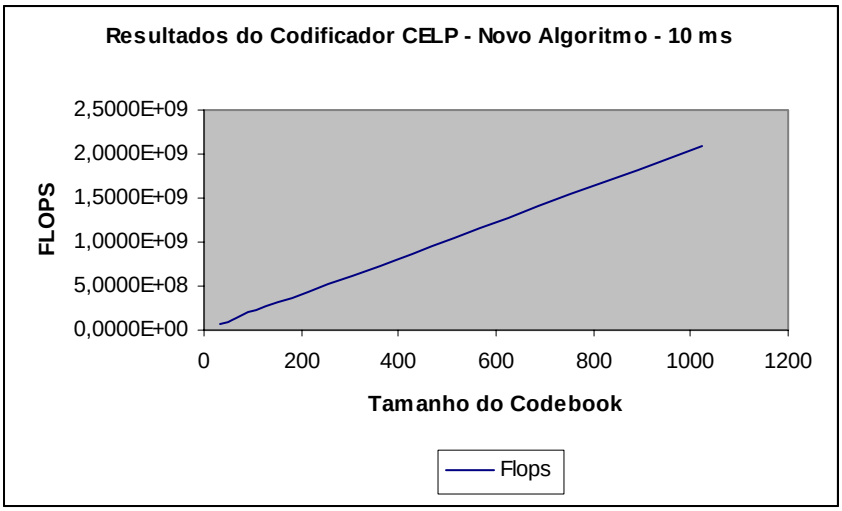

Figura 47.

7. Bibliografia

1. Digital Signal Processing – John Proakis & Dimitri Manolakis
2. Discrete-time Processing of Speech – John Deller, John Proakis & John Hansen.
3. Speech Coding: A computer laboratory textbook – Barnwell, Nayeby & Richardson.
4. Speech Coding: A tutorial review – Andreas Spanias
5. Code Excited Linear Prediction – ICASSP – IEEE – Schroeder & Atal
6. Freqüência de ocorrência dos fones e listas de frases foneticamente balanceadas no português falado no Rio de Janeiro – Alcaim, Solewicz & Moraes