Tree Encoding of Speech Signals at Low Bit Rates

by

Chung Cheung Chu
B.Eng. (Electrical)

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering

Department of Electrical Engineering
McGill University
Montréal, Canada
March, 1986

Abstract

The use of a delayed decision multi-path tree quantizer in a Code Excited Linear Predictive (CELP) coder is studied. In a CELP coder, the predictable information is efficiently removed from the input speech signals using a cascade of two time varying predictor filters. The first exploits near-sample correlations while the second exploits far-sample correlations related to the pitch excitation. The resulting prediction residual signals are low in amplitude and noiselike in appearance. The delayed decision multi-path tree encoder implemented by the (M, L)−algorithm realizes waveform coding of the prediction residual. Knowledge of human speech perception is used to define a frequency weighted error measure. This criterion is both inside and outside the tree quantizer to increase fidelity of reconstructed speech signals. The quantizer, which take into account the past history, is capable of approximating the prediction residual using a large set of de-structured codewords at fractional encoding bit rates (1/4 bit per sample). The vector quantizer used in the original CELP system studied by Schroeder and Atal is a special case of the tree quantizer. When controlled by proper combinations of five relevant parameters, the tree quantizer has a superior performance to the original vector quantizer in terms of objective performance, subjective performance, computational complexity, and memory requirement. The effects of pre-emphasizing the input signals and de-emphasizing the output signals on overall system performance are studied also. Potential further studies on the system are proposed.

Sommaire

L'étude suivante porte sur l'utilisation de quantificateurs en arbre multi-chemin à décision retardée dans un codeur à Code-Excité Predictif Lineaire (CEPL). Dans un codeur CEPL, l'information prévisible est éfficacement soustraite au signal de parole d'entrée à l'aide de deux filtres prédictifs variables placés en cascade. Le premier filtre exploite la corrélation échantillon-proche tandis que le deuxieme exploite la corrélation échantillon-lointain de l'excitation de fréquence fondamentale. Les signaux de prediction résiduelle sont de faible amplitude et ressemblent à du bruit. Le codeur en arbre multi-chemin à décision retardée, implémenté à l'aide de l'algorithme (M, L), code la prédiction résiduelle. La connaissance des caractéristiques de perception de la parole est utilisée afin de définir une mesure d'erreur pondérée en fréquence. Ce critère est appliqué dans, et en dehors, du quantificateur en arbre afin d'améliorer la qualité du signal de parole reconstruit. Le quantificateur, non causal, est capable d'approximer la prédiction residuelle en utilisant un vaste ensemble de mots de code dé-structurés à débit d'encodage fractionnels (1/4 bit par échantillon). Le quantificateur vectoriel utilisé dans le système original, étudié par Schroeder et Atal, est un cas particulier du quantificateur en arbre. Lorque commandé par une combinaison adéquate de cinq paramètres, le quantificateur en arbre à une meilleure performance que le quantificateur vectoriel en termes de performance objective, performance subjective, complexité de calcul et memoire. Les effets de la préaccentuation du signal d'entrée et de la déaccentuation du signal de sortie sont aussi étudiés. De futures améliorations au systéme sont proposées.

Acknowledgements

I would like to thank Dr. Peter Kabal for being the continual source of guidance and wise counsel throughout this study. I am deeply indebted to other members of my family especially to my parents for their continuing and invaluable support and encouragement. Also, I sincerely thank Mr. Yacine Rahmouni for his excellent work in translating the abstract of this thesis into French.

This study is supported financially by the Natural Science and Engineering Research Council of Canada (NSERC) and is carried out using the advanced facilities of the Intitut national de la recherche scientifique (INRS)—Telecommunications, Université du Quebec. Their support is gratefully acknowledged.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

Speech coding is a methodology which transforms analog speech signals continuous in time and amplitude into digitized speech signals discrete in both domains. After digitization, speech signals are represented, in general, by bit streams. Efficient and accurate transmission, storage, and digital signal processing for many purposes are then feasible. There is a trade off between the number of bits used in a coding process and the quality of the speech signals reproduced from these bit streams. In most but not all cases, better quality is achieved with a higher bit rate. The goal in speech coding is to digitize and compress signals such that the average bit rate required to represent the signals is small but the quality of the reconstructed signals is high. For instance, the present study in speech coding aims to push the bit rate down to 4.8 kilobits per second (kb/s) while maintaining toll quality.

Several advantages are immediately realizable in low bit rate speech coding. First, compressed digital speech signals require less storage space on a physical device than uncompressed signals. One of the applications of this saving is in voice mail service. Second, a given communication channel can handle more voice signals.

In a time domain multiplexed T1 carrier system, a maximum of 24 voice channels, each with a bit rate of 64 kb/s, is allowed. With the bit rate dropping to 9.6 kb/s, as many as 160 voice channels are possible [1]. Third, with high speed voiceband data modem available nowadays, digital speech coded at low bit rates can be transmitted over public switched loop circuits. Transmission of voice and data signals using the same channel can also be realized. Savings from such a digital network promise to be significant for the telecommunication industry. Fourth, compressed digital signals can be more efficiently scrambled and unscrambled. This advantage makes compressed digital signals the logical choices for secure communications [2]. The many advantages of digitally encoded speech signals over analog voice signals merit the effort in speech coding.

In general, there are three classes of speech coding techniques: waveform coding, source coding, and hybrid coding. Each class encompasses speech coding techniques processing input speech signals in the time domain and/or the frequency domain. The class of waveform coding techniques makes use of speech statistics in encoding. Time domain waveform coding techniques attempt to replicate input signal waveforms or waveforms derived from the input signals. Examples of time domain waveform coding are pulse code modulation (PCM) and differential pulse code modulation (DPCM). Frequency domain waveform coding techniques divide the frequency spectrum into bands and code each band individually. Examples of this technique are sub-band coding (SBC) and adaptive transform coding (ATC). Time domain waveform coders can be very simple and easy to implement. High quality reconstructed speech is possible with higher transmission bit rates. Typical figures for PCM and simple DPCM are 64 kb/s and 32 kb/s. Lower bit rate waveform coding is possible with more complicated time domain coders or frequency domain coders such as SBC or ATC.

Low bit rate speech coding can be provided by source coders. Source coders digitize and compress speech signals by modeling, approximating and coding the parameters describing a speech production model. A requirement of source coding is a good model of speech production including differentiation between voiced and unvoiced sounds. Voiced speech is generated by exciting the vocal tract with periodic pulse trains while unvoiced speech uses a noise excitation. The parameters necessary include voiced/unvoiced decisions, the periodic excitation rates, gain factors, and parameters describing the vocal tract. A time domain example of source coder is linear predictive coding (LPC) coder. Transmission bit rates around 2 kb/s are feasible with LPC coding. However, because of inaccuracies and rigidity in the speech production model, the reconstructed speech has a synthetic quality.

Recently, advanced VLSI technology allows design and implementation of many sophisticated hardware systems. Custom designed microprocessors permit complex coding systems to be implemented in real time. Many new speech coding designs based on combining aspects of waveform coding and source coding are possible. The purpose of this work is to obtain a hybrid speech coding technique with the properties of both high reconstructed speech quality and low bit rates. Two promising techniques have been proposed. The first is called code excited linear predictive coding (CELP) [3] and the other is called multipulse speech coding [4][5]. In a CELP coder, the residual signal after linear prediction is quantized. Indices of the approximations are transmitted. Information left in the residual signals is captured and used to excite the synthesis filters in the receiver. In multi-pulse speech coding, the excitation to the synthesis model is a sequence of samples whose positions and amplitudes are coded. The limitation or rigidity of exciting LPC inverse filters with either a periodic pulse train or a noise signal is eliminated. Good quality reconstructed signals are obtainable with either hybrid scheme with transmission bit rates between 4.8 kb/s and 16 kb/s.

A speech coding technique is studied in this thesis. This coding technique is similar to a CELP coding in the way input speech signals are processed. The complete system consists of three main parts: a prefilter, a quantizer in the transmitter and a postfilter in the decoder. Because speech signals have time varying characteristics, all three stages have to be adaptive in order to perform well. The prefilter is comprised of a cascade of predictor filters as in a CELP system to whiten input speech signals and remove predictable components from the input speech signals. Prediction residual is approximated by the quantizer by blocks of samples. The quantizer is designed on the basis of a delayed decision multi-path tree coding technique. Specifically, the $(M, L)-$algorithm is used where M is the maximum number of tree paths available after quantizing a block of gain normalized residual samples and L is the tree depth which is directly related to the delay in number of blocks of prediction residual. The size of a block is larger than one, such that fractional encoding bit rates per sample are possible. It is realized that the hearing system does not measure noise on the basis of mean squared error. Knowledge of human speech perception is exploited in the form of spectral weighting filters to help choose subjectively optimal excitation from a set of codewords such that perceptual quantization error in the receiver end is reduced. The postfilter is just a decoder of the coding system. It reconstructs speech signals based on encoded and side information received. The purpose of this study is to find out the performance of a predictive coding system with $(M, L)-$algorithm as functions of parameters M, L, block size, and a few more. The optimal set of parameters is obtained to give high fidelity in reconstructed speech with minimum amount of delay and memory requirements.

The inception of the design of a CELP system is due to Schroeder and Atal [3] while the $(M, L)-$algorithm has been studied in many literatures [6][7][8][9]. The new system differs basically from the original CELP system in the use of the tree coding quantizer. The structure of a CELP system will be reviewed in Chapter 2. The main focus will be on the design of the predictor filters and the corresponding inverse predictor filters or synthesis filters. Since the principles of designing the predictor filters and their inverses for both coding systems are identical, the designs of the predictor filters and synthesis filters for the new system will be illustrated. Chapter 3 is devoted to the study of the delayed decision multi-path tree quantizer. In Chapter 4, an additional part of the entire coding system is introduced. This new part is designed to improve the performance of the coder. Computer simulation and its results will be presented in Chapter 5. Computational complexity and other restrictions to the design and to input signals are also covered in this chapter. Chapter 6 summarizes the work to be presented.

Chapter 2 Code Excited Linear Prediction

Essentially, a code excited linear predictive (CELP) encoder divides the information in speech signals into two categories: predictable and unpredictable. Predictable information is obtained with linear prediction techniques and unpredictable information is extracted from prediction residual using waveform coding techniques. These two types of information are quantized and transmitted to the receiving end of the system to reconstruct input speech signals. Because these two pieces of information are efficiently extracted and quantized, good quality reconstructed speech signals can be obtained with transmission bit rates as low as 4.8 kb/s[†].

As shown in the block diagram in Fig. 2.1, a CELP system consists of a predictor filter and a quantizer in the encoder and an inverse predictor filter or synthesizer in the decoder. The schematic diagram of a CELP system shown in Fig. 2.1 is similar to a modified DPCM system. The basic principles behind a CELP system is identical to those of a DPCM system without quantization noise feedback [10].

Speech signals can be uniquely characterized by their short time power spectra. Essentially, every speech predictive coding technique attempts to extract in-

[†] An 8 kHz sampling frequency is assumed in this text.

formation from input speech signals such that the short time power spectra of the reproduced signals based on this extracted information are identical to the original short time power spectra. The predictor filter in a CELP system parameterizes information constituting the spectral envelope and fine structure of the spectrum. In the time domain, the spectral envelope or formant locations are reflected by near sample redundancies while the fine structure of the spectrum is related to the relatively long time delay periodicity for voiced speech. For unvoiced speech, the speech signal is noise like—it is unpredictable. A good predictor filter for extracting predictable information from speech signals can be designed as a combination of an adaptive formant and an adaptive pitch predictor filter.

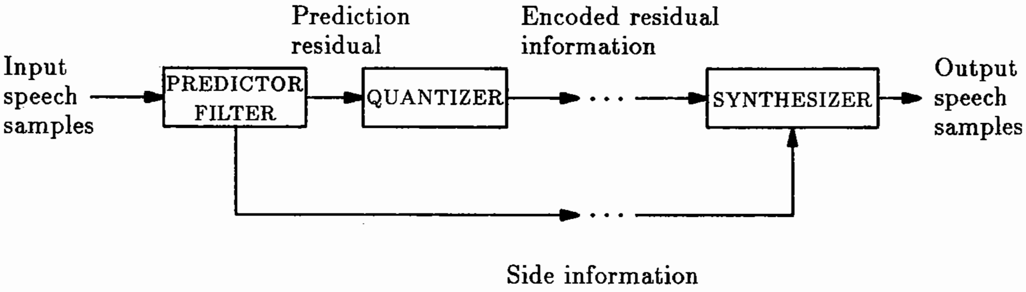


Fig. 2.1 Principal structure of a Code Excited Linear Predictive (CELP) coder

2.1 Formant Predictor Design

A formant predictor filter is designed to remove near sample redundancies in speech waveforms or equivalently to flatten the short time spectral envelope. In the present context, a feedforward nonrecursive formant predictor filter is of interest. The predictor used in the filter estimates the value of an input sample based on a linear combination of past input samples. The $z-$transform of a h^{th} order formant predictor is

$$F(z) = \sum_{k=1}^{h} a_k z^{-k} \tag{2.1}$$

where a_k's, $1 \leq k \leq h$, are predictor coefficients. For an adaptive formant predictor, the coefficients a_k's have to be updated periodically based on the input signals. The resulting predictor filter is a time varying linear filter.

There are at least three different ways to compute the a_k's. These are the autocorrelation method, the covariance method, and the lattice method. They differ from each other in terms of efficiency, complexity of the resulting predictor filter and stability of the resulting all-pole inverse predictor filter [11]. In the CELP system studied by Schroeder and Atal [3], a weighted stabilized covariance method was used to determine the a_k's [12]. In the system being studied in the text, an autocorrelation method is used instead. The autocorrelation formulation is briefly reviewed below. Detailed information can be found in [11].

In an autocorrelation formulation, the coefficients for a frame of input samples are determined by minimizing the prediction error energy $\mathcal{E}_f$. If $\{x_n : 1 \leq n \leq T\}$ with $x_n = 0$ for $n < 1$ or $n > T$ is the input frame and $\tilde{x}_n$ is the predicted input for x_n, then

$$\begin{aligned} \mathcal{E}_f &= \sum_{n=1}^{T+h} (x_n - \tilde{x}_n)^2 && 2.2.a \\ &= \sum_{n=1}^{T+h} (x_n - \sum_{k=1}^{h} a_k x_{n-k})^2 && 2.2.b \end{aligned}$$

The summation limits in Eq. (2.2) are so defined because x_n is assumed zero for $n < 1$ and $n > T$. Differentiating $\mathcal{E}_f$ with respect to each a_k for $1 \leq k \leq h$ and setting the resulting equations to zeros give a set of linear equations. This linear equation system can be expressed in matrix notation as

$$\begin{bmatrix} \phi(1,1) & \phi(2,1) & ... & \phi(h,1) \\ \phi(1,2) & \phi(2,2) & ... & \phi(h,2) \\ ... & ... & ... & ... \\ \phi(1,h) & \phi(2,h) & ... & \phi(h,h) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ ... \\ a_h \end{bmatrix} = \begin{bmatrix} \phi(0,1) \\ \phi(0,2) \\ ... \\ \phi(0,h) \end{bmatrix}$$

where

$$\phi(k,j) = \sum_{n=1}^{T+h} (x_{n-k} x_{n-j}) \quad 0 \leq k \leq h \text{ and } 1 \leq j \leq h$$

Because x_{n-k} and x_{n-j} are samples from the same input frame defined within the interval $[1,T]$

$$\phi(k,j) = \phi(j,k)$$

$$= R(|j-k|)$$

The term $R(\cdot)$ is the autocorrelation function of x_n. The square matrix is a Toeplitz matrix, and the coefficients can be computed with efficient methods.

Because x_n is zero outside the interval $[1,T]$, error sample values $(x_n - \tilde{x}_n)^2$ are large near both ends of the interval $[1, T+h]$ when they are compared to real error sample values located in the middle of the interval $[1, T+h]$. The input sample sequence $\{x_n\}$ should be smoothed out near both ends of the interval $[1,T]$ to reduce the effect of this undesired error on both ends of the prediction error sequence. The relative importance of the undesired error in a prediction error sequence can also be reduced if T is large.

The coefficients obtained using the autocorrelation method always give a stable inverse predictor filter (an all-pole formant synthesizer) if the computation is performed with sufficient precision. The amplitude of the transfer function of the inverse predictor filter can be used to define an approximated spectral envelope.

The predictor order h (samples) is basically a function of sampling frequency and the length of the vocal tract from which the input speech signals are generated. In practice, an 8^{th} order formant synthesizer is enough to represent the vocal tract contribution with an 8 kHz sampling frequency. However, four or five more poles are necessary in the formant synthesizer to represent the glottal and lip radiation effect [13].

2.2 Pitch Predictor Design

The pitch predictor used in this study is identical to the one used by Schroeder and Atal in their CELP system. The pitch predictor is a 3 tap transversal filter with $z-$transform

$$P(z) = \sum_{j=-1}^{1} \beta_j z^{-(p+j)}$$

where β_j's, $-1 \leq j \leq 1$, are the tap coefficients of the transversal filter and p should correspond to a pitch period for voiced speech. In this formulation, the value of p is chosen as the delay which gives maximum normalized correlation value for the input signals. Because a real pitch period may not be an integral multiple of sampling periods, two input samples $y_{n-(p-1)}$ and $y_{n-(p+1)}$ around y_{n-p}, where y_n is an input sample to the pitch predictor, are considered also to obtain minimal prediction error [14]. Thus a 3 tap pitch predictor is used.

The coefficients can be obtained by minimizing the mean-squared prediction error. As opposed to the autocorrelation method discussed in previous sections, past input samples are required for long time delay prediction. If the mean-squared error for a frame of S new input samples is defined as

$$\mathcal{E}_p = \sum_{n=1}^{S} (y_n - \tilde{y}_n)^2 \qquad S \geq 1$$

where $\tilde{y}_n$ being the predicted value of y_n, then

$$\mathcal{E}_p = \sum_{n=1}^{S} (y_n - \sum_{j=-1}^{1} \beta_j y_{n-(p+j)})^2 \quad S \geq 1$$

Input samples required for the analysis are $\{y_n : -p \leq n \leq S\}$. Long time delayed samples are needed. A set of linear equations can be obtained if $\mathcal{E}_p$ is differentiated with respect to the β_j's. The coefficients can be computed by solving the linear equations

$$\sum_{n=1}^{S} y_n y_{n-(p+i)} = \sum_{j=-1}^{1} \beta_j \sum_{n=1}^{S} y_{n-(p+j)} y_{n-(p+i)} \quad \text{for} \quad -1 \leq i \leq 1$$

For unvoiced speech, the coefficients β_j's usually have very small values and the value of p is not important.

The predictor filter output signals consist of unpredictable speech components such as noise in silent portions and unvoiced speech. These unpredictable signals still contain valuable information. Logically, they are good choices of excitations to the inverse predictor filters in the receiver.

2.3 Vector Quantizer Design

The residual signal quantizer is essentially a waveform coder. It looks for good approximations from a set of noise samples to represent the prediction residual. Because input signals to the quantizer are prediction residual, correlation between input samples is low. Besides, variance of prediction residual signals is small. Waveform quantization can be performed more efficiently. In the CELP design studied by Schroeder and Atal, a codebook quantizer encodes blocks of 40 residual samples (5 ms signals with an 8 kHz sampling frequency). For every input block of residual samples, the same set of 1024 codewords is used as candidate output sequences. After subtracting the input block from each codeword, the quantizer synthesizes the difference sequences individually. For the selection process an adaptive noise filter is used inside the quantizer to weight each synthesized sequences. Reconstructed noise is spectrally weighted in the quantizer so that the output noise spectra are masked everywhere by short time signal spectra along the frequency axis. The index of the codeword which gives minimum weighted error energy is transmitted to the receiver. Except for the computational and block delays, the quantization process is instantaneous.

2.4 Decoder Design

The structures of the receiver in a CELP system and the system studied in this text are simple compared to that of the transmitter. Essentially, it consists of a time varying linear inverse pitch predictor filter $(1 - P(z))^{-1}$ followed by a time varying linear inverse formant predictor filter $(1 - F(z))^{-1}$ where $P(z)$ and $F(z)$ are the $z-$transforms of the adaptive linear pitch and formant predictors. After receiving a codeword index, the decoder does codebook lookup to fetch the corresponding codeword from a codebook which is identical to the codebook used in the encoder.

Each codeword is then sent to the inverse predictor filters determined by the side information to reconstruct speech signals.

A code excited linear predictive coder designed in this way is capable of efficiently encoding speech signals with transmission bit rates as low as 4.8 kb/s. Because short time spectrum is reproduced with encoded residual signals rather than noise or periodic pulse trains as excitation, the reconstructed speech quality is better than that of a LPC coding system. The rest of this thesis will study the CELP system with a delayed decision multi-path tree quantizer. The objective and subjective performance will be evaluated as functions of different control parameters.

Chapter 3 Prediction Residual Quantization

In practice the predictor filters designed to remove redundancies from input speech leave some speech components which are unpredictable. The output samples of the prefilters are not completely independent and identically distributed. Information remained in the residual signals is important and crucial to the intelligibility and especially to the naturalness of the reconstructed signals at the receiving end of the system. Unlike an LPC vocoder which transmits only the predictable information, the system studied here has a quantizer which is designed to capture this residual information using low bit rates. The quantizer designed has a very different structure from an ordinary scalar quantizer employed in a waveform coder such as a PCM or a DPCM system. A block quantization scheme is used to maintain a low bit rate. To take into account perceptual effects of the reconstructed noise, a tree encoder with embedded noise shaping filters is employed. The block quantizer used in a CELP system is a special case of the resulting quantizer. The structure and design of this quantizer will be discussed in this chapter in detail. In the sequel, the terms quantizer and encoder will be used interchangeably.

3.1 Delayed Decision Coding

As opposed to instantaneous decision coding such as the PCM coding system, a delayed decision encoder makes use of artificial delays to code quantizer input signals. Artificial delays include any delay generated on purpose in an encoder. The form of an artificial delay varies with the particular method or algorithm in use to implement the quantizer. It may be a delay generated by a quantizer which makes use of the quantizer inputs at sample instant $i+j$ to determine the optimal approximation to an quantizer input at sample instant i. Another form of artificial delay may be a delay in encoding input sequences by blocks. In that case, the encoder has output sequences in block form also. The input sample values are sent serially to the encoder, whereas the corresponding output sample values are specified simultaneously and output in parallel. In general, the first form of artificial delays is found in tree or trellis encoders. The latter kind of artificial delays is used by the so called codebook encoder or vector quantizer.

A proper choice of artificial delay can make a quantizer perform better in terms of signal-to-quantization-noise gain and/or reduction of transmission bit rates. A memoryless encoder chooses an output sample instantaneously after the arrival of an input sample. This irrevocable decision may be the best match to the input sample according to a distortion measure, however it may not necessarily be optimal when its long term effect on the entire reconstructed signal is concerned. On the other hand, the advantage of delayed decision coding in a tree encoder is that excitation sequences for reproducing speech signals are selected only if they satisfy a long term distortion measure. The other advantage of this form of artificial delay is that more candidate output sequences are allowed in the selection process. This point will be discussed in more detail in the section on tree coding. Block coding allows the realization of encoding at a bit rate of $R/N < 1$ bit per sample. The encoding bit rate of an optimum source code approaches the lower bound specified by the rate-distortion function as the block size N approaches infinity [15].

The quantizer studied here uses these two types of artificial delays. While the selection process is implemented to minimize the expected long term perceptual error, the encoder processes blocks of samples to obtain fractional encoding bit rates. In other words, with delayed decision and block coding the long term performance can be optimal with fractional encoding bit rates.

3.2 Tree Coding

Theoretically, a tree quantizer can perform arbitrarily close to the rate-distortion bound for memoryless sources [16]. A tree used in an encoder can be classified as either a single path or a multi-path tree. The differences between these two kinds of trees are in terms of the number of candidate output sequences considered for each quantizer input sequence and the depth of the tree which controls the artificial delays involved in encoding an input sequence. Because of the increase in the number of candidate output sequences and the delay, a multi-path tree encoder is in general more complex than a single path tree encoder at the same encoding bit rates. Before the discussion of a multi-path tree encoder, it is worth discussing the structure of a single path encoder to explain a few definitions common to both tree structures.

3.2.1 Single Path Tree Encoder

A tree is characterized by its nodes, branches and leaves. In a single path tree encoder, a node of the tree at a specific time instant is associated with the optimal quantizer output sequence approximating a quantizer input sequence of the same size according to a chosen distortion measure. Suppose the encoding bit rate is R bits per block and an input block sequence to be quantized at a time has a constant size $N = 1$ sample. Then the encoding bit rate is R bits per sample. In response to the input at the next time instant, $2^R = 2^{RN}$ branches stem from the node which was generated after the best approximation for the last input sample has been selected. The leaf of each branch has a candidate output sample associated with it. The encoder chooses from these 2^R choices the best sample to approximate the input sample. After the best match has been selected, the tip or leaf of the selected branch becomes a node for the next matching cycle. The index of the branch associated with an output sample is the information to be transmitted. The optimal branch numbers form an innovation sequence or a path map which is transmitted to the receiver. The output sequence formed by the optimal samples selected in the encoder is determined in the decoder by tracing along the path map. This simple sequential tree coding technique is found in many popular encoding systems such as PCM and DPCM systems.

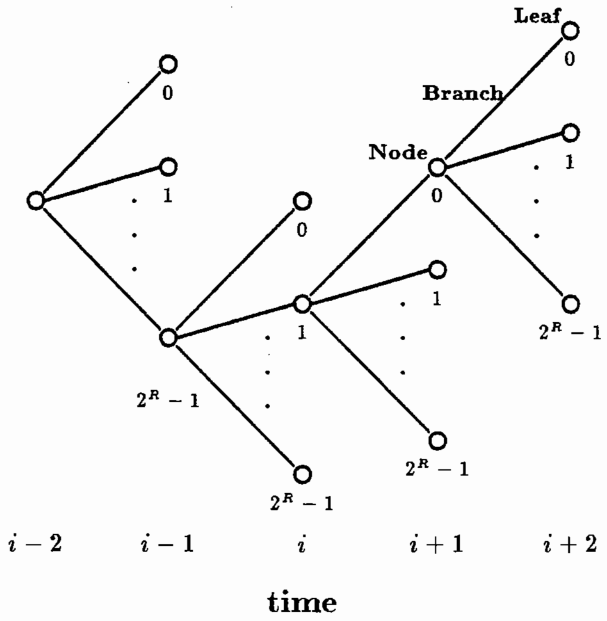


Fig. 3.1 Single path tree encoding

In the coding mechanism discussed above, population of the tree continues until the last input sample is approximated. The selected branches form a single continuous path connected together at the nodes, and hence form a single path encoder. Since the selection is performed in the forward direction without delay, the optimal branch numbers can be transmitted nearly instantaneously.

3.2.2 Multi-path Tree Encoder

There is only one node available at a sample instant in a single path tree encoder. The maximum number of candidate output sequences considered is 2^R. With the use of delayed decision, it is possible to have more than one node and thus more than 2^R sequences available at a time to encode an input sequence without increasing the encoding bit rates. A tree encoder capable of doing this is called a multi-path

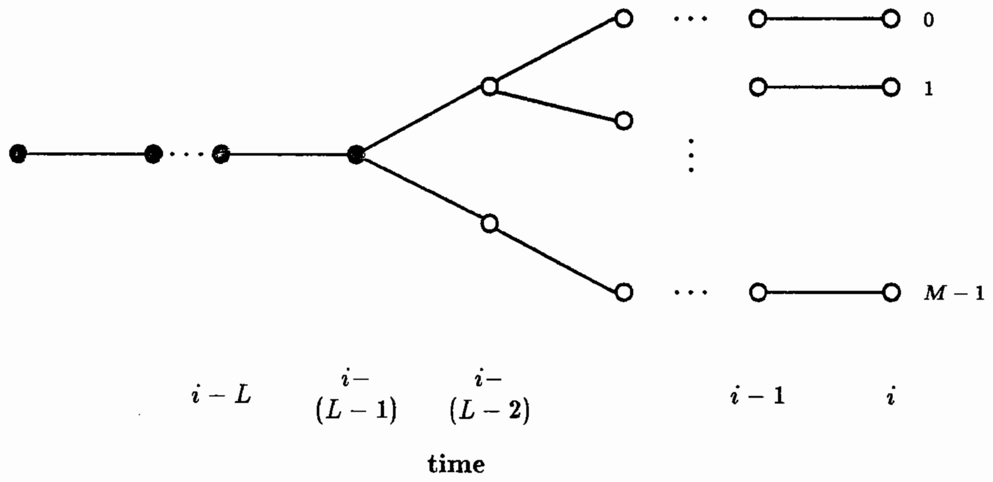


Fig. 3.2 Multi-path tree encoding

tree encoder. To illustrate the structure of a multi-path tree encoder, a special algorithm that implements multi-path tree encoding is discussed. This is known as the $(M, L)-$algorithm which has been studied in different literatures and will be used in the system studied here [6][7][8][9].

3.2.3 $(M, L)-$Algorithm

The $(M, L)-$algorithm is a search algorithm which assumes a tree with a maximum of M nodes after an input block of samples is approximated. The tree depth L is the number of branches in series required in the path selection process. The resulting tree is like a trellis. However, a trellis has constant number of nodes at different levels after an initial fanout period in which the trellis grows exponentially. The tree due to the $(M, L)-$algorithm grows gradually. The number of nodes at different levels is non-decreasing with time in this tree. The total number of nodes at the final level is upper bounded by M.

As mentioned in previous sections, quantization is done in blocks of N samples. Assume there are M nodes after the i^{th} input block as shown in Fig. 3.2 is processed. Each node signifies the existence of a path, and there are M paths present. These M paths converge at a root node which is $L-1$ branches behind at time $i-(L-1)$.

Suppose the encoding bit rate is R bits per block of input residual samples. Then, 2^R branches are populated from each one of the M nodes for the $(i+1)^{st}$ quantizer input block. The branches are numbered from 0 to $2^R - 1$. Associated with the leaf of each branch is a possible output block. It is clear at this point that there are M times more candidate output blocks available in this tree than in a single path tree provided there exist M paths and the encoding bit rates are R bits per block. Because there is more than one set of branches with the same branch numbers, the address of a block of output samples on the leaf of a particular branch has to be computed based on information other than a single branch number. A unique mapping will be defined in later sections of this chapter to find the address of an output block based on a set of branch numbers without increasing the amount of information transmitted to the receiver such that all the codes may be different.

In the quantization process, the best candidate output block out of $M2^R$ blocks is chosen according to a long term distortion criterion to be described later. Upon the selection of a branch at time $i+1$, the output block on the node at time $i-(L-2)$ leading to this branch is considered as the optimal approximation to the $(i-(L-2))^{th}$ quantizer input block. The index of the branch connecting the root node at time $i-(L-1)$ and the optimal node at time $i-(L-2)$ is transmitted. Each index is transmitted $L-1$ blocks after the corresponding input block has been considered. A maximum of M paths are chosen from a subset of the $M2^R$ possible paths. Each path in this subset has to converge to the newly found optimal node. These new paths are determined according to the same long term distortion criterion. The path linking the optimal node at the $(i-(L-2))^{th}$ level and the optimal branch at the $(i+1)^{st}$ level is always included. It is important to see that these paths are all continuous and convergent to the already output optimal path which is defined up to time $i-(L-2)$. The encoding process is repeated for further input blocks.

The $(M, L)-$algorithm defined in the last paragraph is slightly different from that used by Svendsen, Jayant, and Christensen [7][9]. In their studies, M paths are first selected from $M2^R$ paths, then the optimal path from these M paths is chosen. The optimal output for the $(i-(L-2))^{th}$ input block is found by tracing backward along this optimal path. After the optimal block is located, only a subset of the M paths chosen does diverge from this optimal node and remains. The rest of the paths are invalidated. Although both approaches attempt to keep as many as M paths for the next cycle, the modified approach used in this system achieves this goal more closely. This is true because the original approach has some of the M paths originating from suboptimal nodes, while some of the paths diverging from the optimal node are not considered. In the modified approach, all paths diverging from the optimal node are considered in the selection of M new paths. Therefore, more paths are kept in the modified approach.

Because the branching factor of a node is 2^R, the maximum number of nodes available is $(2^R)^{L-1} = 2^{R(L-1)}$ for a tree with depth L. This puts an upper bound to the size of M with

$$M \leq 2^{R(L-1)}$$

In the case of $L = 1$, only one branch is involved in encoding. The multi-path tree encoder collapses to a single path tree encoder with $M = 1$ which is commonly called a codebook quantizer for $N > 1$ and a scalar quantizer or a single path memoryless tree encoder for $N = 1$. In the system studied both M and L are varied. When $M = 2^{R(L-1)}$, all possible paths diverging from an optimal node are considered. This exhaustive search is therefore optimal. A suboptimal approach is in force if M is less than $2^{R(L-1)}$. In the case of $M = 1$, there exists only one node or one path after quantizing a quantizer input block. All 2^R branches are populated from this single node in the next stage. It is useless to have $L > 1$ because the optimal branches to be output are on this single path which is known. Therefore when $M = 1$, L has to be equal to 1, and vice versa.

3.3 Distortion Measures

In response to a block of N prediction residual samples, the multi-path tree prepares at most $M2^R$ codewords of the same size. One of these codewords will be chosen as the optimal approximation to the input block. In order to take advantage of the delayed decision encoder and knowledge of human perception, the distortion measure cannot be based on the mean-squared error criterion applied independently in each time instant. A codeword is chosen not because the power of the difference sequence is minimum among difference sequences of other codeword and input pairs.

The rest of this section discusses a set of more meaningful and useful error measures employed in the system being studied.

3.3.1 Noise Shaping

According to noise masking theory, the optimal short time spectrum of the reconstructed noise should have an envelope resembling to that of the system input speech. Because signal energy levels around formant regions of speech signals are high, distortion at these frequencies is masked by input speech signals. Most subjective distortions are due to noise in the regions between formants. As a consequence of this, the noise between formants should be weighted more relative to that at the formants. A frequency weighting filter used in the optimality criterion is required to suppress the noise power around formant regions and to increase the noise power in the regions between formants where the error is perceptually more important.

In a differential pulse code modulation system the power spectral density of quantization error is properly weighted to reduce perceptual effect of noise at the receiving end with the use of a quantization noise feedback path. On the contrary, the system studied here is characterized by a feedforward configuration. Nevertheless, knowledge of speech perception can still be used to minimize subjective noise to achieve high fidelity in the reconstructed signals. An analysis-by-synthesis model is used to achieve this goal.

Figure 3.3 shows the synthesis part of the quantizer on each branch. As in Chapter 2, $P(z)$ and $F(z)$ stand for the $z-$transforms of the adaptive linear pitch predictor and formant predictor in the prefilter of the system. In the decoder, the

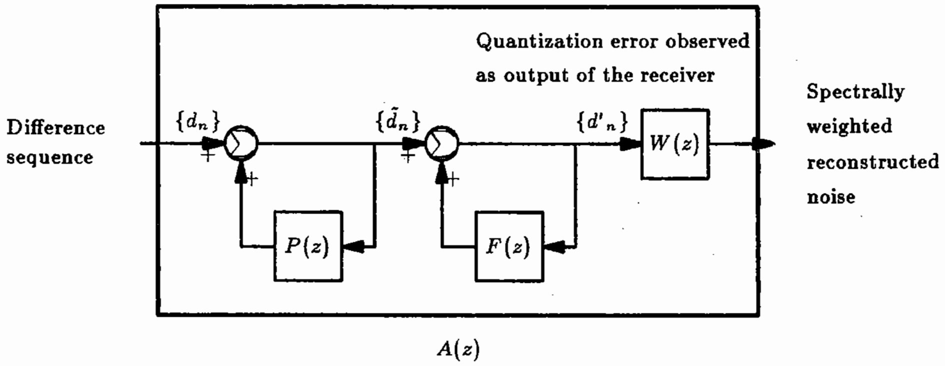


Fig. 3.3 Design of a filter combination on a tree branch

reconstructed noise is the output of adaptive pitch and formant inverse filters $(1-P(z))^{-1}$ and $(1-F(z))^{-1}$ in response to the difference between the quantizer input sequence and its optimum output sequence. The reconstructed noise is anticipated and simulated in the quantizer for each possible output block. Because the number of samples in a block is more than one, spectral shaping on quantization noise is feasible. The filter $A(z)$ is the cascade of $(1-P(z))^{-1}$, $(1-F(z))^{-1}$, and an adaptive frequency weighting filter $W(z)$.

It has to be emphasized that there is no spectral weighting in the receiver at all. In the system, spectral shaping in the quantizer does not change the real noise output observed at the receiver as in the case of noise feedback coding [12]. The reconstructed noise spectrum and average noise power level are not shaped because the weighting filter is present in the quantizer only. The objective of designing a weighting filter is to help choose codewords which are subjectively optimal.

3.3.2 Choice of Weighting Filter $W(z)$

Because speech is a non-stationary process, the locations of formant and interformant frequencies in a short time spectrum are time varying. A band of frequency weighting filters are installed on all branches of the tree. The weighting filters have to be adaptive to the short time spectral envelope of the system input speech. Therefore, the weighting filter $W(z)$ to be designed should be related to the $z-$transform of the formant predictor filter.

The weighting filter being sought should be able to transform the shape of a noise spectrum into a power measure. The filter excited by a subjectively optimal noise with respect to the input speech power spectrum should give an output with minimum average objective power level. Since minimization of mean-squared error implies choosing the whitest noise, the subjectively optimal noise should have the whitest power spectrum at the output of the weighting filter.

The filter is so designed that its characteristics are controlled by a parameter γ which is a real number between zero and one inclusively. Motivated by the design used by Schroeder and Atal, the following weighting filter is employed:

$$W(z) = \frac{1 - F(z)}{1 - F(\gamma^{-1}z)} \quad 0 \leq \gamma \leq 1 \tag{3.1}$$

where $F(z) = \sum_{k=1}^{h} a_k z^{-k}$ is the h^{th} order adaptive linear formant predictor. When $\gamma = 1$, the transfer function of the filter is 1. With such a design, the output spectrum of noise is identical to the input spectrum of noise to the weighting filter. If the input is white noise, the output will be white noise also. When $\gamma = 0$, the noise input with a spectral envelope identical to that of the input speech signals is whitened. When γ assumes an intermediate value between 0 and 1, a power spectrum resembling to the short time power spectrum of input speech to any desired degree can be whitened by $W(z)$.

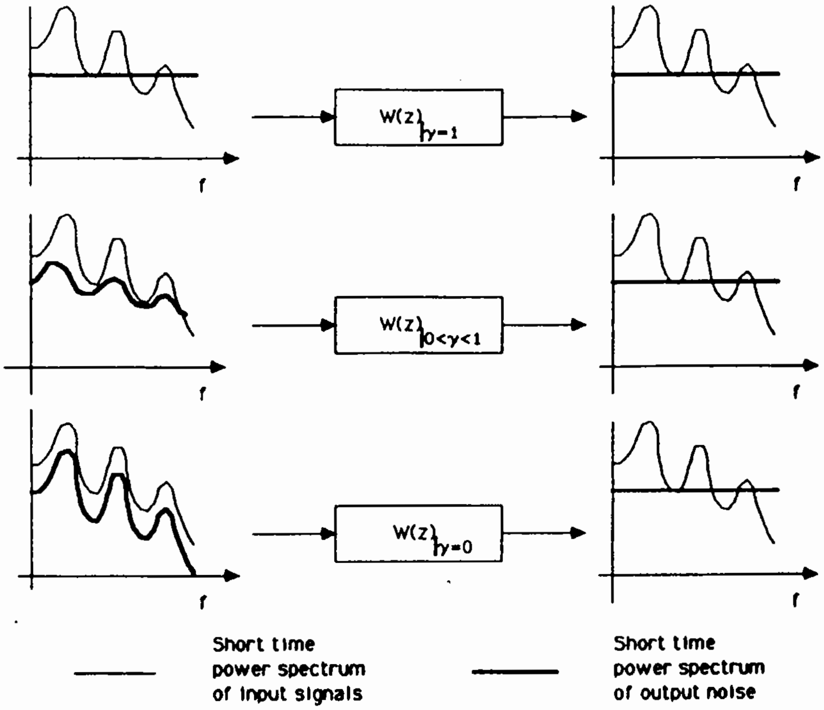

f
$W(z)|_{\gamma=1}$
f
f
$W(z)|_{0<\gamma<1}$
f
f
$W(z)|_{\gamma=0}$
f
Short time power spectrum of input signals
Short time power spectrum of output noise

Fig. 3.4 Comparison of effects of a weighting filter for different values of γ

A filter so designed will have operations as illustrated in Fig. 3.4. When $\gamma = 1$, quantization noise components in regions between formants are audible. Perceptual effect of quantization noise can be reduced with $\gamma < 1$. On the other hand, human ears are not equally sensitive to noise at all frequencies, the value of γ is usually larger than zero. The typical value of γ falls in the range [0.7, 0.9] for low bit rate encoding. The exact value of γ which gives best system performance in subjective listening tests is not critical [17].

The final structure of filter $A(z)$ is shown in Fig. 3.5. Effectively, the filter $A(z)$ is a cascade of $(1-P(z))^{-1}$ and $(1-F(\gamma^{-1}z))^{-1}$. With $\gamma=0$, the resultant transfer function of $(1-F(\gamma^{-1}z))^{-1}$ is 1 and with $\gamma=1$ the transfer function is $(1-F(z))^{-1}$. Hence the effect of cascading the inverse formant predictor filter with $W(z)$ with $\gamma<1$ is to increase the bandwidths of the zeros of $(1-F(z))^{-1}$ and to lower its amplitude in the formant regions.

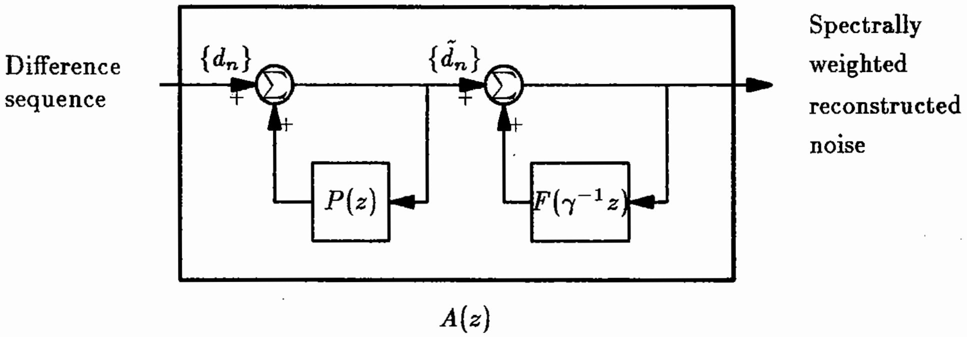


Fig. 3.5 Effective synthesis filters on each tree branch of the delayed decision multi-path tree

3.3.3 Cumulative Error Criterion

The error measure is a very important factor to the performance of the system. More specifically, the error measure chosen for the encoder determines how the tree is populated and how the paths are constructed. As mentioned before, an advantage with delayed decision coding is that the reconstructed signals can be selected based on the minimization of long term distortions. The effect of distortion in a multi-path tree caused by approximating a quantizer input block with an output block extends to the future and is then evaluated. The quantizer is designed to choose at time instant $i+1$ the optimal output block for the $(i-(L-2))^{th}$ input block such that the best long term distortion E_{opt} is

$$E_{\text{opt}} = \min_j \Big[\sum_{k=0}^{i+1} e_j^2(k)\Big] \qquad 0 \le j \le K2^R - 1 \tag{3.2}$$

where $e_j^2(k)$ is the energy of the spectrally weighted error sequence between the input block and the output block on the j^{th} branch of the tree at time k, and K is the number of nodes chosen at time i[†]. Put differently, accumulated errors along the paths are compared to determine a delayed output. It is shown in Fig. 3.2 that the accumulated error

$$\sum_{k=0}^{i-(L-1)} e_j^2(k)$$

is constant for all j's because all the paths diverge from the already determined node at the $(i-(L-1))^{th}$ level. Equation (3.2) can be rewritten as

$$E_{\text{opt}} = \min_j \Big[\sum_{k=i-(L-2)}^{i+1} e_j^2(k)\Big] \qquad 0 \le j \le K2^R - 1 \tag{3.3}$$

Or more generally, the optimal delayed decision is made for the i^{th} input block according to the cumulative error measure

$$E_{\text{opt}} = \min_j \Big[\sum_{k=i}^{i+L-1} e_j^2(k)\Big] \qquad 0 \le j \le g - 1 \tag{3.4}$$

with g being the total number of candidate output blocks available for the $(i+L-1)^{st}$ input block. The effect of the error energy $e_j^2(i)$ at time i will be considered in the future at times $i+1$, $i+2$, ..., and $i+L-1$ before a decision on the best block

[†] The term $e_j^2(k)$ can also be considered as the mean-squared spectrally weighted error. The constant normalization factor $1/N$ required in this interpretation has no effect on the minimization process.

of excitation waveform for the i^{th} block of quantizer input is made. Therefore the artificial delay increases with tree depth L.

3.4 Generation of Output Codes

In the design of the adaptive whitening filters in the encoder, system input speech signals have most of their redundancies removed. For the purposes of populating the codewords, prediction residual can be assumed to consist of samples of independent and identically distributed random variables. Studies by Atal and Schroeder show that normalized distribution of the residual speech process except in unvoiced stop and unvoiced silent to voiced speech transitions can be approximated by a Gaussian distribution [12][3]. Also the prediction residual is white during steady speech portions [18]. Hence, the normalized prediction residual can be assumed to be white Gaussian noise with zero mean and unit variance.

Output sequences used in this system are populated by a Gaussian random number generator. Noise samples from the Gaussian random number generator are divided into successive blocks of size N to form codewords. Two identical copies of the codebook generated in this way are stored in the encoder and the decoder respectively.

Output sequences so generated are in general better than deterministically generated output sequences because the quantizer input statistics are well matched in the former case [10]. Nevertheless, the encoder may still perform differently with different codebooks. It has been shown for memoryless Gamma source that better performance can be achieved with an output sequence optimized iteratively with a set of input training sequences [10]. Since the main objective of this study is not the study of a codebook coder, optimization of output sequences with input training sequences is not explored further.

3.4.1 Relations between Branch Numbers and Codeword Addresses

It is known that the output of the quantizer is a sequence of indices of the optimal branches which form the optimal path in approximating the prediction residual. It is assumed then that the codewords associated with the branches are given and known to both the encoder and the decoder. How the codeword addresses are determined and loaded from the codebook onto the tree has not been discussed. This section discusses how the encoder and the decoder address the correct codewords based only on sequences of branch numbers.

Suppose K paths, $K \leq M$, are kept after quantizing the $(i-1)^{st}$ block of residual signals. There are K nodes on the tree. In response to the input of the i^{th} block of prediction residual samples, a total of $K2^R$ branches are populated from these K nodes, where R is the number of bits per block and 2^R is the branching factor of a node. A branch is completely specified if the node from which this branch is populated and the branch number are given. If a different combination of node and branch numbers implies a different codeword address, $K2^R$ codewords are necessary. However, the structure of the tree encoder allows only the transmission of branch numbers to the receiver. Communication of node information increases the encoding bit rates and the load on the channel. Each codeword address must be properly defined as a function of branch numbers such that it may be recalled in the receiver based on the received indices or branch numbers.

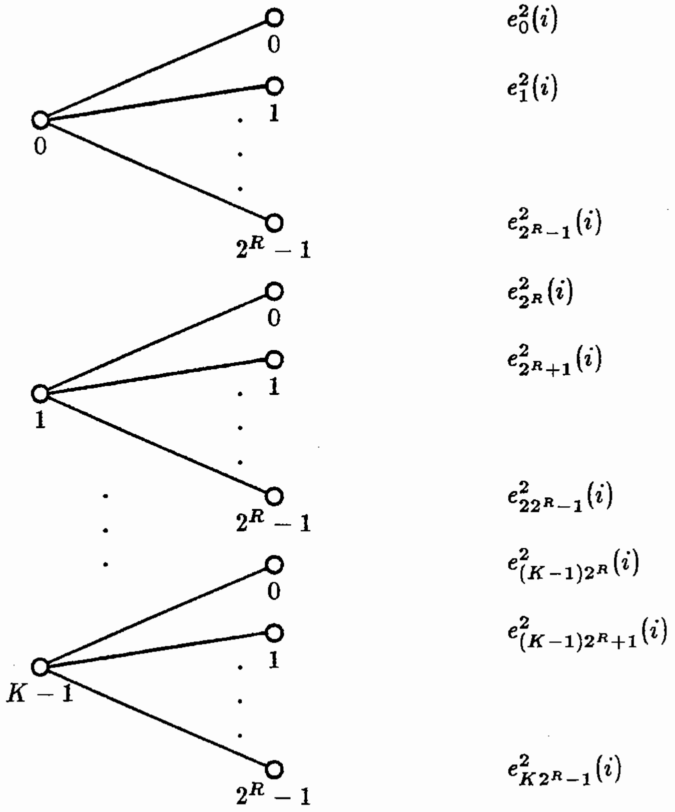


Fig. 3.6 A set of new branches populated from K nodes generated at time $i-1$

It is indeed possible to address a particular codeword using a single branch number without knowledge of the location of the node from which this branch originates. However, such a one to one mapping between a branch number and a codeword address implies that only 2^R codewords will be used by the quantizer. Each node will have identical set of codewords associated with the branches leaving it. In effect, only one set of 2^R mean-squared errors is available at time i for all K

nodes. As illustrated in Fig. 3.6, this implies

$$e_0^2(i) = e_{2^R}^2(i) = \ldots = e_{(K-1)2^R}^2(i)$$

$$\ldots \qquad \ldots \qquad \ldots$$

$$e_{2^R-1}^2(i) = e_{22^R-1}^2(i) = \ldots = e_{K2^R-1}^2(i)$$

The only factor which differentiates the paths is the set of cumulative errors obtained up to the K nodes. Because the entire set of codewords used is too structured, the advantage of multi-path tree coding is severly hampered.

It is noted that each path of the tree in the encoder and the optimal path received by the decoder are each composed of branches numbered appropriately. A parameter N_s is defined. Its value is equal to the number of successive branches on each path used in the mapping to determine a codeword address. The address of a codeword associated with a newly generated or received branch at time i is defined to be

$$A = b_{i-(N_s-1)}(2^R)^{N_s-1} + \cdots + b_{i-2}(2^R)^2 + b_{i-1}(2^R)^1 + b_i \tag{3.5}$$

where $b_{i-l} \in [0, 2^R - 1]$ for $l = 0, 1, ..., N_s - 1$ is the branch number of the branch at time $i - l$ forming the path leading to that particular branch at time i [†]. Provided the branch numbers are known, a codeword address A can be determined for both encoding and decoding purposes. According to Eq. (3.5), the total number of codewords used out of the codebook can be as many as $(2^R)^{N_s}$. All the codewords considered at time i as well as all $e_j^2(i)$'s can be different from each other if the value of N_s is large[‡].

[†] A codebook is assumed to have codeword addresses starting from 0.

[‡] It is still true that same set of codewords will be used for two nodes in the quantization of a

An additional parameter M_o is defined in the simulation. The motivation for defining this parameter is that the codebook size increases exponentially with N_s. Sometimes the total number of codewords obtained is much less or much more than the desired codebook size. In some case when the branching factor 2^R is large, say 1024, the value of A is so large as to cause overflow problem in arithmetic representation even though the value of N_s is small. For example, this problem happens when the block size N is 40 and the desired encoding bit rate R/N is 1/4 bit per sample. The parameter M_o is used to adjust the effective range of codeword addresses. A better control of the codebook size is therefore possible. Suppose A' is the final codeword address corresponding to a sequence of branch numbers, then

$$A' = \mathrm{mod}(A, M_o) \tag{3.6}$$

This equation is somewhat arbitrarily defined to compress the real number of codewords used. The value of M_o is assumed to be constant and defined when the codebook is generated. Both the encoder and the decoder are assumed to have knowledge of its value. No additional load is added onto the transmission channel. So if the value of A is less than that of M_o, $A' = A$. The codeword address A calculated using Eq. (3.5) is used. If $A = M_o$, $A' = 0$. If the value of M_o is less than or equal to the designed codebook size, only the first M_o codewords in the codebook will be used to approximate input blocks of residual signals. If the value of M_o is larger than or equal to the given codebook size, then all potential codewords can be used.

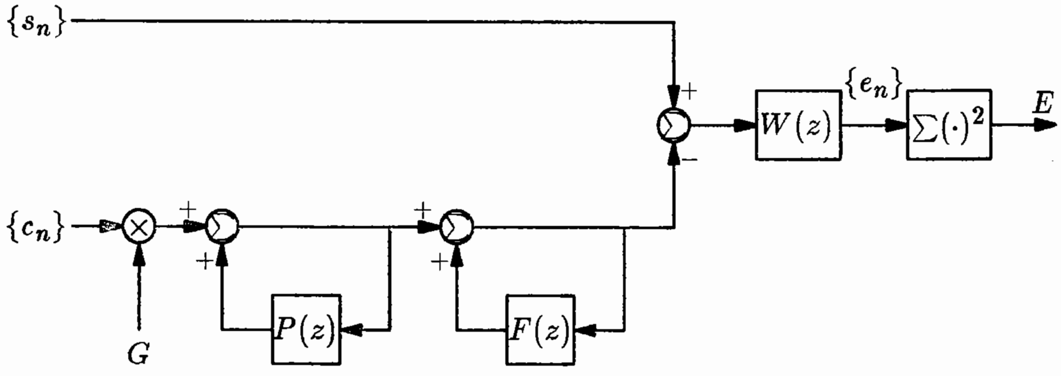
block of prediction residual samples if the value of N_s is small. For example, if $N_s = 2$, $L = 3$, and two paths obtained from the last stage of quantization are $(1,1)$ and $(2,1)$, two identical sets of codewords will be addressed.

Chapter 4 Reoptimization of the Synthesizer

Reoptimization of the synthesis filters in the receiver is useful in order to further improve the performance of the system. After an excitation waveform has been selected, a set of jointly optimized predictor coefficients can be computed to improve the reproduced speech quality. Since the codewords are fixed and the predictor coefficients are reoptimized, a time varying gain factor is required to adjust the amplitude of the noise sample sequence so as to better match the corresponding blocks of prediction residual signals.

4.1 Design of Reoptimizer

The design of the reoptimizer desired is based on the block diagram shown in Fig. 4.1 where S stands for the constant number of samples in a subframe. Given a subframe of system input speech samples and the corresponding codeword sequences chosen by the quantizer, the reoptimizer first scales the codeword sequences with the constant gain factor and then synthesizes output speech samples based on these scaled codeword sequences. The synthesized output is then compared to the subframe of input speech. Assume that there are no transmission errors, the difference signal is just the reconstructed quantization noise generated in the receiver. This noise is filtered by the weighting filter $W(z)$ designed in Chapter 3 to actually shape the noise spectrum to reduce perceptual effect of quantization noise.



$\{s_n\}$ = A subframe of input speech samples $1 \leq n \leq S = 40$

$\{c_n\}$ = A subframe of codewords selected by the quantizer $1 \leq n \leq S = 40$

$\{e_n\}$ = A subframe of spectrally weighted synthesized noise $1 \leq n \leq S = 40$

G = Gain factor

Fig. 4.1 Operations assumed in reoptimization of predictor coefficients

The lower branch of the diagram in Fig. 4.1 consists mainly of two closed loops. They are the pitch synthesizer and the formant synthesizer respectively. Theoretically, there are two sets of predictor coefficients to be reoptimized. These predictor coefficients are the β_j's, $j = -1, 0, 1$, for the pitch synthesizer and the a_k's, $1 \leq k \leq h$, for the formant synthesizer. Because $F(z) = \sum_{k=1}^{h} a_k z^{-k}$, the formant synthesizer requires knowledge of short time delayed samples from the past. However, these samples are not known before the a_k's are determined. Moreover, the weighting filter is a function of the formant synthesizer. These interlocked factors give rise to a set of nonlinear equations in the formulation of reoptimization. Therefore reoptimization of the formant predictor coefficients is not attempted.

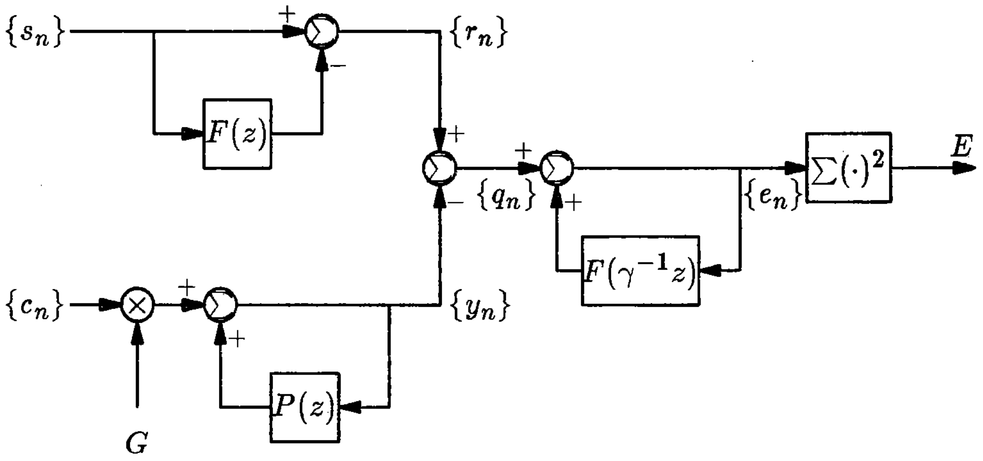
On the other hand, the pitch synthesizer has $P(z) = \sum_{j=-1}^{1} \beta_j z^{-(p+j)}$ where p is between 42 and 120 for an 8 kHz sampling frequency [†]. Only long time delayed samples from the past are required to determine the current pitch synthesizer output samples. Since the subframe size S is 40 samples long, all past samples required by the pitch synthesizer in this block diagram are outside the current subframe of signals being processed [‡]. Linear equations can be set up to determine the optimal β_j's and new pitch synthesis filter outputs for that subframe. Accordingly, the reoptimization process in the system solves linear equation systems to find gain factors and new sets of pitch predictor coefficients to optimize fine structures of noise spectra only.

With the formant predictor coefficients given for the current subframe of input and codeword sequences, the schematic diagram in Fig. 4.1 is equivalent to that in Fig. 4.2. The closed loop in the upper branch is just the formant predictor filter used in the prefilter. $\{r_n\}$ is the spectrally flattened system input speech which has already been computed in the prefilter and is assumed to be given in the design. In Fig. 4.2, the weighting filter becomes a formant synthesizer with the coefficients weighted by powers of γ. Also the figure shows that

$$e_n = q_n + \sum_{k=1}^{h} a_k \gamma^k e_{n-k} \qquad 1 \le n \le S \le 40 \tag{4.1}$$

[†] Fundamental frequency for voiced speech is assumed to be between 65 Hz and 200 Hz.

[‡] If the maximum subframe size is 20 samples long, the value of p can be between 22 and 120. The corresponding pitch frequency is between 65 Hz and 360 Hz.



$\{r_n\}$ = A subframe of formant predicted input signals $1 \le n \le S = 40$

$\{y_n\}$ = A subframe of pitch synthesized excitation signals $1 \le n \le S = 40$

Fig. 4.2 Equivalent diagram of Fig. 4.1. It is used in the computation of reoptimized coefficients.

Short time delayed samples of e_n are required in Eq. (4.1). However if $E(z)$ and $Q(z)$ are $z-$transforms of e_n and q_n respectively, then Eq. (4.1) can be written as

$$E(z) = \frac{Q(z)}{1 - \sum_{k=1}^{h} a_k(\gamma z^{-1})^k}$$

Define

$$\begin{aligned} H(z) &= \frac{1}{1 - \sum_{k=1}^{h} a_k(\gamma z^{-1})^k} \\ &= \sum_{k=0}^{\infty} h_k z^{-k} \end{aligned}$$

where h_k's are functions of a_k's and γ. Then

$$E(z) = H(z)Q(z)$$

For a stable filter $h_k \to 0$ as $k \to \infty$.

$$\begin{aligned} e_n &= \sum_{k=0}^{\infty} h_k q_{n-k} \qquad 1 \le n \le S \le 40 \\ &\approx \sum_{k=0}^{\tau} h_k q_{n-k} \qquad \tau \le S \end{aligned} \tag{4.2}$$

The approximation is especially true since h_k is linearly proportional to γ^k and $\gamma < 1$. In the sequel, τ will be set equal to S to obtain the best approximation. Also from Fig. 4.2

$$y_n = Gc_n + \sum_{j=-1}^{1} \beta_j y_{n-(p+j)}$$

$$q_n = r_n - y_n$$

Define

$$w_n \equiv \sum_{k=0}^{S} h_k r_{n-k}$$

$$v_n \equiv \sum_{k=0}^{S} h_k c_{n-k}$$

$$u_{nj} \equiv \sum_{k=0}^{S} h_k y_{n-k-(p+j)}$$

$$E \equiv \sum_{n=1}^{S} e_n^2$$

then

$$E = \sum_{n=1}^{S} (w_n - Gv_n - \sum_{j=-1}^{1} \beta_j u_{nj})^2 \tag{4.3}$$

After differentiating the energy of quantization noise E with respect to the β_j's and G and setting them to zeros, a set of linear equations is obtained as follows:

$$\sum_{j=-1}^{1} \beta_j (\sum_{n=1}^{S} u_{nj} u_{nk}) + G \sum_{n=1}^{S} v_n u_{nk} = \sum_{n=1}^{S} w_n u_{nk} \quad \text{for } k = -1, 0, 1$$

and

$$\sum_{j=-1}^{1} \beta_j (\sum_{n=1}^{S} v_n u_{nj}) + G \sum_{n=1}^{S} v_n^2 = \sum_{n=1}^{S} w_n v_n$$

Further define

$$\phi(j,k) \equiv \sum_{n=1}^{S} u_{nj}u_{nk}$$
$$\theta(k) \equiv \sum_{n=1}^{S} v_n u_{nk}$$
$$\xi(k) \equiv \sum_{n=1}^{S} w_n u_{nk}$$
$$\chi \equiv \sum_{n=1}^{S} v_n^2$$

and

$$\eta \equiv \sum_{n=1}^{S} w_n v_n$$

This set of linear equations can be written in matrix form as

$$\begin{bmatrix} \phi(-1,-1) & \phi(0,-1) & \phi(1,-1) & \theta(-1) \\ \phi(-1,0) & \phi(0,0) & \phi(1,0) & \theta(0) \\ \phi(-1,1) & \phi(0,1) & \phi(1,1) & \theta(1) \\ \theta(-1) & \theta(0) & \theta(1) & \chi \end{bmatrix} \begin{bmatrix} \beta_{-1} \\ \beta_0 \\ \beta_1 \\ G \end{bmatrix} = \begin{bmatrix} \xi(-1) \\ \xi(0) \\ \xi(1) \\ \eta \end{bmatrix} \tag{4.4}$$

Since

$$\phi(j,k) = \sum_{n=1}^{S} u_{nj}u_{nk}$$

Hence

$$\phi(j,k) = \phi(k,j)$$

The square matrix on the left hand side of the matrix Eq. (4.4) is symmetric. This matrix is positive definite unless the amplitudes of all the system input samples are zero. The matrix equation can be efficiently solved for the β_j's and G using Cholesky decomposition method. In case of a non-positive definite matrix, the reoptimized coefficients and gain factor are set equal to the original coefficients and normalization factor of the input residual signal, namely the standard deviation of the corresponding subframe of prediction residual (see section 4 of Chapter 5 for more details of normalization of prediction residual signals.)

This system reoptimizes the coefficients and gain factors in the transmitter where the codewords and input residual signals are known. The reoptimized coefficients are sent to the receiver to realize the pitch synthesis filter. The pitch synthesis filter with reoptimized tap coefficients is no longer an inverse of the pitch predictor filter in the prefilter. However, the original tap coefficients are not computed with the knowledge of quantization noise and the method used does not guarantee stability of the synthesis filter. If the synthesis filter is unstable, coding noise can be accentuated. Although the reoptimized pitch synthesizer is not guaranteed to be stable either, noise buildup is reduced with the tap coefficients reoptimized by minimizing the output noise. Reoptimization contributes an additional distortion to reconstructed signals. In the absence of quantization errors, the original set of coefficients is better. Nevertheless, Eq. (4.3) reveals that E is a convex function of G and β_j's. The total reconstructed noise including quantization errors never exceeds the reconstructed noise when the original set of coefficients is used. All parameters are updated at a rate equal to the original update rate of the pitch predictor coefficients. System performance is improved without an increase in total transmission bit rate. An additional computational load is added to the encoding process, however.

It is important to realize that this design is based on the assumption that the subframe size is less than or equal to 40 samples long and maximum pitch frequency is 200 Hz. Also h_k is assumed to approach zero as k goes to infinity. Without these assumptions, the linear equation system cannot be established. Moreover, the design is for a 3 tap pitch predictor and synthesis filter. Generalization to other cases is straightforward.

Chapter 5 Computer Simulation

The structure of the predictive coding system and the theories behind the design of the system have been studied in previous chapters. The speech coding system with a transmission bit rate around 4.8 kb/s —2.0 kb/s for the encoded prediction residual and around 2.8 kb/s for the side information—was simulated on a Vax 8600 computer in order to evaluate its performance. The programs were written in Fortran. All variables and arrays were floating point numbers. The beginning of this chapter will cover initializations of the system, simulation inputs, prefilter and its output signals, and synchronization of encoded residual information and side information. In later sections, the addition of a pre-emphasis and a de-emphasis filter to the system will be discussed. Other aspects of the system including computational complexity, and memory space requirements will be covered. According to Nakatsui and Mermelstein [19], the segmental signal-to-noise ratio (segSNR), which is defined as the averaged value of signal-to-noise ratios (SNR) measured in dB, is a good indicator of overall performance of adaptive coders. Therefore, segSNR for 16 ms intervals is used to show the objective performance of this system. Informal subjective listening tests were performed in sound proof conditions with the help of waveform and spectrogram displays. Because loudspeakers have better response to low frequency components in the signals and headsets can be used to discriminate high frequency components, they were both used in the subjective tests. Simulation results will be given at appropriate locations in the sequel.

5.1 Initialization of The System

A few variables have to be initialized to start the encoder. These include delayed input samples for the predictor filters, delayed samples for the synthesis filters in the quantizer, reoptimizer, and receiver. The initial multi-path tree and a set of cumulative errors on each node of the tree also have to be initialized.

The delayed input samples for the predictor filters and delayed output samples for the synthesis filters are set to zeros for simplicity. An initial multi-path tree with M nodes and depth L are to be set up. A simple choice is shown in Fig. 5.1. The initial set of M cumulative errors on each path of the imaginary tree is set to zero. Experiments on encoding a speech sentence show that different initial trees lead to different optimal paths and therefore different objective performance. However, the segSNR differences are less than 0.5 dB for the whole utterance. Subjectively, the reconstructed signals differing by such a small margin of objective performance have the same quality. The value of the γ factor for noise weighting filters is set to 0.75 such that the desired short time spectrum of noise will be masked by the short time power spectrum of system input signals [12][17].

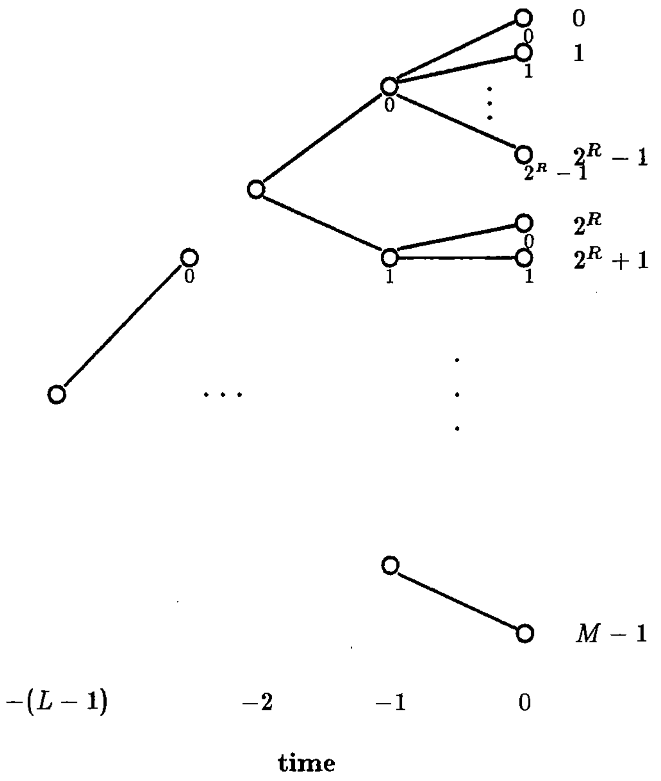


Fig. 5.1 A multi-path tree for starting the encoder

5.2 Simulation Inputs

Two high quality speech sentences were used as inputs to the system to test its objective and subjective performance. These two sentences are:

1. Cats and dogs each hate the other.

2. It's easy to tell the depth of a well.

The first sentence was spoken by a female speaker and the second sentence by a male speaker. The maximum pitch frequency was 220 Hz. These two sentences will be referred to as CATF8 and WELLM8 respectively. The input speech signals were low pass filtered at a frequency below 4 kHz and then sampled with an 8 kHz sampling frequency. Each sample is represented by a 15 bits value.

5.3 Prefilters

The prefilter stage consists of a 12^{th} order adaptive linear formant predictor filter and a 3 tap pitch predictor filter cascaded in this order. The formant predictor coefficients are computed using the autocorrelation method and transmitted without quantization to the receiving end. Blocks of 200 input samples are windowed using a Hamming window. The formant predictor filter is used to eliminate near sample redundancies for frames of 80 samples which are centered with respect to the corresponding windows. New sets of coefficients are computed for successive frames of 80 input samples (corresponding to 10 milliseconds (ms)). Because the window length is longer than the frame size, successive windows of samples for predictor coefficients analyses overlap.

The 3 tap pitch predictor coefficients are calculated using a covariance approach [12]. These coefficients are also transmitted to the receiver unquantized. For each subframe of 40 samples inside a frame, a set of 3 optimal coefficients are obtained. This corresponds to an update rate of 5 ms. The pitch period for a subframe is selected as the delay shift which gives the maximum normalized autocorrelation value. The resulted pitch period is limited to the range $[42, 121]$ in samples. For unvoiced speech portions, the signal consists of unpredictable noise. Resulting pitch predictor coefficients are small and pitch prediction had no effect on unvoiced speech signals [20].

Each subframe of prediction residual samples is further sub-divided into integral number of blocks of N samples. However, only a subset of factors of 40 is allowable if a specific encoding bit rate is desired. For example, if the branching factor of the tree 2^R is 1024 and the encoding bit rate R/N is 1/4 bit/sample (the encoding bit rate was 2 kb/s), each coding block has to contain $N = 40$ samples.

5.4 Normalization of Prediction Residual

According to a study by Schroeder and Atal [18], normalized residual signals of the predictor filters have approximately a Gaussian distribution. In the simulation, the mean value of the residual signals is small compared to the standard deviation and is assumed to be zero. The sequences for the codebook are generated using a normalized Gaussian random number generator. The long term variance of this signal is unity. One approach is to scale each codeword sample by a gain factor to match variance of the codeword sequences to that of prediction residual. The computation requirements are large with a large codebook. Instead, each sample in a subframe is normalized by the standard deviation of that subframe to get a new subframe of residual signals with zero mean and unit variance. For each new subframe of prediction residual samples, delayed signals for the synthesis filters used for codeword selection are scaled by the standard deviation of the previous subframe and renormalized with respect to the standard deviation of the current subframe to prevent boundary problems between the two subframes. Because the inverse predictor filters and the spectral weighting filter on each branch of the tree are linear, the mean-squared value $e_j^2(i)$ (see Eq. (3.4)) of spectrally weighted output noise is normalized by the variance of the subframe in which the i^{th} block of prediction residual samples locates. A time domain weighting factor equal to the variance of the subframe is required to cancel this normalization factor. Optimal quantizer decisions are then based on unnormalized cumulative mean-squared errors.

5.5 Quantization of Side Information and Bit Rate Allocation

Time varying side information required by the receiver for signal reconstruction includes the formant predictor coefficients for each frame of 80 samples, the pitch predictor coefficients for each subframe of 40 samples, and the pitch frequency and gain factor of the same subframe. In a real digital communication system, this side information has to be quantized and transmitted through the digital channel to the receiver. This contributes to additional quantization errors to the reconstructed signals at the receiver. Because this thesis is concerned with the study of a predictive coding system using a specially designed quantizer for the residual signals, quantization errors due to other quantizers are not considered. The main concern is with the performance of the system when using a tree encoder. In the simulation of the predictive coding system all time varying side information mentioned is therefore sent to the receiver directly without quantization. However, it is assumed that the maximum transmission bit rate for side information is around 2.8 kb/s. Since transmission bit rate for quantized prediction residual is set to 2.0 kb/s, the total bit rate is around 4.8 kb/s with an 8 kHz sampling frequency.

5.6 Information Synchronization

In Chapter 3, it was shown that the artificial delay in the $(M, L)-$algorithm was always $(L-1)$ blocks of N samples. The optimal decision for the i^{th} quantizer input block is made $(L-1)$ blocks later after the $(i+L-1)^{st}$ block has been processed. Side information for the i^{th} input block of prediction residual samples is not used until the $(i+L-1)^{st}$ block is considered. Memory buffers are required to delay the transmission of side information so that these two kinds of information can be synchronized.

If each frame of side information, which includes side information of each subframe, requires one buffer of a constant size, the total number of buffers necessitated, which is denoted by M_b, can be obtained as a function of L and the number of blocks in a frame.

L (blocks)	Delay (blocks)	M_b (buffers)
2	1	1
3	2	1
...	...	...
F_b+1	F_b	1
F_b+2	F_b+1	2
...	...	...
$2F_b+1$	$2F_b$	2
...	...	...

Table 5.1 Number of additional buffers required to synchronize operations of the system

Suppose L is two and the total number of blocks in a frame is F_b. There is a delay of one block in this case. The optimal index for the last block of a frame is determined only after the first block of the next frame is considered. One additional buffer is required for the side information of this frame while the second frame is being processed. Consider the case in which the value of L is equal to $F_b + 1$. The optimal decision on the index for the last block of a frame is made after the last block of the next frame is processed. The number of additional buffer required for one frame of side information is still one. When L increases by one further from $F_b + 1$ to $F_b + 2$, the delay is $F_b + 1$ blocks long. According to what was observed above for $L = F_b + 1$, the optimal index for the last block of a frame is determined after the first block of the second to the next frame is processed. Two additional buffers are necessary in this case. One is for the frame where the optimal indices of its blocks are being determined and the other is for the next frame whose input blocks of prediction residual samples have been processed but not completely quantized. The data obtained from these analyses are listed in Table 5.1. Mathematically, M_b, L and delay can be related by

$$\text{Delay} = L - 1$$

$$M_b = \left\lfloor \frac{\text{Delay} - 1}{F_b} \right\rfloor + 1$$

where the division is an integer division in which the remainder is ignored.

When $L = 1$, the optimal index for an input block of prediction residual samples is determined right after the block is processed without delay. Side information and optimal indices of a frame of residual samples can be sent to the receiver before analysis and quantization on the next frame are performed. No additional buffer besides the one holding the current side information is required.

In the previous discussions, a buffer is assumed to hold all the side information of a frame. This information includes side information for different subframes generated at different times. In fact, subframes of side information within a frame do not have to be used simultaneously. If the decoder reconstructs speech signals whenever side information for a subframe is available without waiting for the arrival of side information for other subframes, then the amount of subframe information delayed in the encoder can be reduced. Less memory spaces are required by the encoder to synchronize the information. However, more complicated control is required and the saving diminishes with L. In the simulation, the encoder holds entire frames of side information. A frame of side information is released to the decoder after all indices for the frame of unpredictable signals are determined. In the receiver, the excitation sequences are pitch synthesized for every subframe. The entire frame of pitch synthesized samples is then formant synthesized.

5.7 Pre-emphasis and De-emphasis Filters

Although the formant synthesizer analyzed using autocorrelation method is theoretically stable, instability may occur due to round off errors in real implementation. Studies by Gray showed that such a problem could be reduced if the system input signals were first pre-emphasized to flatten the spectrum [13]. A pre-emphasis filter is installed in the encoder just before the time varying linear formant predictor filter to study the effects of pre-emphasis and de-emphasis on system performance. The formant predictor coefficients are computed based on pre-emphasized signals

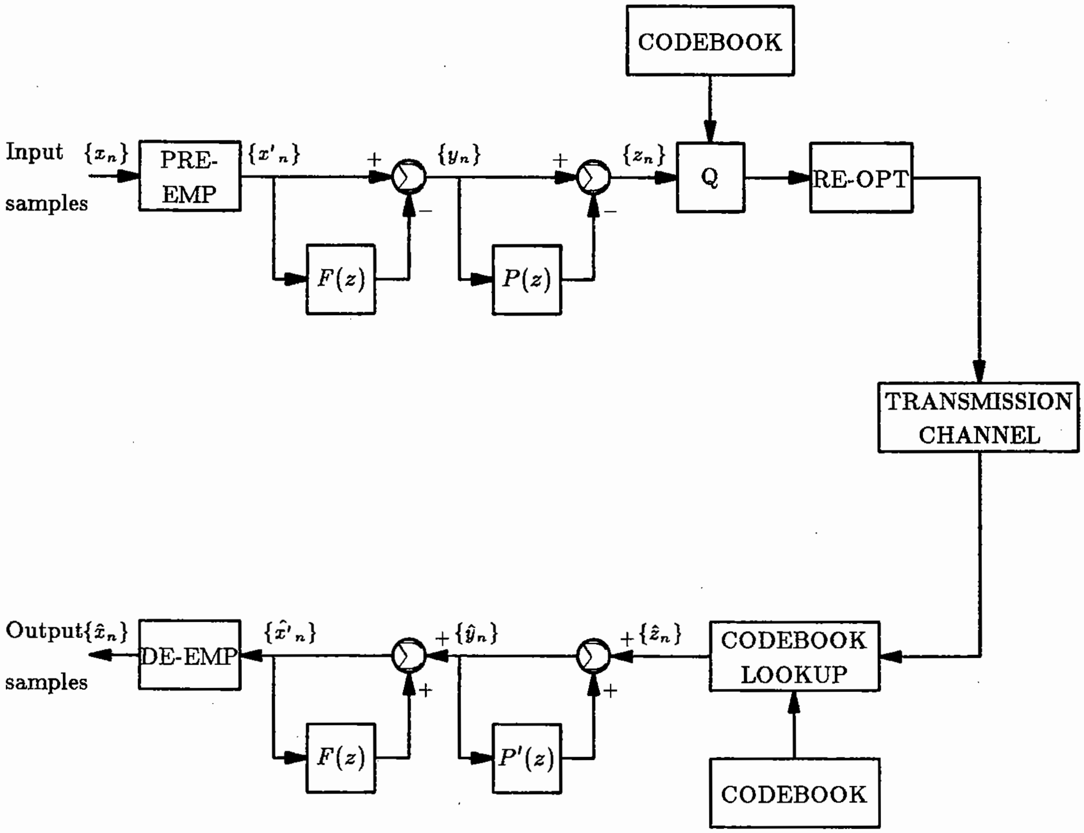


Fig. 5.2 Complete block diagram of the speech coding system with pre-emphasis and de-emphasis filters

and used to predict pre-emphasized speech samples. The synthesized output of the inverse formant predictor filter in the decoder is the reconstructed version of the pre-emphasized signals. A de-emphasis filter is put at the decoder after the inverse formant predictor filter to obtain the reconstructed signal of the original speech input as in Fig. 5.2. The $z-$transforms of the pre-emphasis filter and the de-emphasis filter are respectively as follows:

$$1 - \mu_p z^{-1}$$

and

$$\frac{1}{1 - \mu_d z^{-1}}$$

where $\mu_p = \mu_d = \mu$, $0 \leq \mu < 1$ are tap coefficients of the pre-emphasis and de-emphasis filters. Because the quantizer was designed to perform on the basis of analysis-by-synthesis technique, a de-emphasis filter with the same z – transform is put after the weighting filter on every branch of the tree. The performance of the complete coding system is then evaluated as a function of the coefficient μ.

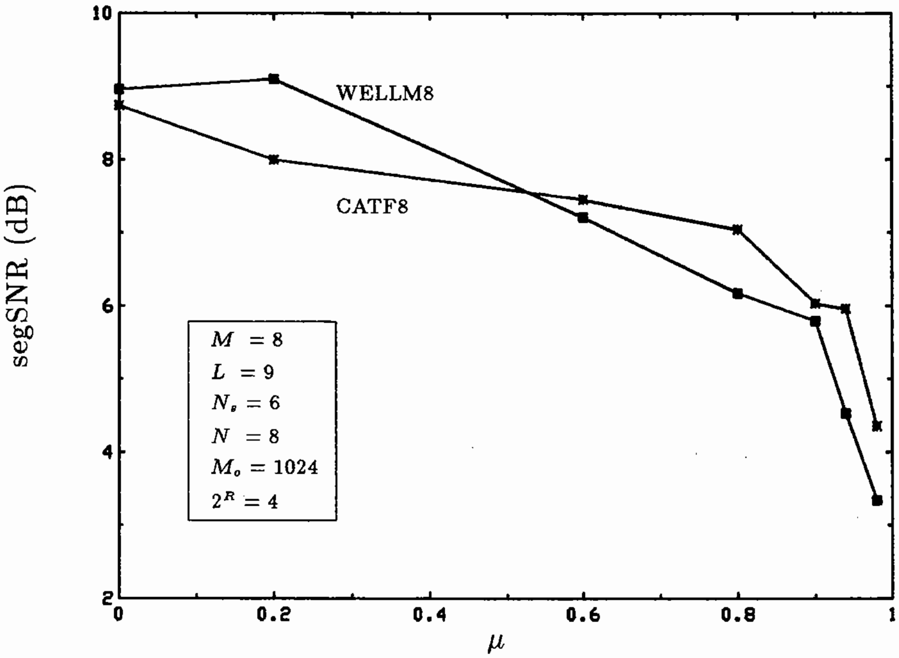


Fig. 5.3 Performance of the system as a function of pre-emphasis filter coefficient μ

As shown in Fig. 5.3, the segSNR value of the system drops with μ for both system input speech signals. Subjectively, low frequency booming can be detected over loudspeakers in the reconstructed speech signals. Spectral analyses show that very low frequency components are generated in reproduced speech signals if the value of μ is large. Normalized autocorrelation functions of the prediction residual are computed. Almost identical normalized autocorrelation functions for all values of μ indicate that the optimized predictor filters whiten system input speech signals to the same degree. Distortions are therefore caused by de-emphasis filters in the tree quantizer and the decoder.

Recall that the spectral weighting filter $W(z)$ was designed to flatten a perceptually optimal short time noise power spectrum (see Fig. 3.4). Quantization noise which originally has flat power spectrum is suppressed at the formant regions. The suppression is especially obvious at the first formant. The effect of low frequency amplification due to the de-emphasis filter tends to flatten the spectrum of the latter noise. Minimization of accumulated mean-squared error chooses the codeword associated with this perceptually poor noise. Accordingly, the reconstructed noise very likely contains components which mask the short time signal power spectra at interformant regions (see Fig. 3.4.) Moreover, there is spectral weighting in the decoder due to the de-emphasis filter. As the value of μ approaches 1, the equivalent low pass filter has higher and higher low frequency gain, and more and more low frequency noise is amplified. This effect has been independently verified by running the same system without de-emphasis filter inside the tree quantizer. Therefore, it is concluded that pre-emphasis and de-emphasis filter should not be used in the coding system. Instability problem in the autocorrelation formulation can be avoided if computation is done with sufficient precision.

5.8 Limitations to the Quantizer Design

Discussions in previous chapters indicate that the encoder of the system, which includes a linear formant predictor filter, a linear pitch predictor filter and a delayed decision multi-path tree quantizer, is more complex in structure than the decoder. Strictly speaking, most of the complexity of the encoder is due to the tree quantizer. The predictor filters and the synthesizers are inverses of each other. Operations on input signals in the predictor filters and synthesizers are straightforward and almost instantaneous. The only delay arises from the computation of the predictor coefficients. The number of arithmetic operations and size of the delays caused by these prefilters and postfilters are negligible when compared with those of the quantizer.

In the following sections, computational complexity of the delayed decision multi-path tree quantizer, its memory and delay considerations will be analyzed. These analyses are performed for the worst case. The number of paths kept in each stage is assumed to be M, and the encoding bit rate is R bits per block of N prediction residual samples. In view of the performance of the system given in last section, pre-emphasis and de-emphasis filters are not used.

5.8.1 Computational Complexity of the Quantizer

Assume as a result of the analyses in the i^{th} stage that M paths (nodes) are available. In the $(i+1)^{st}$ stage, a block of residual samples of size N enters the quantizer. In response to this, 2^R branches are populated from each node. There are $M2^R$ codewords assigned to the leaves of these $M2^R$ branches. In order to find a delayed decision, each one of these codewords is subtracted from the quantizer input block sample by sample. Assume for the time being that mean-squared errors are computed directly from the difference sequences. Since the block size is N, the total number of multiplications T_m, and additions T_a (which includes the number of subtractions) required are respectively equal to

$$T_m = M2^R N \qquad \text{per block}$$

$$T_a = M2^R(N-1) + M2^R N \quad \text{per block}$$

If computations due to $\gamma-$weighted formant synthesizer and pitch synthesizer on each branch are taken into account, these numbers have to increase. Let $\{d_n\}$, $\{\tilde{d}_n\}$ stand for the input sequence to the inverse 3 tap pitch predictor filter and $\gamma-$weighted inverse h^{th} order formant predictor filter for $1 \leq n \leq N$ respectively. Also suppose that $\{\hat{d}_n : 1 \leq n \leq N\}$ is the $\gamma-$weighted inverse formant predictor filter output sequence. The difference equation for the pitch synthesis filter is

$$\tilde{d}_n = d_n + \sum_{j=-1}^{1} \beta_j \tilde{d}_{n-(p+j)} \qquad 1 \leq n \leq N \tag{5.1}$$

where β_j, $-1 \leq j \leq 1$, are pitch predictor coefficients. For the formant synthesis filter

$$\hat{d}_n = \tilde{d}_n + \sum_{k=1}^{h} a'_k \hat{d}_{n-k} \qquad 1 \leq n \leq N \tag{5.2}$$

with $a'_k = a_k \gamma^k$, $1 \leq k \leq h$, being $\gamma-$weighted formant predictor coefficients. Equations (5.1) and (5.2) show that the filtering requires $M2^R(3+h)N$ multiplications and $M2^R(3+h)N$ additions per block of prediction residual samples.

After the $M2^R$ error energies for the $(i+1)^{th}$ quantizer input block are calculated, the mean-squared error on each leaf is added to the cumulative error of the node from which the corresponding branch is populated. An additional $M2^R$ additions are necessary. This new set of $M2^R$ cumulative errors is compared with each other until an optimal delayed decision is found. Since comparisons are essentially subtractions, the total number of arithmetic operations are

$$\begin{aligned} T_m &= M2^R N(h+4) && \text{per block} \\ T_a &= M2^R[N(h+5)+1] && \text{per block} \\ &\approx M2^R N(h+5) && \text{per block} \end{aligned}$$

If $R/N = r$ bit/sample, then

$$\begin{aligned} T_m &= M2^{rN}(h+4) \quad \text{per sample} \\ T_a &= M2^{rN}(h+5) \quad \text{per sample} \end{aligned} \tag{5.3}$$

Equation 5.3 show that the complexity is linearly proportional to M and increases exponentially with N. The value of L or the delay has no effect on computational complexity.

N	T_m	T_a
4	$32M$	$34M$
8	$64M$	$68M$
20	$512M$	$544M$
40	$16384M$	$17408M$

Table 5.2 Total numbers of arithmetic operations in the quantizer per residual sample for different block sizes N with $r = 1/4$ bit/sample and $h = 12$

The total numbers of arithmetic operations in the quantizer per residual sample for those values of N compatible with coding at $1/4$ bit/sample are tabulated in Table 5.2.

5.8.2 Quantizer Memory Consideration

The design of the quantizer requires a statistically distributed codebook and a tree of depth L and maximum number of paths M. Assume $M_o \leq 2^{RN_s}$ then M_o is equal to the number of codewords available. Because a path connecting a root node and a newly generated node consists of $L-1$ branches, memory space required to store the paths and the codebook is

$$M_o N + M(L-1)$$

Recall that delayed output samples are required for each pitch synthesizer and $\gamma-$weighted formant synthesizer on a tree branch. Although there are $M2^R$ branches available at time i, only M sets of delayed output samples are required from time $i-1$. Assume that pitch period $p \leq p_{\max}$ where $p_{\max}$ is equal to 120 in the simulation then the amount of additional memory is equal to

$$M(p_{\max} + h)$$

The resulting memory is around

$$M_o N + M(L + p_{\max} + h)$$

Recall that the codes have maximum randomness if $M_o = M2^R = M2^{rN}$. If additional buffer area required in synchronization of information is also considered, the total amount of memory space required is [†]

$$\begin{aligned} &M(N2^{rN} + L + p_{\max} + h) + M_b(h+10) && \text{for } L > 1 \\ &M(N2^{rN} + L + p_{\max} + h) && \text{for } L = 1 \end{aligned}$$

[†] In addition to the formant predictor coefficients there are 3 pitch predictor coefficients, one gain factor and one pitch frequency for each subframe.

It is important to see that a new set of at most M paths are selected after an optimal delayed decision is made. All cumulative error values and synthesizer outputs obtained during computations for the optimal decision are required to select the M new paths. This information can be saved at the expense of extra memory. However in this simulation all the information is regenerated because too much memory is required in the case of large values of N. In the worst case, synthesizer outputs on all $M2^R$ branches are required. There is a tradeoff between computation and memory. For regeneration of the information computational complexity is approximately doubled. Effective numbers of arithmetic operations per sample are[◇]

$$\begin{aligned} T_m &= 2M2^{rN}(h+4) \\ T_a &= 2M2^{rN}(h+5) + \frac{M^2}{N}(2^{rN}-1) \end{aligned} \tag{5.4}$$

for $L > 1$ or

$$\begin{aligned} T_m &= M2^{rN}(h+4) \\ T_a &= M2^{rN}(h+5) \end{aligned} \tag{5.5}$$

for $L = 1$. If the information is saved, memory is increased by

$$M2^R(p_{\max} + h)$$

and effective memory space required is approximately equal to

$$\begin{aligned} &M(2^{rN}(N + p_{\max} + h) + L + p_{\max} + h) + M_b(h+10) \quad &&\text{for } L > 1 \\ &N2^{rN} + L + p_{\max} + h &&\text{for } L = 1 \end{aligned} \tag{5.6}$$

Equations (5.4), (5.5), and (5.6) are exponential functions of N. If computational complexity is important and the information can be saved, then typical values

[◇] The second term on the right of the equal sign for T_a is due to comparisons which are performed by taking the first M results as best and then updating them as $M2^R - M$ new results are being generated. This term exists even if the information is saved and no regeneration is required.

of memory required for $L > 1$, $r = 1/4$ bit/sample, $p_{\max} = 120$ and $h = 12$ are in the order of $176260M + 22M_b$ for $N = 40$ and $M(692 + L) + 22M_b$ for $N = 8$. For a large block size N, the memory required by the tree quantizer is large.

5.9 Performance of the System

In this section, objective and subjective system performance are given as functions of M, L, N_s, M_o, and N. In view of the deleterious effect of pre-emphasis and de-emphasis filters, pre-emphasis and de-emphasis filters are turned off. Comparisons of system performance obtained with and without pitch predictor coefficient reoptimization show that both objective and subjective performance are improved when the extra operations are performed. Subjectively, high frequency background noise and other defects like clicks appear in the absence of reoptimization. Recall that the reoptimization process was postulated for speech with a maximum pitch frequency of 200 Hz. Subjective listening tests with CATF8 show that there is an improvement in the reconstructed signal fidelity even though CATF8 has a maximum pitch frequency of 220 Hz. Therefore, pitch predictor coefficient reoptimization process is employed to obtain the following results.

5.9.1 Performance as a Function of N_s

A quantizer with less structured codes performs well [8][7]. An example is the adaptive delta modulation (ADM) which has adaptive rather than fixed step size. In Chapter 3, the parameter N_s was defined as the number of successive branch numbers required to determine codeword addresses. A less structured codeword collection for the quantizer is defined with a large value of N_s. Nevertheless, there is a difference between how structureless codes are generated in an ADM encoder and in the present quantizer. In an ADM system, a step size is logically adjusted with an intention to reduce either the overload or granular quantization noise based on past information. Equation (3.5) was however defined to increase the number of codewords considered in the quantization process and randomness of the codeword collection. A codeword is chosen not based on any knowledge of how well this codeword will affect the performance. Instead, all potential codewords in the codebook are considered equivalent because they are generated by the same random noise generator. It is the structure of the existing paths which determine the codewords to be used.

The objective performance of the system as a function of N_s is shown in Fig. 5.4. The experiment was performed with a constant set of parameters M, L, M_o, block size N, and branching factor 2^R. As expected, the objective performance does increase with N_s. The performance saturates for large values of N_s. A logical explanation for this saturation rests on the branching factor 2^R and the maximum number of paths kept M. The maximum number of branches allowed on a tree at time i is known to be $M2^R$ and the maximum number of potential codewords according to Eq. (3.5) is 2^{RN_s}. It is then obvious that the codewords associated with all branches at time i are different if $2^{RN_s} \geq M2^R$. Maximum randomness is achieved in these cases. Further increase in the value of N_s will not increase the number of different candidate codewords chosen at time i. For example, satura-

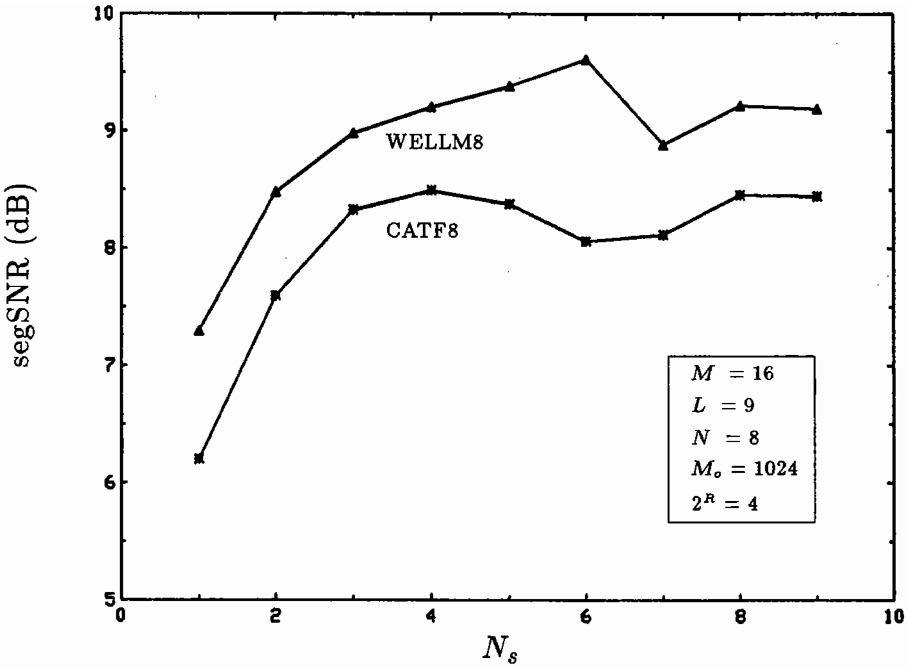


Fig. 5.4 Objective performance of the system as a function of N_s

tion occurs at $N_s = 3$ when M and 2^R are set to 16 and 4 respectively. On the other hand, the system uses different codeword combinations for each value of N_s after saturation. This suggests the reason for the fluctuations in performance after saturation.

Subjectively, each reconstructed speech sentence of WELLM8 has very high quality in terms of intelligibility and naturalness. Although the quality of the reconstructed sentences of CATF8 is also high, perceptual distortion in the forms of clicks on the words "CATS"and "HATE"can be detected. Overall performance of the reconstructed sentences are roughly as indicated by the objective segSNR measures although the measure does not quantify different types of distortion. As N_s decreases, degradation in the words "EASY"and "CATS"become more obvious.

In CATF8, high frequency background noise appears when $N_s = 1$.

Another possible explanation for the improvement in performance with N_s is that the use of a large N_s allows the present codewords to be determined based on statistics of codewords on past branches forming the paths. However, this explanation is not likely to be completely correct according to the following arguments. Firstly, the codeword associated with a branch is determined based on N_s branch numbers. They are the number of the present branch and $N_s - 1$ branch numbers of the previous $N_s - 1$ branches. In response to an input block of prediction residual samples, new sets of N_s numbers containing only part of previous sets of N_s numbers are used to determine new codes. With only incomplete information, the addresses of previous codewords cannot be found. Moreover, the address of a codeword and the statistics of the codeword are unrelated. The encoder has no knowledge of which codeword was chosen for a previous branch forming the path. Determination of a new codeword is somewhat independent of the codewords on previous branches. Secondly, all codewords are generated by a random noise generator and thus have the same statistics. The use of N_s branch numbers (see Eq. (3.5)) to determine the codewords to be used is somewhat arbitrary. The encoder does not try to interpret and exploit the statistics of codewords on each existing branch to result in a set of good codewords. The conclusion of these arguments points that the increase in performance is due to an increase in the number of codeword choices or in the degree of randomness. Past codeword statistics plays no role in improving the system performance.

It is interesting to note from Fig. 5.5 that the objective performance of the

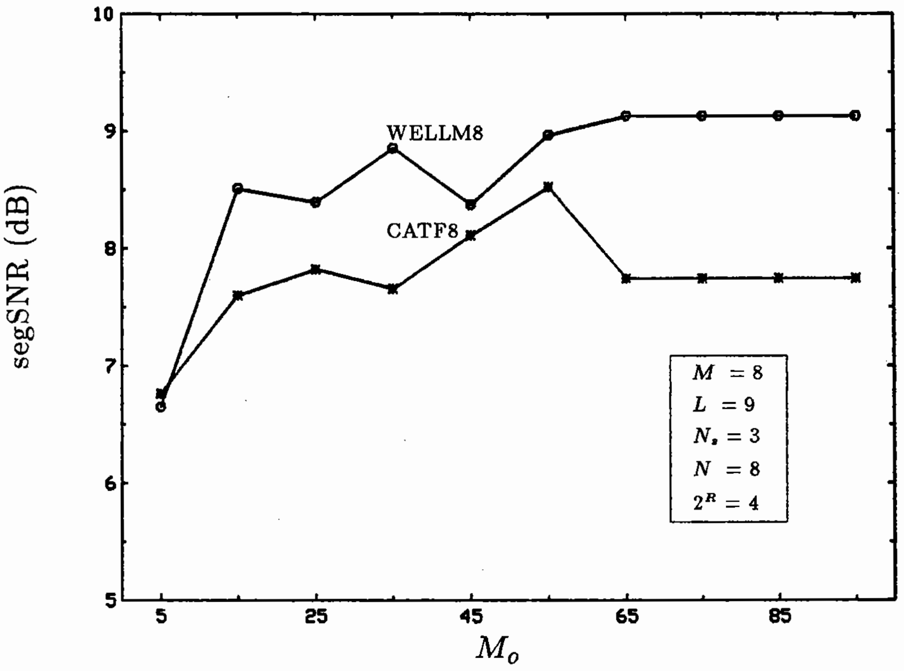


Fig. 5.5 Objective performance of the system as a function of M_o

system stays around a constant level for $M_o \geq M2^R = 32$, which is equal to the maximum number of branches considered in each level. Fluctuations in performance occur as the value of M_o increases below the maximum value of A in Eq. (3.5)—2^{RN_s}=64—and new codewords are introduced into the tree. This suggests that the codewords are not optimal. Some codewords are better than the others. As long as the value of M_o or the effective codebook size is not too small compared to the maximum number of branches available, the system behaves consistently and satisfactorily. Reasonable irregularity and randomness are guaranteed when the number of codewords in the codebook is sufficient. As the effective codebook size becomes too small, the codebook receives insufficient irregularity and the performance of the quantizer degrades. High frequency background noise can be detected as in the case with small N_s if M_o is small. This noise is thought to be equivalent to granular noise generated in a PCM system when a coarse quantizer is in use. Proper combinations of N_s and M_o give the system good performance without using too many codewords.

5.9.2 Performance as a Function of M

It is recalled that the parameter M is the maximum number of paths kept or the maximum number of nodes generated after the processing of an input block of residual samples in determining the optimal approximation of a delayed block of prediction residual samples. In a delayed decision multi-path tree quantizer implemented by the $(M, L)-$algorithm, the actual number of paths kept is upper bounded by M whose effective value is in turn upper bounded by $2^{R(L-1)}$. As the value of M increases by one under the upper bound, one more path is kept ,and one more node is available in the tree for the next quantizer input block. It is clear that 2^R more codewords will be considered in response to a new input block.

If the values of N_s and M_o are large enough, this additional set of 2^R codewords will be different from the codewords already selected for the same input block of residual samples. The overall performance of the system is expected to increase. This expectation is verified by the plot in Fig. 5.6 which shows the objective performance of the system as a function of M. The data were obtained with $L = 6$, $N_s = 6$, $M_o = 1024$, and $R = 2$ bits per block. The maximum value of M allowed in this example is $(2^2)^{(6-1)} = 2^{10} = 1024$. Overall subjective quality is very high with no background noise. Degradation is in the form of low volume and

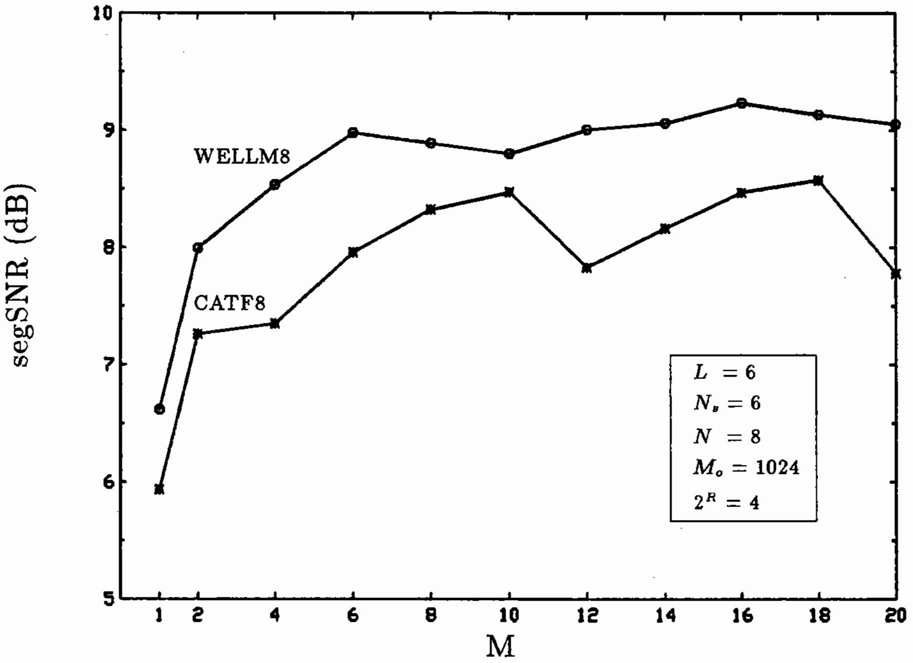


Fig. 5.6 Objective performance of the system as a function of M (The first point of each curve has $(M, L, N_s, N) = (1, 1, 6, 8)$)

less sharpness on each word. The relative perceptual quality of the reconstructed signals is as indicated by the objective measure. However, the best and the worst reconstructed signals are not very different except for the reconstructed speeches of CATF8. Clicks are detected in the reconstructed CATF8 characterized by $M = 1$ and $L = 1$. Differences less than 0.5 dB cannot be detected perceptually.

There are always codewords which do not give satisfactory error performance when used to approximate blocks of prediction residual samples. Nevertheless, they are by no means all bad if their long term effects are considered together with other codewords considered at different times. If the value of M is small, many of these bad codewords will be eliminated very soon by the quantizer so that other codewords with good short term performance can be retained. As M increases, some of those codewords with a poor short term performance will be kept and considered together with old and new codewords forming the additional paths. The increase in overall objective performance with M as shown in Fig. 5.6 confirms that such consideration of long term effects does benefit the encoding system. The use of coding delay due to L is partially justified.

The plot in Fig. 5.6 shows that the objective performance tends to saturate long before M reaches its maximum value allowed. In other words, a full search in a delayed decision encoder with a finite value of L is not worthwhile. High performance can be obtained if M is properly chosen. Saturation in system performance occurs because the value of L is finite for an instrumentable tree quantizer. Cumulative errors on the paths cannot be averaged out in too long a time span. As M increases further, more bad codeword combinations are retained for further consideration. However, their cumulative errors are too large to be acceptable when they are compared with many other cumulative errors. Their presence in the tree may occasionally alter the optimal path and thus the overall performance, but they will be rejected from further consideration eventually. The overall performance becomes almost constant with M.

5.9.3 Performance as a Function of Tree Depth L

It is understood that short term optimal decisions may not be good for the long term performance of the system when these short term decisions are combined together to reconstruct speech signals. The main effect of artificial delay due to L is that quantization error is averaged out and considered over a period of LN samples.

Another advantage of L is the feasibility of multi-path encoding for $L > 1$. The delay caused by L allows the selection of long term optimal and revocable approximations to quantizer inputs from a large set of choices.

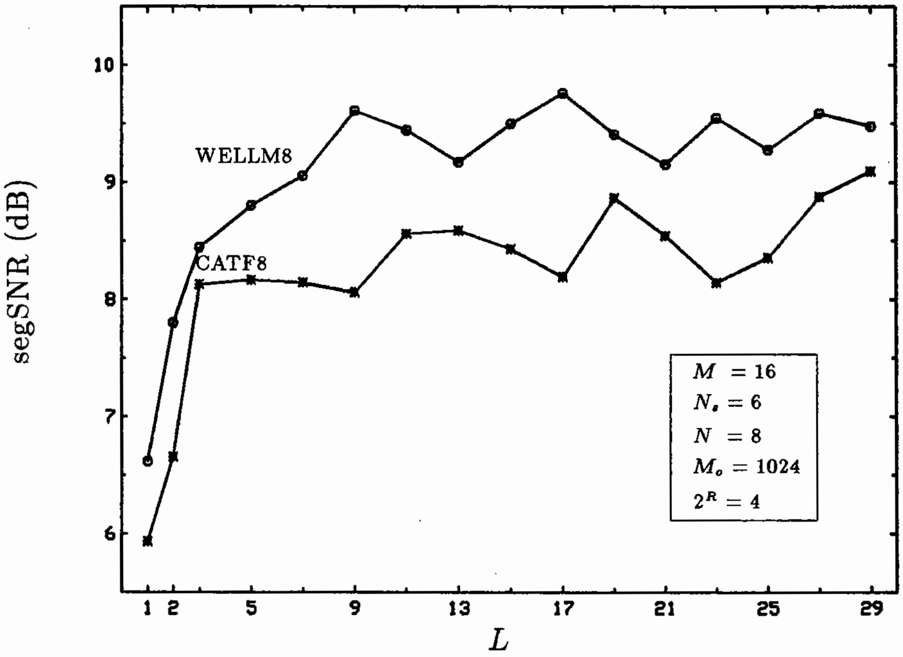


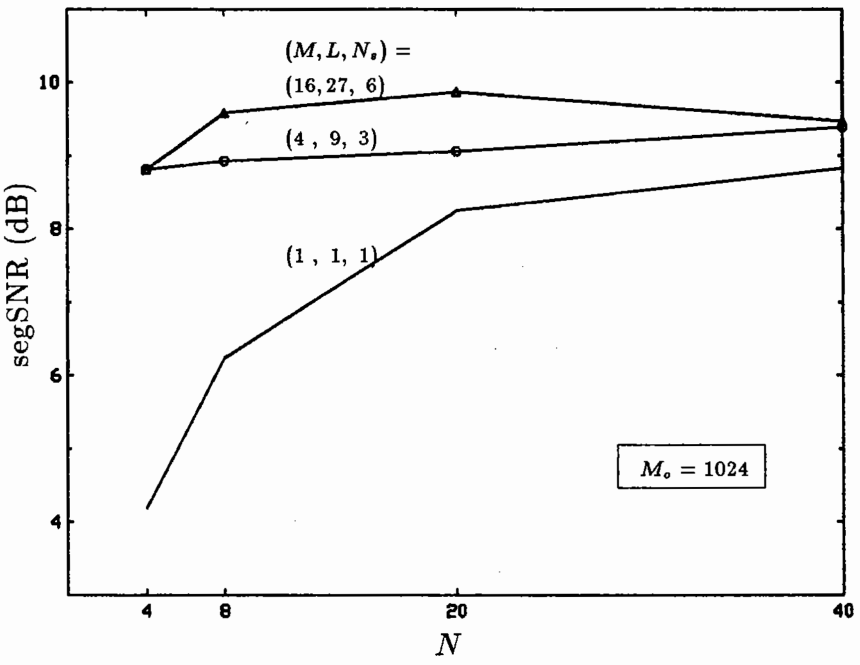
Fig. 5.7 Objective performance of the system as a function of L (The first point of each curve has $(M, L, N_s, N) = (1, 1, 6, 8)$ and the value of M is 4 when $L = 2$.)

The objective performance of the system is shown in Fig. 5.7. Although the plots are not smooth, they indicate an increase in performance with L. Subjectively, the reconstructed sentences are very natural and intelligible as those obtained in the studies of M, N_s, and M_o. Difference between adjacent reconstructed speech sentences are small. Clicks can be heard in those reconstructed sentences especially of CATF8 whose objective performance is below 7 dB. The results are obtained based on $M = 16$, $N_s = 6$, and $M_o = 1024$ except for those with $L = 1$ or $L = 2$.

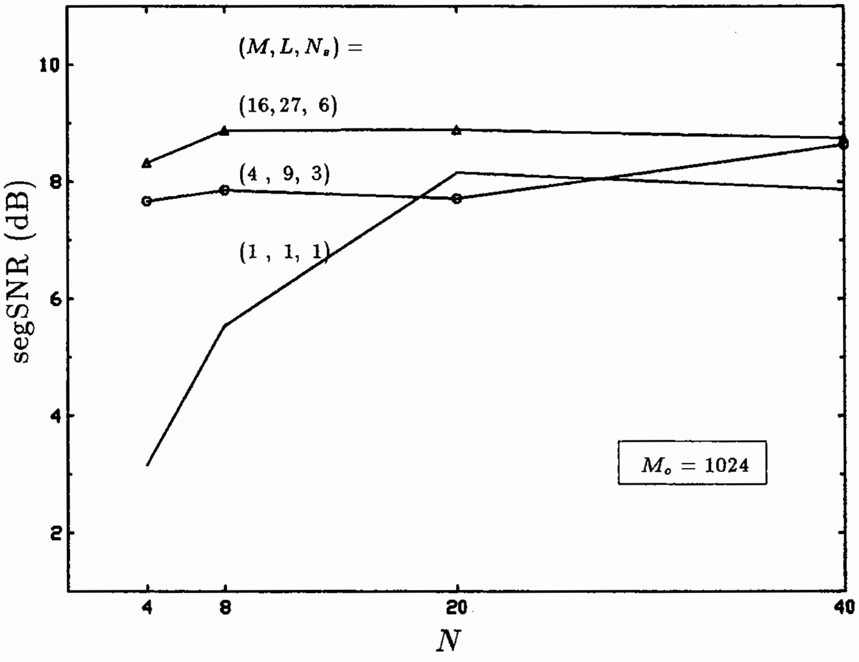
The previous plots show that there are close relations between the parameters M, L, N_s, and M_o. For a tree with a depth of L, the number of nodes available M is upper bounded by $2^{R(L-1)}$ while the values of N_s and M_o should be less than or equal to $M2^R$. Analyses in previous sections indicate that good performance is possible with large value of M in the presence of large N_s and M_o. On the other hand, a high performance can be achieved with large values of N_s and M_o which require a large value of M. Put differently, the positive effect of M, N_s and M_o are enhanced or limited by each other, while L plays a more independent role. In view of the saturation in performance and system instrumentability the values of L and thus M, N_s, and M_o should not be too large.

5.9.4 Performance as a Function of Block Size N

According to rate distortion theory, quantization noise decreases as the quantization block size increases for a fixed encoding rate. The performance of the coding system is evaluated as a function of block size N. The maximum number of codewords used is limited to 1024 for all block sizes and the transmission bit rate is 1/4 bit/sample. The objective performance of the system as a function of N is plotted in Fig. 5.8(a) and 5.8(b). Each curve is characterized by a combination of (M, L, N_s). As expected, the objective performance is increasing with quantization block size N. Curves characterized by higher values of M, L, and N_s have better segSNR measures. A dramatic increase in performance with an increase in N is found when $(M, L, N_s) = (1,\ 1,\ 1)$. When compared to the original sentences, utterances in all reconstructed signals are less loud and sharp. However, except for



(a) $N = 4, 8, 20, 40$ and input sentence is WELLM8



(b) $N = 4, 8, 20, 40$ and input sentence is CATF8

Fig. 5.8 Objective performance of the system as a function of N

(M, L, N_s, N) = (1, 1, 1, 4) and (1, 1, 1, 8) all cases are highly intelligible and natural. For the (M, L, N_s) = (16, 27, 6) and (4, 9, 3) cases, the degradation is small in general. Slightly more distortion is found on the words "CATS"and "HATE"when $N = 4$. This difference is not noticeable if a loudspeaker is used.

Subjective quality for (M, L, N_s, N) = (1, 1, 1, 4) is unacceptable. Although it is intelligible, severe distortion on each word makes the speech very unnatural. This is true especially with the reconstructed CATF8. When $N = 8$, distortion appears in the reconstructed CATF8 as a background noise instead of as a degradation on the words. The reconstructed sentences have the same quality as the other two (M, L, N_s) combinations when $N = 20$ and 40.

Although the segSNR measure does not perfectly quantify the subjective performance, it is a very good indicator of the relative perceptual quality. Figures 5.8(a) and 5.8(b) show that with small block size N, the tree quantizer outperforms a single path codebook quantizer. Analyses on M, L, and N_s show that performance of the system saturated with large values of M, L, and N_s. This is demonstrated in Fig. 5.8(a) and 5.8(b) again by the fact that the upper two plots are not too far apart. When $N = 40$, the number of codewords used is upper bounded at 1024 for (M, L, N_s) = (16, 27, 6) and (4, 9, 3). No more increase in performance is observed for these two combinations. On the other hand, the number of codewords is 32 for (M, L, N_s) = (1, 1, 1) when $N = 20$. A further increase in performance is observed in this case as N increases to 40 and the total number of codewords to 1024.

Figure 5.9 shows the objective performance of the system as a function of the tree depth L and block size N while the product of L and N, the total artificial delay in samples, is constant. For a quantization block size $N \geq 8$, the performance is approximately constant. When compared with the points at $L = 3$ or at $L = 15$ $(N = 8)$, the additional two points show that a decrease in either the block size N or the tree depth L alone causes reduction in performance. The effects of L and N on the system performance compensate each other when LN is constant. However, a decrease in performance is obvious as the block size N drops to 4 samples. The effect of the block size N is stronger than that of the tree depth L. The values of L and N should be carefully chosen after the effects of these two parameters on the system performance and complexity are considered.

According to Eq. (5.4) (5.5) and (5.6) computational complexity and memory increase exponentially with N. Therefore, it may be worth adding delay to the coding system in the form of large L and M if high performance is desired with minimal computational complexity and memory. Since coding complexity is an exponential function of N, the overall coding complexity with a small N can be smaller than that with a large N. For example, near optimal performance is obtained with $(M, L, N_s, N) = (16,\ 27,\ 6,\ 8)$ and $(1,\ 1,\ 1,\ 40)$ respectively. However, numbers tabulated in Table 5.3 show that a system using the first combination has much less coding complexity. Overall system requirements with the delayed decision multi-path tree quantizer are much simpler. The tree quantizer when characterized by such a combination is superior to the codebook quantizer used in the original CELP design.

	CASE 1		CASE 2
	Information Regenerated	Information Saved	
Computational Complexity	$T_m \approx 2^{11}$ $T_a \approx 2^{11}$	$T_m \approx 2^{10}$ $T_a \approx 2^{10}$	$T_m \approx 2^{14}$ $T_a \approx 2^{14}$
Memory	3,122	11,570	41,093

Table 5.3 Comparison of quantizer structures for two different combinations of M, L, N_s, and N CASE 1 has $(M, L, N_s, N) = (16, 27, 6, 8)$ and CASE 2 has $(M, L, N_s, N) = (1, 1, 1, 40)$

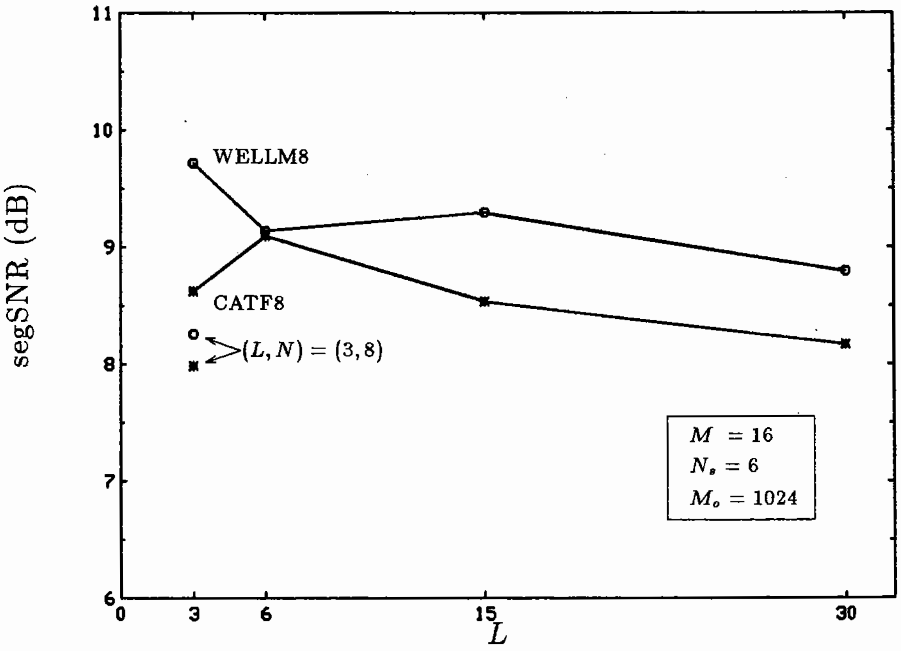


Fig. 5.9 Effects of constant total artificial delay LN in samples on the performance of the system The values of L and N change such that $LN = 120$.

5.10 Recommendations for Future Studies

Although the code excited linear predictive coding system using the delayed decision multi-path tree quantizer has been found superior to the same system studied in [3] using a codebook quantizer, the system studied in this text is by no means optimal and the simulation is not completely realistic. Further studies on the system are recommended.

Recall that the use of N_s branch numbers to determine a codeword address and the use of a modulo operator (mod M_o) to limit the size of the codebook were defined somewhat arbitrarily. System performance may be improved with a more efficient use of branch numbers to generate the codes. It was assumed in the computer simulation that the transmission channel was error free. This is never the case in practical communication systems. Moreover, side information was transmitted to the receiver unquantized. Stability of the recursive filters in the quantizer and the decoder may be affected after these two issues are taken into account. Effects of channel errors and side information quantization on the coding system should be explored.

Chapter 6 Conclusion

This thesis has studied the use of a delayed decision multi-path tree quantizer in a code excited linear predictive coding system. In Chapter 2, a code excited linear predictive coding system was described. The techniques for designing an adaptive linear 12^{th} order formant and a 3 tap pitch predictor filters and their inverses were given. Chapter 3 described the prediction residual quantizer implemented by a modified (M, L) search algorithm which assumes a delayed decision multi-path tree structure. Two additional control parameters N_s and M_o were introduced to the design of the quantizer to increase the flexibility of the coding process. Techniques to utilize knowledge of speech perception to increase the perceptual quality of the reconstructed speech were also covered in Chapter 3 and Chapter 4.

Computer simulation of the speech coding system indicates that the system performance is positively affected by the tree depth, number of multi-path, randomness of the codes, and quantization block size. Each factor plays an important role in increasing the system performance. High frequency granular noise in the reconstructed signals is apparent with codes with insufficient irregularity. Tree codes with M less than maximum (full search) perform essentially as well as full search schemes which have $M = (2^R)^{L-1}$. Experimental results also show that de-emphasis filters introduce an undesired spectral weighting effect to quantization noise generated in the system. Pre-emphasis and de-emphasis filters should not be used in the speech coding system studied. The objective performance segSNR is less than 11 dB. This is mainly because the encoded residual signal has a very low bit rate (1/4 bit per sample) and the maximum number of candidate codewords are upper bounded by M_o. The best reconstructed signals, however, have very high fidelity in terms of intelligibility and naturalness.

The structure of a delayed decision multi-path tree quantizer is in general more complicated than a codebook quantizer. The memory and computational requirements are exponential functions of block sizes. However, the delayed decision multi-path tree quantizer studied is more flexible. With proper combinations of the control parameters M, L, N_s, M_o, and N, better performance with less computational complexity, memory and delay requirements can be obtained with the tree quantizer. For example, an instrumentable tree quantizer characterized by $(M, L, N_s, N, M_o) = (16, 27, 6, 8, 1024)$ is superior to a codebook quantizer which can equivalently be described by $(M, L, N_s, N, M_o) = (1, 1, 1, 40, 1024)$ in terms of objective performance, subjective performance, computational complexity, and memory requirement. High performance is feasible with the tree quantizer without using large quantization block size N.

With the predictable information being parameterized as predictor coefficients, pitch frequencies and gain factors while the unpredictable information processed by the specially designed waveform coder, input speech signals to the coding system can be efficiently encoded with high fidelity at transmission bit rates as low as 4.8 kb/s with an 8 kHz sampling frequency. Computational requirements can be made small by properly selecting the control parameters without sacrificing quality in the reconstructed signals or increasing the total encoding bit rates. The coding system studied can be realized to code speech signals with high fidelity in real time. Many other potential usages such as in voice mail services over telephone lines, integrated voice and data communications, and high density speech storage are also feasible. An even better system may be obtained if the code generator including the definitions of N_s and M_o is more efficiently designed.

References

1. B.S. Atal and R.Q. Hofacker, Jr., "The Telephone Voice of the Future", *AT&T Bell Laboratories Record, pp. 4–10, July 1985*
2. R.H. Robinson, "Digital Voice Compression", *Telecommunications, Global Ed., Vol. 20. No. 2, pp. 33–39, Feb. 1986*
3. M. Schroeder and B.S. Atal, "Code Excited Linear Prediction (CELP): High Quality Speech at Very Low Bit Rates", *Proc. Int. Conf. on Acoustics, Speech and Signal Proc., Vol. 3, Paper no. 25.1, Mar. 1985*
4. B.S. Atal and J.R. Remde "A New Model of LPC Excitation for Producing Natural-Sounding Speech at Low Bit Rates", *Proc. Int. Conf. on Acoustics, Speech and Signal Proc., Vol. 1, pp. 614–617, May 1982*
5. S. Singhal and B.S. Atal, "Improving Performance of Multi-pulse LPC Coders at Low Bit Rates", *Proc. Int. Conf. on Acoustics, Speech and Signal Proc., Vol. 1, Paper no. 1.3, Mar. 1984*
6. F. Jelinek and J.B. Anderson, "Instrumentable Tree Encoding of Information Sources", *IEEE Transactions on Information Theory, pp. 118–119, Jan. 1971*
7. N.S. Jayant and S.A. Christensen, "Tree-Encoding of Speech Using the $(M, L)-$Algorithm and Adaptive Quantization", *IEEE Transactions on Communications, Vol. COM-26, No. 9, pp. 1376–1379, Sept. 1978*
8. J.B. Anderson and J.B. Bodie, "Tree Encoding of Speech", *IEEE Transaction on Information Theory, Vol. IT-21, No. 4, pp. 379–387, July 1975*
9. T. Svendsen, "Tree Encoding of the LPC Residual", *Proc. Int. Conf. on Acoustics, Speech and Signal Proc., Vol. 1, pp. 10.11.1–10.11.4, Mar. 1984*
10. N.S. Jayant and P. Noll, Digital Coding of Waveforms, *Prentice-Hall, N.J., 1984*
11. L.R. Rabiner and R.W. Schafer, Digital Processing of Speech signals, *Prentice-Hall, N.J., 1978*
12. B.S. Atal, "Predictive Coding of Speech at Low Bit Rates", *IEEE Transactions on Communications, Vol. COM-30, No. 4, pp. 600–614, April 1982*
13. J.D. Markel and A.H. Gray, Jr., Linear Prediction of Speech, *Springer-Verlag, New York, 1976*
14. B.S. Atal and M.R. Schroeder, "Predictive Coding of Speech Signals and Subjective Error Criteria", *IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-27, No. 3, pp. 247–254, June 1979*
15. T. Berger, Rate Distortion Theory, A Mathematical Basis for Data Compression, *Englewood Cliffs, N.J., Prentice-Hall 1971*
16. F. Jelinek, "Tree Encoding of Memoryless Time-Discrete Sources with a Fidelity Criterion", *IEEE Trans. on Information Theory, pp. 584–590, Sept. 1969*
17. B.S. Atal and J.R. Remde, "A New Model of LPC Excitation for Producing Natural Sounding Speech at Low Bit Rates", *Proc. Int. Conf. on Acoustics, Speech and Signal Proc., Paris, pp. 614–617, May 1982*

18. M.R. Schroeder and B.S. Atal, “Rate Distortion Theory and Predictive Coding”, *Proc. Int. Conf. on Acoustics, Speech and Signal Proc., Vol. 1, pp. 201–204, Mar. 1981*

19. M. Nakatsui and P. Mermelstein, “Subjective Speech-to-Noise Ratio as a Measure of Speech Quality for Digital Waveform Coders”, *J. Acoust.Soc. Am. 72(4), pp.1136–1144, October 1982*

20. J.L. Flanagan et al, “Speech Coding”, *IEEE Transactions on Communications, Vol. COM-27, No. 4, pp.710–736, April, 1979*