Handbook of Neural Networks for Speech Processing (First edition) pp. 000–000
Ed. S. Katagiri

CHAPTER 4

SPEECH CODING

S. Sridharan

Speech Research Laboratory

Queensland University of Technology
2 George Street, Brisbane, Queensland, 4001, Australia
Email: s.sridharan@qut.edu.au

J. Leis

Faculty of Engineering

University of Southern Queensland
Darling Heights, Queensland, 4350, Australia
Email: leis@usq.edu.au

K. K. Paliwal

School of Microelectronic Engineering

Griffith University
Brisbane, Queensland, 4111, Australia
Email: k.paliwal@me.gu.edu.au

Keywords: Speech coding, vector quantization, linear prediction, signal compression

4.1. Introduction

By "speech coding" we mean a method of reducing the amount of information needed to represent a speech signal. Speech coding has become an exciting and active area of research — particularly in the past decade. Due to the development of several fundamental and powerful ideas, the subject had a rebirth in the 1980's. Speech coding provides a solution to the handling of the huge and increasing volume of information that needs to be carried from one point to another which often leads to the saturation of the capacity of existing telecommunications links – even with the enormous channel capacities of fiber optic transmission systems. Furthermore, in the emerging era of large scale wireless communication, the use of speech coding techniques is essential for the tetherless transmission of information. Not only communication but also voice storage and multimedia applications now require digital speech coding. The recent advances in programmable digital signal processing chips have enabled cost effective speech coders to be designed for these applications.

This chapter focuses on some key concepts and paradigms with the intention of providing an introduction to research in the area of speech coding. It is not the intention of the authors to discuss the theoretical principles underpinning the various speech coding methods nor to provide a comprehensive or extensive review of speech coding research carried out to date.

The fundamental aim is to complement other more technically detailed literature on speech coding by pointing out to the reader a number of useful and interesting techniques that have been explored for speech coding and give references where the reader might find further information regarding these techniques. The chapter also considers the broader aspects of the speech coder which need to be considered when a speech coder is to be incorporated into a complete system.

4.2. Attributes of Speech Coders

A large number of speech coding paradigms have been put forward in the literature. The particular choice for any given application scenario depends on the constraints for that application and invariably, a tradeoff must be made between two or more aspects of the coding method. This is not to imply that simultaneous improvements in several attributes of speech coders cannot be made — indeed, balancing seemingly conflicting requirements in the light of new approaches is likely to be a fruitful area of research for some time to come.

At first sight, it may seem that the primary goal of a speech coding algorithm is to minimize the bit rate. Whilst this aspect is of major importance in many current applications, it is not the only attribute of importance and indeed other attributes may be more important in some cases. The main attributes of a speech coder are:

Bit rate This is the number of bits per second (bps) which is required to encode the speech into a data stream.

Subjective quality This is the perceived quality of the reconstructed speech at the receiver. It may not necessarily correlate to objective measures such as the signal-to-noise ratio. Subjective quality may be further subdivided into *intelligibility* and *naturalness*. The former refers to the ability of the spoken word to be understood; the latter refers to the "human-like" rather than "robotic" or "metallic" characteristic of many current low-rate coders.

Complexity The computational complexity is still an issue despite the availability of ever-increasing processing power. Invariably, coders which are able to reduce the bit rate require greater algorithmic complexity – often by several orders of magnitude.

Memory The memory storage requirements are also related to the algorithmic complexity. Template-based coders require large amounts of fast memory to store algorithm coefficients and waveform prototypes.

Delay Some processing delay is inevitable in a speech coder. This is due not only to the algorithmic complexity (and hence computation time), but also to the buffering requirements of the algorithm. For real-time speech coders, the coding delay must be minimized in order to achieve acceptable levels of performance.

Error sensitivity High-complexity coders, which are able to leverage more complex algorithms to achieve lower bit rates, often produce bit streams which are more susceptible to channel or storage errors. This may manifest itself in the form of noise bursts or other artifacts.

Bandwidth refers to the frequency range which the coder is able to faithfully reproduce. Telephony applications are usually able to accept a lower bandwidth, with the possibility of compromising the speech intelligibility.

Some of these attributes are discussed in greater detail in Section 4.11 and in [1].

4.3. Basic Principles of Speech Coders

In essence, the fundamental aim of a speech coder is to characterize the waveform using as few bits as possible, whilst maintaining the perceived quality of the signal as much as possible. A waveform with a bandwidth of B Hz requires a sampling rate greater than $2B$ samples per second. Each sample in turn requires N bits in order to quantize it. For telephony, a bandwidth of 4 kHz and a quantization to 12 bits is usually required. Simplistic approaches merely use the nonlinear amplitude characteristics of the signal and the human perception of amplitude. The mathematical redundancy that exists between adjacent samples may also be exploited.

In order to achieve a truly low-rate coder, the characteristics of both the signal and the perception mechanism must be considered. A basic division often used to characterize a speech signal is into either *voiced* or *unvoiced* sounds as illustrated in the upper panel of Figure 4.1. The voiced vowel evidently contains two or more periodic components and one would expect that a simpler description of these components would suffice. The pseudo-stationary nature of the waveform means that such a parameterization would suffice over a small but finite time frame. An unvoiced sound as shown in the lower panel of Figure 4.1 appears to contain only random components. It might be expected that in order to code the unvoiced sound a substantially larger number of bits would be required. Although this is true in a mathematical sense, when the aural perceptual mechanism is taken into account the reverse is true.

One basic characterization of voiced sounds is that of the pitch. Figure 4.2 shows the autocorrelation function computed over a short time window for the time-domain waveforms previously shown. The pitch, which is due to the excitation of the vocal tract, is now quite evident for the voiced sound. Thus, the pitch is one parameter which gives the initial characterization of the sound.

In addition to the pitch for voiced sounds, the vocal tract and mouth modulate the speech during its production. Note that in Figure 4.3, the voiced sound contains a definite spectral envelope. The peaks of this envelope correspond to the *formants* or vocal-tract resonances. A speech coder must be able to characterize these resonances. This is usually done through a short-term linear prediction (LP) technique (Section 4.5).

The unvoiced sound also contains less-obvious resonances. However its power spectrum indicates a broader spread of energy across the spectrum. Transform techniques are able to exploit such a non-flat power spectrum by first transforming the time signal into transform-domain coefficients, and then allocating bits in priority order of contribution to overall distortion or perceptual relevance. Noteworthy here is that linear transform techniques have been the mainstay of such coding approaches – nonlinear techniques, although promising, have not been fully exploited.

4.4. Quantization

4.4.1. Scalar Quantization

Quantization is an essential component of speech coding systems. Scalar quantization is the process by which the signal samples are independently quantized. The process is based on the probability density function of the signal samples. An N-level scalar quantizer may be viewed as a one-dimensional mapping of the input range $\mathcal{R}$ onto an index in a mapping table (or codebook) $\mathbf{C}$. Thus

$$Q : \mathcal{R} \rightarrow \mathbf{C} \quad \mathbf{C} \subset \mathcal{R} \tag{4.1}$$

The receiver (decoder) uses this index to reconstruct an approximation to the input level. Optimal scalar quantizers are matched to the distribution of the source samples, which may or may not be known in advance. If the distribution is not known in advance, an empirical choice may be made (for example, a Gaussian or Laplacian distribution) for the purpose of designing the scalar quantizer [2].

4.4.2. Vector Quantization

Vector quantization is a process whereby the elements of a vector of k signal samples are *jointly* quantized. Vector quantization is more efficient than scalar quantization (in terms of error at a given bit rate) by accounting for the linear as well as non-linear interdependencies of the signal samples [3].

The central component of a Vector Quantizer (VQ) is a codebook $\mathbf{C}$ of size $N \times k$, which maps the k-dimensional space $\mathcal{R}^k$ onto the reproduction vectors (also called *codevectors* or *codewords*):

$$Q : \mathcal{R}^k \rightarrow \mathbf{C} \ , \quad \mathbf{C} = (\mathbf{y}_1 \, \mathbf{y}_2 \, \cdots \, \mathbf{y}_N)^T \quad \mathbf{y}_i \in \mathcal{R}^k \tag{4.2}$$

The codebook can be thought of as a finite list of vectors, $\mathbf{y}_i$: $i = 1, \ldots, N$. The codebook vectors are preselected through a clustering or training process to represent the training data. In the coding process of vector quantization, the input samples are handled in blocks of k samples, which form a vector $\mathbf{x}$. The VQ encoder scans the codebook for an entry $\mathbf{y}_i$

that serves best as an approximation for the current input vector $\mathbf{x}_t$ at time t. In the standard approach to VQ, the encoder minimizes the distortion $\mathcal{D}(\cdot)$ to give the optimal estimated vector $\hat{\mathbf{x}}_t$:

$$\hat{\mathbf{x}}_t = \min_{\mathbf{y}_i \in \mathbf{C}} \mathcal{D}(\mathbf{x}_t, \mathbf{y}_i) \tag{4.3}$$

This is referred to as *nearest neighbor encoding.* The particular index i thus derived constitutes the VQ representation of $\mathbf{x}$. This index that is assigned to the selected code vector is then transmitted to the receiver for reconstruction. Note that identical copies of the codebook $\mathbf{C}$ must be located in both the transmitter and the receiver. The receiver simply performs a table lookup to obtain a quantized copy of the input vector.

The *code rate* or simply the *rate* of a vector quantizer in bits per component is thus

$$r = \frac{\log_2 N}{k} \tag{4.4}$$

This measures the number of bits per vector component used to represent the input vector and gives an indication of the accuracy or precision that is achievable with the vector quantizer if the codebook is well designed. Rearranging Equation 4.4, it may be seen that $N = 2^{rk}$, and thus both the encoding search complexity and codebook storage size grow exponentially with dimension k and rate r.

Vector quantization training procedures require a rich combination of source material to produce codebooks which are sufficiently robust for quantization of data not represented in the training set. Examples of some of the conditions which might enrich the training set include varying microphones, acoustic background noise, different languages and gender. In general, a large diverse training set will provide a reasonably robust codebook but there is no guarantee that a new unseen application may not arise. A practical limitation is that codebook design algorithms, such as the Generalized Lloyd Algorithm (GLA) yield only locally optimized codebooks [4]. More recent methods, such as deterministic annealing [5] and genetic optimization [6] promise to overcome this drawback at the expense of greater computational requirements.

4.5. Linear Prediction

4.5.1. Linear Prediction Principles

Linear prediction is the most fundamental technique for removing redundancy in a signal. Linear prediction estimates the value of the current speech sample based on a linear combination of past speech samples. Let $s(n)$ be the sequence of speech samples and a_k be the k^{th} predictor coefficient in a predictor of order p. The estimated speech sequence $\hat{s}(n)$ is given by

$$\hat{s}(n) = a_1 s(n-1) + a_2 s(n-2) + \cdots + a_p s(n-p)$$

$$= \sum_{k=1}^{p} a_k s(n-k) \tag{4.5}$$

The prediction error $e(n)$ is found from

$$e(n) = s(n) - \hat{s}(n) \tag{4.6}$$

By minimizing the mean square prediction error with respect to the filter coefficients we obtain the linear prediction coefficients (see, for example [7]). These coefficients form the *analysis filter*

$$\begin{aligned} A(z) &= 1 - \sum_{j=1}^{p} a_j z^{-j} \\ &= 1 - P_s(z) \end{aligned} \tag{4.7}$$

The filter is sometimes known as a "whitening" filter due to the spectrum of the prediction error, which is (in the ideal case) flat. The process removes the short term correlation from the signal. The linear prediction coefficients are an efficient way to represent the short term spectrum of the speech signal.

For effective use of the linear prediction of speech it is necessary to have a time-varying filter. This is usually effected by redesigning the filter once per frame in order to track the time-varying characteristics of the speech statistics due to the time-varying vocal tract shape associated with successive distinct sounds. Using this "quasi-stationary" assumption, typical speech coders use a frame size of the order of 20 ms (corresponding to 160 samples at an 8 kHz sampling rate).

4.5.2. Speech Coding Based on Linear Prediction

One way in which the linear prediction filter may be used in speech coders is as follows:

1. Subtract the predictable component $\tilde{s}(n)$ of a signal sample $s(n)$ forming the difference or error signal $e(n)$.

2. Quantize $e(n)$ to form $\hat{e}(n)$ and index i.

3. Digitally transmit i to the receiver.

4. At the receiver perform inverse quantization to recover $\hat{e}(n)$.

5. Add $\tilde{s}(n)$ to this quantized difference to form the final reproduction of $\hat{s}(n)$.

Note that the same prediction $\tilde{s}(n)$ has to be generated at the transmitter and the receiver. This is done by using the linear predictor operating on previous reconstructed speech samples $\hat{s}(n)$ to generate $\tilde{s}(n)$. Since $\hat{s}(n)$ is available both at the encoder and decoder the same prediction is generated from either location. This process is illustrated in Figure 4.4. The distribution of the prediction error is normally such that scalar quantization may be applied using an appropriate quantizer [2].

The predictive quantization described here is the basis of the well known Differential Pulse Code Modulation (DPCM) and adaptive differential (ADPCM) – an important standard for speech coding at rates of 24 to 48 kbps.

Speech coders such as ADPCM belong to the category of waveform coders which attempt to reproduce the original speech waveform as accurately as possible. Another class of coders, which are also based on linear prediction, are known as *parametric coders* or *vocoders*. These make no attempt to reproduce the speech waveform at the receiver. Instead, such coders aim to generate a signal that merely sounds similar to the original speech. The key idea is to excite a filter representing the vocal tract by a simple artificial signal which at least crudely mimics typical excitation signals generated by the human glottis. The excitation of the vocal tract is modeled as either a periodic pulse train for voiced speech, or a white random number sequence for unvoiced speech [8]. The speech signal is typically analyzed at a rate of 50 frames per second. For each frame the following parameters are transmitted in quantized form:

1. the linear prediction coefficients;
2. the signal power;
3. the pitch period, and
4. the voicing decision.

This process is shown diagrammatically in Figure 4.5.

The linear predictor $P_s(z)$ that specifies the analysis filter $A(z)$ is called the "formant" or short-term predictor. It is an all pole filter model for the vocal tract and models the short term spectral envelope of the speech signal. The vocoder scheme can synthesize intelligible speech at the very low bitrate of 2400 bps (bit per second) and has served as the underlying technology for secure voice communications. A version of the LP vocoder has been used for several years as the US Government Federal Standard 1015 for secure voice communication (also known as LPC10 because it uses 10th order linear prediction [8]). The bit allocation for this coder is summarized in Table 4.1.

The main weakness of the basic linear prediction based vocoder is the binary decision between voiced and unvoiced speech. Such binary voicing decisions result in low performance for speech segments where both periodic and aperiodic frequency bands are present. More recent work has resulted in the so-called Mixed Excitation Linear Prediction (MELP) coder which has significantly increased the quality of the LPC coder. In this scheme the excitation signal is generated with different mixtures of pulses and noise in each of a number of frequency bands [10]. This scheme (with other innovations) has been selected as the new US Government standard for 2400 bps coding [11].

4.5.3. The Analysis-by-Synthesis Principle

An important concept in speech coding that has become central to most speech coders of commercial interest today is Linear Prediction based

Analysis-by-Synthesis (LPAS) coding. In the LPC vocoder (as described in the previous section), the speech signal is represented by a combination of parameters (filter, gain, pitch coefficients). One method of quantizing each parameter is to compare its value to the stored values in a quantization table and to select the nearest quantized values. The index corresponding to this value is transmitted. The receiver uses this index to retrieve the quantized parameter values for synthesis. This quantization of the parameters is called *open-loop* quantization. An alternative is a process known as *closed-loop* quantization using analysis-by-synthesis. In this method the quantized parameters are used to resynthesize the original signal, and the quantized value which results in the most accurate reconstruction is selected. The analysis-by-synthesis process is most effective when it is performed simultaneously for a number of parameters.

A major reason for using the analysis-by-synthesis coder structure is that it is relatively straightforward to incorporate knowledge about perception. This can be achieved by incorporating a model of the human auditory system in the coder structure. It is well know that otherwise audible signals may become inaudible under the presence of a louder signal. This perceptual effect is called *masking* [12]. Analysis-by-synthesis coders commonly exploit a particular form of masking called *spectral masking*. Given that the original signal has a certain spectrum, the coder attempts to shape the spectrum of the quantization noise such that it is minimally audible under the presence of a louder signal. This means that most of the quantization noise energy is located in spectral regions where the original signal has most of its energy.

In the LPAS approach (Figure 4.6), the reconstructed speech is produced by filtering the signal produced by the excitation generator through both a long-term synthesis filter $1/P(z)$ and a short-term synthesis filter $1/A(z)$. The excitation signal is found by minimizing the weighted mean square error over several samples, with the error signal obtained by filtering the difference between the original and the reconstructed signals through a weighting filter $W(z)$. Both short term and long term predictors are adapted over time. The coder operates on a block-by-block basis. Using the analysis-by-synthesis paradigm, a large number of excitation configurations are tried for each block and the excitation configuration that results in the lowest distortion is selected for transmission. To achieve a low overall bitrate, each frame of excitation samples has to be represented such that the average number of bits per sample is small.

The multipulse excitation coder represents the excitation as a sequence of pulses located at non-uniformly spaced intervals [13]. The excitation analysis procedure has to determine both the amplitudes and positions of the pulses. Finding these parameters all at once is a difficult problem and simpler procedures such as determining the locations and amplitudes one pulse at a time are used. For each pulse, the best position and amplitudes are determined and the contribution of this pulse is subtracted before the next pulse is searched. The number of pulses required for acceptable speech quality varies between 4 to 6 pulses per 5 ms.

In the regular pulse excitation (RPE) coder [14], the excitation signal is represented by a set of uniformly spaced pulses (typically 10 pulses per 5 ms). The position of the first pulse within a frame and the amplitudes of these pulses are determined during the encoding procedure. The bit allocation for the RPE coder as used in the GSM digital mobile telephony standard is shown in Table 4.2.

Code- or vector-excited coders (CELP) use another approach to reduce the number of bits per sample [16]. Here both the encoder and the decoder store a collection of N possible sequences of length k in the codebook, as illustrated in Figure 4.7. The excitation of each frame is described completely by the index to an appropriate vector in the codebook. The index is found by an exhaustive search over all possible codebook vectors and the selection of one that produces the smallest error between the original and the reconstructed signals. The bit allocation for CELP at 4800 bps is summarized in Table 4.3.

The CELP coder exploits the fact that after removing the short and long term prediction from the speech signal, the residual signal has little correlation with itself. A Gaussian process with slowly varying power spectrum can be used to represent the residual signal and the speech waveform is generated by filtering a white Gaussian innovation sequence through time varying long-term and short-term synthesis filters. The optimum innovation sequence is selected from the codebook of random white Gaussian sequences by minimizing the subjectively weighted error between the original and the synthesized speech.

CELP can produce good quality speech at rates of 4.8 kbps at the expense of high computational demands due to the exhaustive search of a large excitation codebook (usually 512-1024 entries) for determining the optimum innovation sequence. However the complexity of the codebook search has been significantly reduced using structured codebooks. A thorough analysis and description of the above methods of LP-based coding may be found in [17] and [18].

4.5.4. Perceptual Filtering

One important factor in determining the performance of the LPAS family of algorithms at low rates is the modeling of the human auditory system. By using the properties of the human auditory system, one can try to reduce the perceived amount of noise. Frequency masking experiments have shown that greater amounts of quantization noise are undetectable by the auditory system in frequency bands where the speech signal has more energy. To make use of this masking effect the quantization noise has to be properly distributed among different frequency bands. The spectral shaping is achieved by the perceptual filter $W(z)$ as shown in Figure 4.7. The filter is essentially a bandwidth expansion filter [18] of the form

$$W(z) = \frac{A(z)}{A(z/\gamma)} \tag{4.8}$$

with the bandwidth expansion controlled by the parameter γ. The effect of this is to broaden the LP spectral peaks by an amount Δf, which is related to the sampling frequency f_s by

$$\Delta f = -\frac{f_s}{\pi} \ln \gamma \quad \text{Hz} \tag{4.9}$$

Despite the error-weighting perceptual filter, it is not always possible to mask the noise in speech caused by the quantization of the excitation signal. By using a separate post-processing filter after reconstruction by the decoder, the perceived noise can be further reduced. An adaptive postfilter of the form

$$H_{apf}(z) = \frac{\left(1 - \mu z^{-1}\right)\left(1 - \sum_{i=1}^{p} a_i \gamma_1^i z^{-i}\right)}{1 - \sum_{i=1}^{p} a_i \gamma_2^i z^{-i}} \tag{4.10}$$

may be incorporated [18]. The rationale here is to add a high-pass component controlled by μ to emphasize the formant peaks, and a pole-zero stage to “flatten” the spectral envelope. The degree of flattening is controlled by the relative values of γ_1 and γ_2, with large differences yielding a quieter but somewhat “deeper” voice [18].

4.5.5. Quantization of the Linear Prediction Coefficients

Many different representations of the linear prediction coefficients are possible. The Line Spectral Frequency (LSF, also known as Line Spectrum Pair or LSP) transformation provides advantages in terms of quantizer design and channel robustness over other linear prediction representations such as reflection coefficients, arc sine coefficients or log area ratios.

To obtain the line spectrum frequency pair representation of the LPC analysis filter, one must take the analysis filter $A(z)$ and its time reversed counterpart $A(z^{-1})$ to create a sum filter $P(z)$ and a difference filter $Q(z)$ as shown below

$$\left.\begin{array}{c} P(z) \\ Q(z) \end{array}\right\} = A(z) \pm z^{-(p+1)} A(z^{-1}) \tag{4.11}$$

The analysis filter coefficients are recovered simply by

$$A(z) = \frac{P(z) + Q(z)}{2} \tag{4.12}$$

The resulting line spectrum frequencies ω_k: $k = 1, \cdots, p$ are simply the alternating values of the roots of the sum and difference filters $P(z)$ and $Q(z)$ respectively. The roots are spaced around the unit circle and have a mirror image symmetry about the real axis (Figure 4.8). There are a number of properties of the LSF representation which make it desirable in coding systems:

1. All roots of the polynomial $P(z)$ and $Q(z)$ are simple and are interlaced on the unit circle.

2. The minimum phase property of $A(z)$ can be preserved if property number (1) above is intact at the receiver. This minimizes the effect of transmission errors and ensures a stable filter for speech reconstruction at the receiver.

3. The LSF's exhibit frequency selective spectral sensitivity. An error in a single LSF will be confined to the region of the spectrum around that frequency.

Many quantization techniques have been developed for representing the linear prediction coefficients with the smallest number of bits. A study of the quantization of linear prediction coefficients is reported in [19]. Scalar quantization techniques quantize the linear prediction coefficients (or parameters derived from the LP coefficients) individually. Vector quantization quantizes a set of LP coefficients (or derived parameters) jointly as a single vector. Vector quantization exploits more efficiently the dependence of the components of the input vector which cannot be captured in scalar quantization. Generally, vector quantization yields a smaller quantization error than scalar quantization at the same bitrate. It has been reported that it is possible to perform perceptually transparent quantization of LPC parameters using 32-34 bits per frame using scalar quantization techniques, and 24-26 bits per frame using vector quantization techniques [20]. For example, quantization of linear prediction coefficients in the FS1015 LPC10 vocoder uses 34 bit scalar quantization. In the new standard FS1017, 25 bit multistage vector quantization is used [11].

There are two major problems with LPC-VQ that need to be overcome. One is the high complexity associated with VQ algorithms which has hindered their use in real time applications. The second is that VQ usually lacks robustness when speakers outside the training sequence are tested.

The majority of vector quantizers used in spectrum quantization today fall within the realm of full, split and multistage vector quantizers. The most basic is the full vector quantizer where the entire vector is considered as an entity for both codebook training and quantization. The two main drawbacks of full-vector VQ are the storage requirements and the computational complexity. Split and multi-stage VQ schemes (also known as "structured codebooks" or "product-code" schemes) have been introduced to reduce the complexity of VQ. Note that as techniques for more elaborate codebook structures are introduced, an increase in distortion level for the same bit rate is also observed when compared to exhaustive-search VQ. The main reason for different VQ structures is the necessity of lowering the complexity and storage requirements of speech coders.

Multistage VQ works by coarsely quantizing the input vector with a first stage codebook, and in doing so creating an error vector e_1 (Figure 4.9). The error vector is then finely quantized by a second stage, and if there are more than two stages a second error vector e_2 is created by the quantization of e_1. This process continues until each stage of the codebook has been used.

The split VQ structure (Figure 4.10) divides the input vector into two or more sub-vectors which are independently quantized. For example, in [20] for a 10 dimensional LP coefficient vector, a two way split partitions the vector between the 4th and 5th parameters. For a three way split the partition could be between parameters 3, 4 and 6, 7 respectively. Splitting reduces search complexity by dividing the vector into a series of sub-vectors depending on how many bits are used for transmission. For the same data the two way equal-split vector quantizer requires half the computational complexity and half the storage capacity of the two-stage vector quantizer. Comparisons of product code methods may be found in [21] and [22]. A generalized theory of product-code vector quantization may be found in [23].

Further bitrate reduction in the quantization of linear prediction coefficients may be achieved by exploiting interframe correlation or time dependence between spectral parameters. Finite state vector quantization uses an adaptive codebook, which is dependent on previously selected codewords, in quantizing an input vector. Essentially these quantizers exploit the underlying content of speech to provide transmitter and receiver with mutual information which is not transmitted [24]. Variable frame rate VQ quantizes the linear prediction coefficients only when the properties of speech signal have changed significantly. The LP coefficients of the non-quantized frames are regenerated through linear or some other interpolation of the parameters of other quantized frames [25]. These quantization techniques are useful for coding as low as 400 bps but are not sufficient for maintaining reasonable spectral distortion and speech intelligibility for data rates below about 300 bps [26].

Another approach using the time dependence of LPC is to perform VQ on multiple frame vectors. This method is referred to as *matrix quantization* or *segment quantization*. Segment quantization is an extension of vector quantization based on the principle that rate distortion performance is improved by using longer blocks for quantization. The time dependence of consecutive frames is included implicitly in the spectral segment, unlike the situation in variable frame rate VQ. This approach has the potential to achieve speech coding at very low rates (below 300 bps) despite the substantial computational burden.

Adaptive vector quantizers allow for low probability codevectors which have not been used in a specified time period to be eliminated from the codebook. These codevectors are replaced by higher probability codevectors selected from the current input data but not adequately represented in the existing codebook. In effect, the current input data is added to the training set. These updated code vectors are then transmitted to the receiver during periods of silence or low speech activity. In the limit this, technique can approximate a vector quantizer which was trained on source material from the current user, yielding a decrease in overall distortion.

The main drawback of this method is that the transmission of updated code vectors across the channel without errors is required [27].

4.6. Sinusoidal Coding

Instead of using LP coefficients, it is possible to synthesize speech with an entirely different paradigm – namely by generating a sum of sinusoids whose amplitudes, phases and frequencies are varied with time [28]. This certainly seems to be a reasonable way to generate a periodic waveform. In fact it can also be applied to the synthesis of unvoiced speech as well. The synthesis can also be improved by generating a suitable mixture of random noise with a discrete set of sinusoids so that both unvoiced and voiced speech can be more effectively modeled. The synthesizer must seamlessly adjust the sinusoidal parameters to avoid discontinuities at the frame boundaries. Critical to this synthesis concept is effective analysis of the speech, which determines for each frame the needed sinusoidal frequencies, amplitudes and phases. The key to this analysis is to examine and model the short term spectrum of the input speech with a minimal and effective set of parameters.

The sinusoidal representation of the speech waveform using L sinewaves, given an analysis frame of length N samples, is

$$\hat{s}(n) = \sum_{l=1}^{L} A_l \cos(n\omega_l + \phi_l) \tag{4.13}$$

in which the sine waves are multiples of the fundamental frequency ω_l. The corresponding amplitudes A_l and phases ϕ_l are given by the harmonic samples of the Short-Time Fourier Transform (STFT).

When the speech is not perfectly voiced, the STFT will have a multiplicity of peaks that are not necessarily harmonic. These peaks can be used to identify the underlying sine wave structure.

The types of parameters that are coded in sinusoidal coding differ significantly for different bitrates. At higher bitrates, the entire sparse spectrum (magnitudes, phases and frequencies) and overall power are transmitted. At lower rates the phases are modeled and frequencies are constrained to be harmonics. Thus the fundamental frequency, signal power, a description of the sine wave amplitudes, and the parameters of the phase model are transmitted at low bitrates. The sinewave amplitudes can be modeled in a number of ways including all-pole modeling and differential quantization [28]. The phase information is transmitted only for bit rates above 9.6 kbps. Excellent quality output can be obtained with an analysis-synthesis system based on sinusoidal coding when the sparse magnitude and phase spectra are updated every 10 ms. At lower bitrates the phase spectrum of the reconstructed speech is obtained using a model. Different models are used for voiced and unvoiced speech. Sinusoidal coders are now viewed as viable alternative to CELP, particularly at the rates of 2-4 kbps and below.

4.7. Waveform Interpolation Methods

A recently introduced paradigm is that of Prototype Waveform Interpolation (PWI), based upon a representative or Characteristic Waveform (CW). Several observations motivate this approach [29]. Firstly, previous low-rate methods such as CELP are essentially waveform matching procedures which attempt to match the waveform on a frame-by-frame basis using a segmental signal-to-noise (SNR) criteria. As the bitrate is lowered, the quality of the match deteriorates – this is especially so for CELP below about 4.8 kb/s. As pointed out in [29], the SNR is far from an ideal perceptual criteria. The waveform-matching criteria is therefore relaxed, and more emphasis is placed on the pitch waveform periodicity and pitch cycle dynamics (evolution over time). Thus, the *perceptual* quality improves, although the computed SNR decreases.

The operation of the PWI method is essentially as follows. Prototype waveforms are extracted at intervals of 20-30 ms. This extraction may be done in the LP residual domain, with the residual prototype quantized using standard analysis-by-synthesis techniques. Care must be taken to ensure continuity at the block boundaries, to ensure the smoothness and periodicity of the resulting waveform. As pointed out in [30], the PWI method reduces to LP vocoding (a single pitch impulse) if only single impulses are used for the prototype excitation waveforms. The method is claimed to produce good speech at rates of 3-4 kbps, but is suitable only for the voiced sections of speech. More recent work has extended the method to unvoiced speech as well, decomposing the CW into a so-called Rapidly Evolving Waveform (REW) and a Slowly Evolving Waveform (SEW). Recent results have reported that a 2.85 kbps coder operating on this principle achieves perceptual quality comparable to the FS1016 CELP coder [31].

4.8. Sub-band Coding

In sub-band coding (SBC), the speech signal is filtered into a number of subbands and each subband is adaptively encoded. The number of bits used in the encoding process differs for each subband signal with bits assigned to quantizers according to a perceptual criteria. By encoding each subband individually, the quantization noise is confined within its subband. The output bitstreams from each encoder are multiplexed and transmitted. At the receiver demultiplexing is performed, followed by decoding of each subband data signal. The sampled subband signals are then combined to yield the recovered speech.

Note that downsampling of subband signals must occur at the output of the subband filters to avoid oversampling. The downsampling ratio is given by the ratio of original speech bandwidth to subband bandwidth. Conventional filters cannot be used for the production of subband signals because of the finite width of the band-pass transition bands. If the bandpass filters overlap in the frequency domain, subsampling causes aliasing which destroys the harmonic structure of voiced sounds and results in unpleasant perceptual effects. If the bandpass filters don't overlap, the speech signal cannot be perfectly reconstructed because the gaps between the channels introduce an audible echo. Quadrature mirror filter (QMF) banks [32] overcome this problem and enable perfect reconstruction of the speech signal.

4.9. Variable-Rate Coding

4.9.1. Basics

The speech coders described above apply a specific unchanging coding technique to the continuously evolving speech signal without regard to the varying acoustic-phonetic character of the signal. Recently, new varieties of LPAS and sinusoidal coders have emerged which do not apply the same coding technique to each of the input frames. Instead, one of several distinct coding techniques is selected for each frame, each with its own bit allocation scheme for the parameters. The encoder in such a coding scheme selects one out of a predetermined set of techniques as the one best suited to the local character of the speech signal. The decoder, having this information, applies the corresponding decoding algorithm. This type of coding offers the opportunity to dynamically tailor the coding scheme to the widely varying local acoustic and phonetic properties of the speech signal. Variable bitrate coding (VBR) is an extension of the above coding techniques where the total allocation of bits for the frame is allowed to vary, adapting the rate the local phonetic character and/or network conditions. VBR coders benefit from this additional degree of freedom by allocating each frame the minimum bits necessary for the decoder to adequately reproduce a frame. For example, in a typical telephone call, roughly 60% of the time one of the speakers is silent and the signal contains only the background noise. In this situation a unimodal coder diligently encodes such non-speech segments with the same resolution, the same algorithm and the same bit allocation as it does for active speech segments. Clearly this is inefficient. In contrast, a VBR coder will use only a very minimal bitrate to encode the non-speech segments. The overall quality will remain the same but the average bitrate will be much lower than a comparable unimodal fixed rate coder. VBR multimodal coders are particularly advantageous for voice storage systems, code division multiplex access (CDMA) wireless networks, and packetized communication systems.

4.9.2. Phonetic Segmentation

Further bitrate reduction can be achieved by adapting the coder to match phonetically distinct frames. In the work described in [33], speech is segmented in to 3 major categories: voiced, unvoiced or onset. The onset category is defined as the transition from an unvoiced to a voiced region.

The voiced class is further subdivided into four categories through two layers of segmentation. Different coding strategies are used for different classes.

Another approach to very low bitrate coding based on variable rate is Temporal Decomposition (TD). In [34], the method is considered an efficient technique to reduce the amount of spectral information conveyed by spectral parameters through the orthogonalization of the matrix of spectral parameters. Specifically, a $p \times N$ dimensional vector of spectral parameters $\mathbf{Y}$ is approximated in the form

$$\hat{\mathbf{Y}} = \mathbf{A}\Phi \tag{4.14}$$

where Φ is a $m \times N$ matrix of *event functions*, $\mathbf{A}$ is a $p \times m$ matrix of weightings. It has been shown that almost all phonemes can be described by event functions (74% with only single event function) by using a suitable parameter set [34]. It is necessary to code both $\mathbf{A}$ and ϕ for speech coding. In doing so, a considerable coding gain is achieved in coding the spectral parameters themselves. In [35] VQ was used in combination with TD to encode speech. Non real-time speech coders at rates between 450-600 bps with naturally sounding speech were claimed. A new technique called Hierarchical Temporal Decomposition (HTD) has recently been proposed [36-37] which reduces the computational complexity of temporal decomposition.

4.9.3. Variable Rate Coders for ATM Networks

An area in which much of the current variable rate speech coding research effort is being directed is Asynchronous Transfer Mode (ATM) networks. ATM networks have been proposed in order to provide a common format for both high speed data and real time traffic such as speech and video. ATM uses short length packets called *cells* which are 53 bytes long [38]. Of these, 5 bytes are reserved for header information and 48 bytes are for information. One way of using variable rate coding in ATM is to use a hierarchical "package" of bits for a given duration of speech. Through hierarchical packaging of bits (such as putting the more significant bits in one cell and the less significant bits in another cell), priorities can be attached to cells which can be used by ATM network to control the traffic flow. Low priority cells can be dropped when the network becomes congested without severely reducing the speech quality. The technique by which an ATM network drops cells is known as *cell discarding*. Achieving higher compression through rate reduction is not straightforward in ATM systems. As the rate decreases, longer encoding frames must be used to fill fixed length ATM packets. This has implications for both delay and recovery from cell loss due to corrupted headers or buffer overflow. There is a need to design more efficient speech coders for ATM's fixed length packet technology.

4.9.4. Voice over IP

Another area of considerable interest at present is that of transmitting voice signals over internet connections — known as Voice over IP (VoIP). This uses conventional IP infrastructure to transmit digitally encoded voice signals, promising a considerable reduction in costs to the end-user. Because the TCP/IP protocols are based on packet store-and-forward technology with no provision for prioritized sending, traffic bursts may lead to a highly variable transmission time between the communication endpoints. This is the opposite of what is required for real-time voice communications. Problems such as variable network delay, packet errors and missing packets combine to create a very difficult environment for real-time speech traffic. The bandwidth of the compressed speech is also of interest, as a lower bit rate requirement can go some way to reducing the delay and providing more space for buffering and error concealment. At present, a number of competing vendor implementations exist.

4.10. Wideband Coders

The 300-3400 Hz bandwidth requires a sampling frequency of 8 kHz and provides speech quality referred to as "toll quality". Even though this is sufficient for telephone communications and emerging applications such as teleconferencing, improved quality is necessary. By increasing the sampling frequency to 16 kHz, a wider bandwidth ranging from 5 Hz to 7000 Hz can be accommodated. Extending the lower frequency range down to 50 Hz increases naturalness, presence and comfort. At the other end of the spectrum, extending the higher frequency range to 7000 Hz increases intelligibility and makes it easier to differentiate between sounds such as, for example, *s* and *f*. This results in a speech signal that is more natural. In speech transmission there is very little subjective improvement to be gained by further increase of the sample rate beyond 16 kHz.

In wideband coders the perceived quality of speech is very important and therefore the frequency masking properties of the human auditory system should be fully exploited to make the coding noise inaudible. One of the major challenges for wideband coder designs is to retain as much speech modeling as possible (to achieve high speech quality at low bitrates) whilst allowing other types of signals such as music to be encoded without significant degradation. The reason for this is that audio channels in applications such as teleconferencing and high definition television do not only carry a speech signal, even though speech is likely to constitute a large part of the information transmitted on these channels.

In 1986 the International Telegraph and Telephone Consultative Committee (CCITT, now known as the International Telecommunication Union or ITU-T) recommended the G.722 standard for wideband speech and audio coding. This wideband codec provides high quality speech at 64 kbps with an bandwidth of 50 to 7000 Hz. Slightly reduced quality is achieved at 56 and 48 kbps. The G.722 coder is essentially a two subband coder with ADPCM encoding of each subband [39]. New coding schemes for low rates at 16, 24 and 32 kbps are currently being studied for standardization. The G.722 standard will serve as a reference for the development of these alternative coding schemes.

4.11. Measuring the Performance of Speech Coders

The quality of speech output of a speech coder is a function of bitrate, complexity, delay and bandwidth. It is important to consider all these attributes when assessing the performance of a speech coder. These four attributes are related to each other. For example, a low bitrate coder will have higher computational complexity and higher delay compared to a higher bitrate coder.

The delay in a speech coding system consists of three major components [40]. Before speech can be coded it is usually necessary to buffer a frame of data. The delay due to this process cannot be avoided and is known as *algorithmic delay*. The second is the *processing delay* which is the time taken for the encoder to analyze the speech and the decoder to reconstruct the speech. This delay will depend on the hardware used to implement the speech coder/decoder. The *communications delay* is the third component, and is the time taken for the entire data to be transmitted from the encoder to the decoder. The total of these three delays is known as the "one-way" system delay. If there are no echoes in the system then one-way delays of up to 400 ms can be tolerated. In the presence of echoes, the one-way delay can be only 25 ms.

Speech coders are usually implemented using digital signal processors. The complexity of the implementation can be measured by the computing speed requirement of the processor, together with the memory requirements. More complexity results in higher costs and greater power usage. For portable applications, greater power usage means either reduced time between battery recharges, or using larger batteries (which means more expense and weight).

Methods of assessment of speech quality have been important in the development of high quality low bitrate speech coders. Standardization activities over the past few years have resulted in an increasing need to develop and understand the methodologies used to subjectively assess new speech coding systems before they are introduced into the telephone network.

Speech quality has many perceptual dimensions but the most important are the *intelligibility* and *naturalness* of speech. These attributes are tightly coupled, but are not equivalent. For example in speech synthesis, the output speech may sound artificial but could be highly intelligible. The perceived quality of speech in a system incorporating speech coders will depend on a number of factors. For example in a cellular application, the performance depends on the transducers, speech coder, error correction, echo cancellation procedures, switches, transmitters and receivers. Poor performance of any of these parts will affect the speech quality.

For coders that approximate the waveforms of the input signal, one can measure the difference between the input and the output to quantify the quality. Many objective measures have been studied. For example, the Signal to Noise Ratio (SNR) may be computed over the complete duration of the signal. This approach has the drawback that the SNR value will be dominated by the regions that have high energy. Since speech is a non-stationary signal with many high and low energy sections that are perceptually relevant, a better approach is to compute the SNR for shorter segments and then compute the mean over the entire duration. The measure is referred to as Segmental Signal-to-Noise Ratio (SEGSNR). Further refinement of the SEGSNR may be obtained by clipping the maximum and minimum values and excluding silence segments.

Since SNR operates on the complete frequency band it will not give any information about the frequency distribution of the error signal. The frequency-dependent SNR can be computed by filtering the signals through a filter bank and computing the SNR for each frequency band.

Segmental SNR has been shown to correlate reasonably well with speech quality [12]. However these methods are extremely sensitive to waveform misalignment and phase distortions which are not perceptually relevant. One approach that eliminates the effect of phase mismatches is to compute the differences between the power spectra. Since linear prediction techniques accurately model spectral peaks – which are perceptually relevant – the difference between the LP spectra of the original and coded speech may be used to measure the perceptual difference between the two signals. One of the more popular measures is based on the Euclidean distance between the cepstral coefficients. Another measure commonly used in the quantization of LPC coefficients is the *spectral distortion* defined as

$$SD_n = \sqrt{\frac{1}{B}\int_R \left(10\log P_n(\omega) - 10\log \hat{P}_n(\omega)\right)^2\, d\omega} \qquad \text{dB} \qquad (4.15)$$

where $P_n(\omega)$ is the power in the n^{th} frame due to the short-term filter, and $\hat{P}_n(\omega)$ is the power in the quantized version.

As may be observed from Equation 4.15, it is effectively an RMS measure of the difference in power between the quantized and unquantized spectra. For 8 kHz sampling, some authors use R as the band 125 Hz to 3400 Hz, whilst others use a range of 0 to 3 kHz. This measure is useful in the design process for *objectively* determining spectral distortion – however it still lacks a direct correlation with *subjective* measures. To this end, the Bark Spectral Distortion (BSD) is proposed in [12], which is a step towards a fully objective metric that is useful in predicting the subjective quality of speech coders.

In a subjective test, speech is played to a group of listeners who are asked to rate the quality of the speech signal. In most tests the minimum number of listeners is 16 but could be as high as 64. The maximum number of listeners is usually limited by cost and time limitations. For most subjective tests non-expert listeners are used. The main reason is that this would better reflect the conditions under which the system will eventually be used. The Mean Opinion Score (MOS) is an absolute category rating in which listeners are presented with samples of processed material and are asked to give ratings using a 5 point scale — excellent (5), good (4), fair (3), poor (2), bad (1). The average of all votes obtained for a particular system represents the MOS. For some applications the distribution of ratings is relevant. For example, in telecommunication applications the percentage responses that get a rating of “poor” or “bad” quality could identify future user non-acceptance.

Conducting subjective quality tests is an expensive and time-consuming procedure. It would be useful if one could predict the subjective performance through some computational process acting directly on the original and coded signals. One approach is to model the human auditory system and use both the unprocessed and processed speech as input to this model [41]. The output of the model is compared for both signals and the difference indicates the difference in quality. The remaining problem is how to correlate the differences in auditory model outputs to subjective scores — both clustering and heuristic procedures have been used.

The quality of some speech coders is speaker-dependent [42]. This is a direct result of some of the coding algorithms used (such as linear prediction or pitch prediction). It is therefore necessary to test the coder with a number of different speakers. A common number of speakers that has been used in many evaluations tests is at least 4 males, 4 females, and two children. For coders that are used in different countries, it is also important to assess if there is any dependency on the language [43].

Sometimes non-speech signals such as music are presented to the coder. This situation can occur, for example, in telephone applications where the caller is put on hold. During the hold-time it is common to play music to the caller which is referred to as "music on hold". It is important to note that this situation is different from a background music signal, since now it is the primary signal. Although one cannot expect that the low bitrate speech coder would faithfully reproduce music signals, it is often required that no annoying effects be noticeable.

4.12. Speech Coding Over Noisy Channels

The fading channels that are encountered on mobile radio systems often produce high error rates and the speech coders used on these channels must employ techniques to combat the effects of channel errors. Due to the narrow spectral bandwidth assigned to these applications, the number of bits available for forward error detection and correction will necessarily be small. This has forced codec designers to use different levels of error protection. Specifically, a given parameter being encoded with a B bit quantizer will have B_1 of these bits highly protected, an average level of protection will be given to the next B_2 bits and the remaining $B - B_1 - B_2$ bits will be left unprotected. The selection of which bits should be placed in which class is usually done by subjectively evaluating the impact on the received speech quality of an error in a given bit position.

Unequal error protection can be exploited in the design of the quantizer (vector or scalar) to enhance the overall performance. In principle what is needed is to match the error protection to the error sensitivity of the different bit positions of the binary word representing a given parameter. This error sensitivity can be defined as the increase in distortion when that bit position is systematically altered by a channel error.

A tailoring of the bit error sensitivity profile can be accomplished in at least two ways [44]:

1. By a judicious assignment of the binary indices to the output levels in which errors in the more vulnerable bits will most likely cause the transmitted word to be received as one of the neighboring codewords, and

2. By adjusting the quantizer design procedure so that the codevectors are more suitably clustered.

4.13. Speech Coding Standards

Speech coding standards are necessary for interoperability. For interoperability to be achieved, standards must be defined and implemented. All telecommunications applications clearly belong to this class, as well as some storage applications such as compact discs. There are, however, speech coding applications in which interoperability is not an issue, so no standards are required. Examples of such applications are digital answering machines and voice mail storage — customized coders may be used for these applications. Many of the early speech coding standards were created by the US Department of Defense (DoD) for secure speech coding applications. An important standards body is the International Telecommunications Union (ITU). ITU defines standards for international telephone networks. Another standard of importance is MPEG (Motion Picture Experts Group), which defines standards for compression and is released by the ISO/IEC. Recently there has been a flurry of activity in developing standards for wireless cellular transmission.

A speech coding standard currently being prepared is the MPEG4 standard for compression of speech between 2 and 24 kbps. This standard uses Code Excited Linear Prediction (CELP) with Harmonic Vector Excitation Coding (HVXC) [45] over the range 2-4 kbps. A key feature of this standard is scalability [36].

4.14. Conclusions

This chapter introduced some of the issues involved in speech coding research and the design of practical speech coders. The fundamental model around which speech coders at low rates are based – the linear model approach – has remained the dominant research focus in recent times. The linear prediction approach – the heart of most current implementations – was introduced, with some examples to motivate its application. Vector quantization – which is essentially a template-based pattern-matching technique – was also introduced, and shown to be a method which can leverage substantial savings in bit rate at the expense of considerably greater complexity.

Significant advances in speech coding have been made by incorporating vocal tract models and aural perception models. The reason for this success is that these models have captured, however poorly, some basic properties of speech production and perception. Further improvement in speech coding systems will undoubtedly come from better understanding of the speech production and perception mechanisms.

An area of significant research activity currently is very low bit rate coding (below 1000 bps). At these rates the authors believe that the use of language models will play a key role in improving coder performance. By studying the relationship between abstract elements of the language and how they manifest themselves in actual speech waveforms, a clear picture of the acoustic features of speech that need to be preserved in the coding process will be obtained.

To use language models in speech coding, the speech coder may incorporate the speech recognition process and the decoder may incorporate the speech synthesis process. Some recent results in this area are reported in [46-47].

References

1. W. B. Kleijn and K. K. Paliwal (eds), *Speech Coding and Synthesis* (Elsevier, 1995)
2. N. S. Jayant and P. Noll. *Digital Coding of Waveforms: Principles and Applications to Speech and Video* (Prentice-Hall, 1984)
3. A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression.* (Kluwer Academic Publishers, 1992)
4. Y. Linde, A. Buzo, and R. M. Gray. An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications* **COM-28(1)** (January 1980) 84–94.
5. C. R. Nassar and M. R. Soleymani. Codebook Design for Trellis Quantization Using Simulated Annealing. *IEEE Transactions on Speech and Audio Processing* **1(3)** (October 1993) 400–404.

6. S. Choi and W. K. Ng. Vector Quantizer Design Using Genetic Algorithms. *Proc. IEEE Data Compression Conference (DCC-96)* (1996)

7. J. Makhoul. Linear Prediction: A Tutorial Review. *Proceedings of the IEEE* **63(4)** (April 1975) 561–575.

8. T. E. Tremain. The Government Standard Linear Predictive Coding Algorithm. *Speech Technology* (1982) 40–49.

9. T. Parsons. *Voice and Speech Processing* (McGraw-Hill, 1987)

10. A. V. McCree and T. P.Barnwell III. A Mixed Excitation LPC Vocoder Model for Low Bitrate Speech Coding. *IEEE Transactions on Speech and Audio Processing* **3(4)** (July 1995) 242–250.

11. A. V. McCree *et al.* A 2.4 kbit/sec Coder Candidate for the New US Federal Standard, *Proc. ICASSP'96* (1996) 200–203.

12. S. Wang, A. Sekey, and A. Gersho. An Objective Measure for Predicting Subjective Quality of Speech Coders. *IEEE Journal on Selected Areas in Communications* **10(5)** (June 1992) 819–829.

13. B.S. Atal and J.R. Remede, A New Model LPC Excitation for Producing Natural Sounding Speech at Low Bit Rates, *Proc. ICASSP'82* (1982) 614–617.

14. P. Kroon, E. F. Deprettere, and R. J. Sluyter. A Novel Approach to Efficient Multipulse Coding of Speech. *IEEE Transactions on Acoustics, Speech and Signal Processing* **34(5)** (October 1986) 1054–1063.

15. P. Kroon and W. B. Kleijn. Linear Predictive Analysis by Synthesis Coding, in *Modern Methods of Speech Processing*, Chapter 3. (Kluwer Academic Publishers, 1995)

16. M. R. Schroeder and B. S. Atal. Code Excited Linear Prediction (CELP): High Quality Speech at Low Bitrates, *Proc. ICASSP'85* (1985) 937–940.

17. R. Salami, L Hanzo, R. Steele, K. Wong, and I. Wassell. Speech Coding, in *Mobile Radio Communications* (Pentech, 1992)

18. A. M. Kondoz. *Digital Speech – Coding for Low Bit Rate Communications Systems* (John Wiley, 1994)

19. K. K. Paliwal and W. B. Kleijn. LPC Quantization, in *Speech Coding and Synthesis* 433–466 (Elsevier, 1995)

20. K. K. Paliwal and B. S. Atal. Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame. *IEEE Trans. Speech and Audio Processing* **1(1)** (January 1993) 3–14.

21. S. Wang, E. Paksoy, and A. Gersho. Product Code Vector Quantization of LPC Parameters, in *Speech and Audio Coding for Wireless Network Applications*, 250–258 (Kluwer Academic Press, 1993)

22. J. S. Collura. Vector Quantization of Linear Predictor Coefficients, in *Modern Methods of Speech Processing*, Chapter 2. (Kluwer Academic Publishers, 1995)

23. W. Y. Chan and A. Gersho, Generalized Product Code Vector Quanti-

zation: A Family of Efficient Techniques for Signal Compression, *Digital Signal Processing* **4(95)** (1994) 95–126.

24. M. O. Dunham and R. M. Gray. An Algorithm for the Design of Labeled-Transition Finite-State Vector Quantizers. *IEEE Transactions on Communications* **COM-33(1)** (January 1985) 83–89.

25. C.-J. Chung and S.-H. Chen. Variable Frame Rate Speech Coding Using Optimal Interpolation. *IEEE Transactions on Communications* **42(6)** (June 1994) 2215–2218.

26. V. Viswanathan, J. Makhoul, R Schwarz, and A.W.F. Higgins. Variable Frame Rate Transmission: A Review of Methodology and Application to Narrowband LPC Speech Coding. *IEEE Transactions on Communications* **COM-30(4)** (April 1982) 674–686.

27. D. B. Paul. An 800 bps Adaptive Vector Quantization Vocoder Using a Perceptual Distance Measure, *Proc. ICASSP-83*, (1983) 67–71.

28. R. J. McCaulay and T. F. Quatieri. Low-Rate Speech Coding Based on the Sinusoidal Model, in *Advances in Speech Signal Processing*. (Marcel Dekker, 1992)

29. W. B. Kleijn. Encoding Speech Using Prototype Waveforms. *IEEE Transactions on Speech and Audio Processing* **1(3)** (October 1993) 386–399.

30. W. Bastiaan Kleijn and W. Granzow. Waveform Interpolation in Speech Coding, in *Speech and Audio Coding for Wireless Network Applications*

111–118 (Kluwer Academic Press, 1993)

31. W. B. Kleijn and J. Haagen. Transformation and Decomposition of the Speech Signal for Coding. *IEEE Signal Processing Letters* **1(9)** (September 1994) 136–138.

32. D. Esteban and C. Galand. Application of Quadrature Mirror Filters to Split Band Voice Coding Scheme, *Proc. ICASSP-77* (1977) pages 191–195.

33. S. Wang and A.Gersho. Phonetic Segmentation for Low Rate Speech Coding, in *Advances in Speech Coding.* (Kluwer Academic Publishers, 1991)

34. B. S. Atal, Efficient Coding of LPC Parameters by Temporal Decomposition, *Proc. ICASSP'83* (1983) 81–84.

35. Y. M. Cheng and D. O. O'Shaughnessy, Short-Term Temporal Decomposition and its Properties for Speech Compression, *IEEE Trans. on Signal Processing* **39(6)** (1991) 1281–1290.

36. B. Edler, Speech Coding in MPEG4, *International Journal on Speech Technology* **2** (1999) 289–303.

37. S. Ghaemmaghami, M. Deriche and S. Sridharan, Hierarchical Temporal Decomposition: A Novel Approach to Efficient Compresison of Speech, *Intl. Conf on Spoken Language Processing (ICLSLP-98)* (1998) 2567–2570.

38. W. Fischer, E. Wallmeier, T. Worster, S. Davis, and A. Hayter. Data Communications Using ATM: Arthitectures, Protocols and Resource Management. *IEEE Communications Magazine* **32(8)** (August 1994) 24–33.

39. CCITT. 7 kHz Audio Coding at 64 kbits/sec, *Recommendation G.722, Fascile III.4, Blue Book* (1988) 269–341.

40. W. B. Kleijn. An Introduction to Speech Coding, in *Speech Coding and Synthesis* 1–47 (Elsevier, 1995)

41. O. Ghitza. Auditory Models and Human Performance Tasks Related to Speech Coding and Speech Recognition, *IEEE Trans. Speech and Audio Processing* **2(1)** (1994) 115–132.

42. N. Kitawaki. Quality Assessment of Coded Speech, in *Advances in Speech Signal Processing* (Marcel Dekker, 1992)

43. R. Montagna. Selection Phase of the GSM Half Rate Channel, *Proc. IEEE Speech Coding Workshop* (1993) 95–96.

44. J. R. B. de Marca. On Noisy Channel Quantizer Design for Unequal Error Protection, in *Advances in Speech Signal Processing* (Marcel Dekker, 1992)

45. M. Nishiguchi, K. Iijima and J. Matsumoto, Harmonic Vector Excitation Coding at 2kbps, *Proc. IEEE Workshop on Speech Coding for Telecommunications* (1997) 39–40.

46. C. M. Ribeiro and I. M. Trancoso, Phonetic Vocoding with Speaker Adaptation. *Proc. Eurospeech-97* (1997) 1291–1299.

47. K-S. Lee and V. Cox, TTS Based Very Low Bit Rate Speech Coder, *Proc. ICASSP-99* (1999).

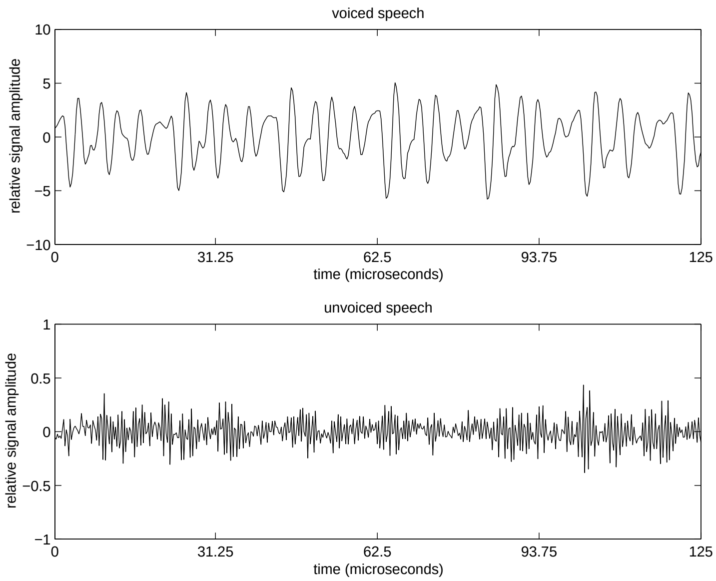

Figure 4.1: Time-domain waveforms for voiced (top) and unvoiced (lower) speech. The sampling rate is 8 kHz.

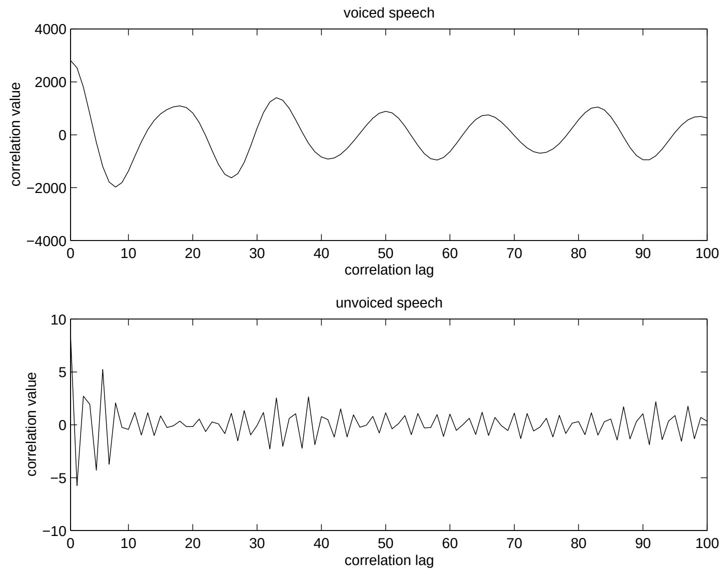

Figure 4.2: Autocorrelation of voiced (top) and unvoiced (lower) speech segments.

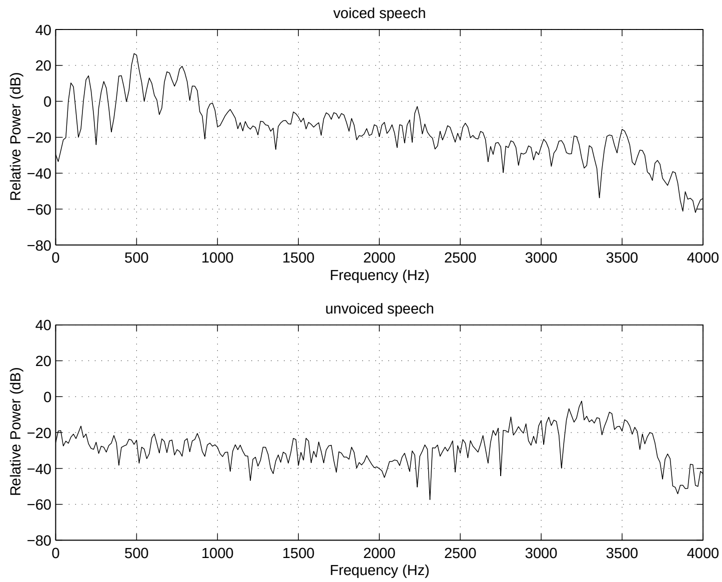

Figure 4.3: Relative power levels of voiced (top) and unvoiced (lower) speech segments.

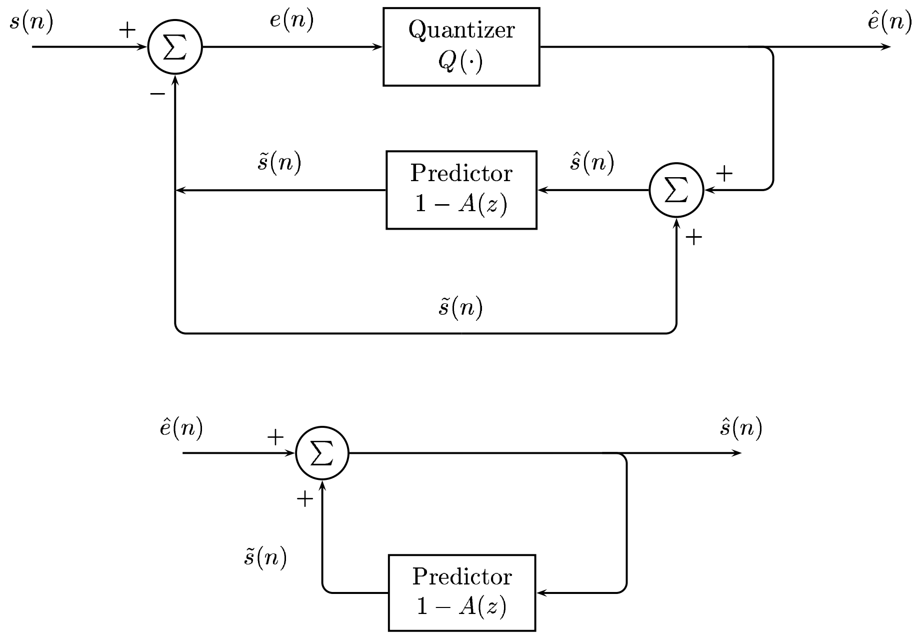

Figure 4.4: A differential PCM (DPCM) coder. At the encoder, the prediction is based upon the *quantized* prediction error $\hat{e}(n)$ together with past predictions $\tilde{s}(n)$. At the decoder, the prediction (based on past quantized outputs) is added to the received error signal $\hat{e}(n)$ to generate each output $\hat{s}(n)$.

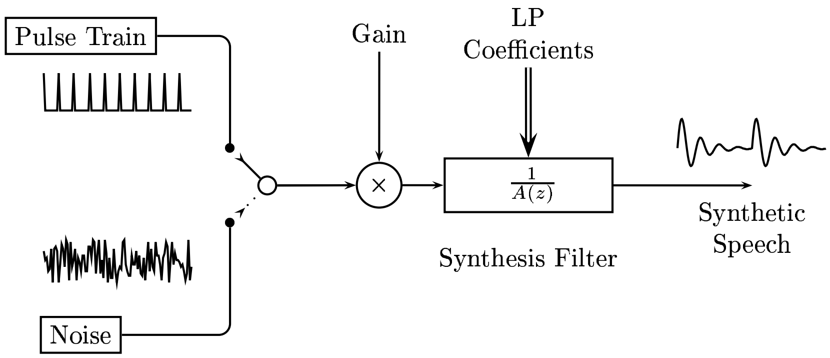

Figure 4.5: Linear predictive (LP) coder using a simple pulse train or noise excitation, corresponding to voiced or unvoiced speech. The LP coefficients and gain must be updated for each frame of speech.

Table 4.1: Summary of bit allocation for 2.4 kbps LPC-10 speech coder (after [9]).

Sample Rate	8 kHz
Frame size	180 samples
Frame rate	44.44 frames/second
Pitch	7 bits
Spectrum (5,5,5,5,4,4,4,4,3,2)	41 bits
Gain	5 bits
Spare	1 bit
Total	54 bits/frame
Bit Rate	$54 \times 44.44 = 2400$ bits/sec

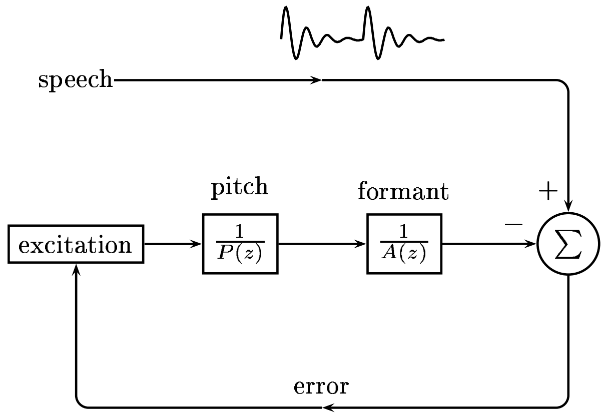

Figure 4.6: Analysis-by-synthesis minimization loop.

Table 4.2: Summary of bit allocation for GSM RPE-LTP 13 kbps speech coder (after [15]).

Sample Rate	8 kHz
Frame size	160 samples (20 ms)
Frame rate	50 frames/second
Subframe size	40 samples (5 ms)
Pulse spacing	3 (13 pulses/5 ms)
Pitch Lag (40 to 120) (7,7,7,7)	28 bits
Pitch Gain (0.1 to 1) (2,2,2,2)	8 bits
Spectrum (6,6,5,5,4,4,3,3)	36 bits
Excitation Pulse Position (2,2,2,2)	8 bits
Subframe Gain (6,6,6,6)	24 bits
Pulse Amplitudes (4×39)	156 bits
Total	260 bits/frame
Bit Rate	$260 \times 50 = 13000$ bits/sec

Table 4.3: Summary of bit allocation for FS1016 4.8 kbps CELP speech coder (after [8]).

Sample Rate	8 kHz
Frame size	240 samples
Frame rate	33.33 frames/second
Subframe size	60 samples (4 subframes/frame)
Pitch Lag (8,6,8,6)	28 bits
Pitch Gain (5,5,5,5)	20 bits
Spectrum (3,4,4,4,4,3,3,3,3,3)	34 bits
Excitation Index (9,9,9,9)	36 bits
Excitation Gain (5,5,5,5)	20 bits
Other (error protection etc)	6 bits
Total	144 bits/frame
Bit Rate	$144 \times 33.33 = 4800$ bits/sec

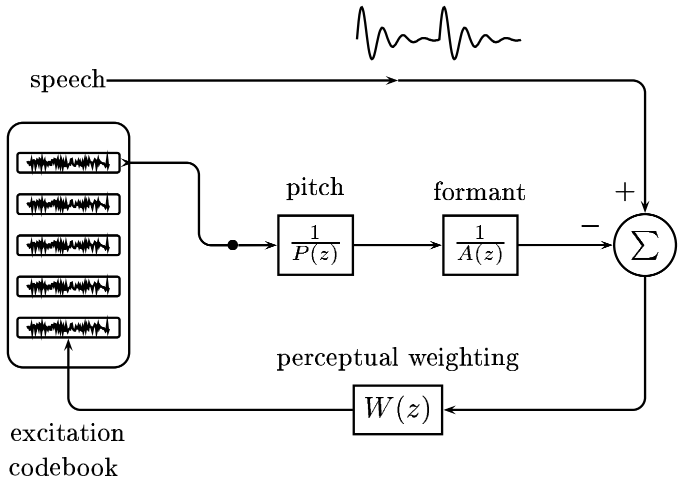

Figure 4.7: Analysis-by-synthesis using vector quantization of the excitation for the synthesis filter. This use of vector quantization is quite distinct from vector quantization of the short-term spectrum.

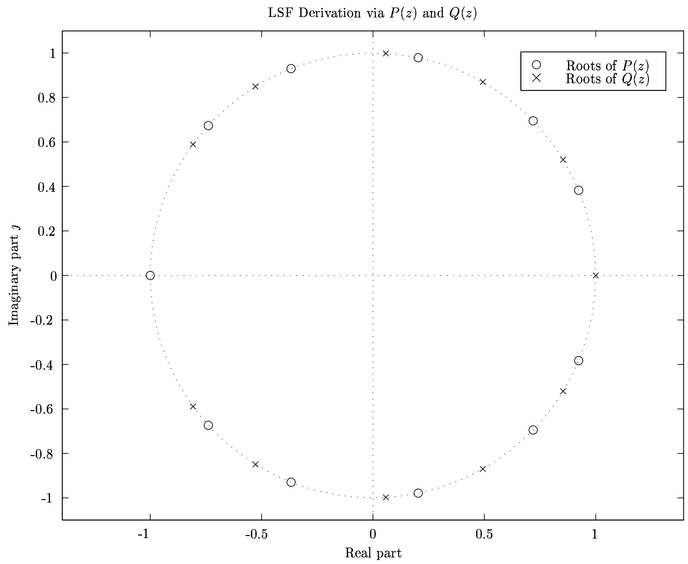

Figure 4.8: The interleaving of the roots of the polynomials $P(z)$ and $Q(z)$.

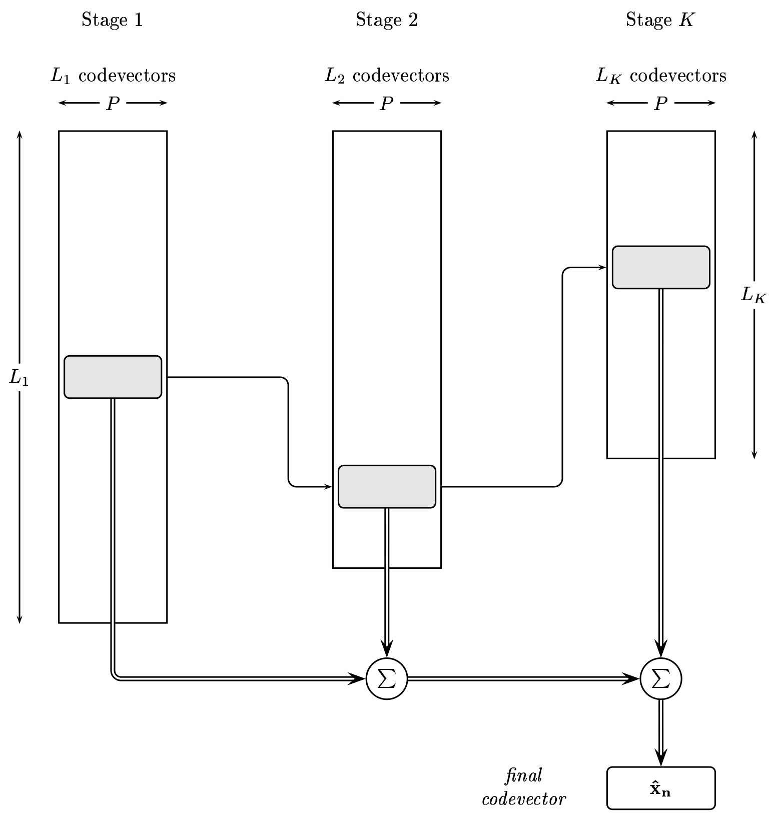

Figure 4.9: A multistage vector quantizer. Several separate codebooks are combined using addition to produce the final codevector. The number of vector elements P in each codebook is identical, and equal to the size of the final reconstructed codevector. Note that the number of entries in each codebook L_K need not be equal for each stage.

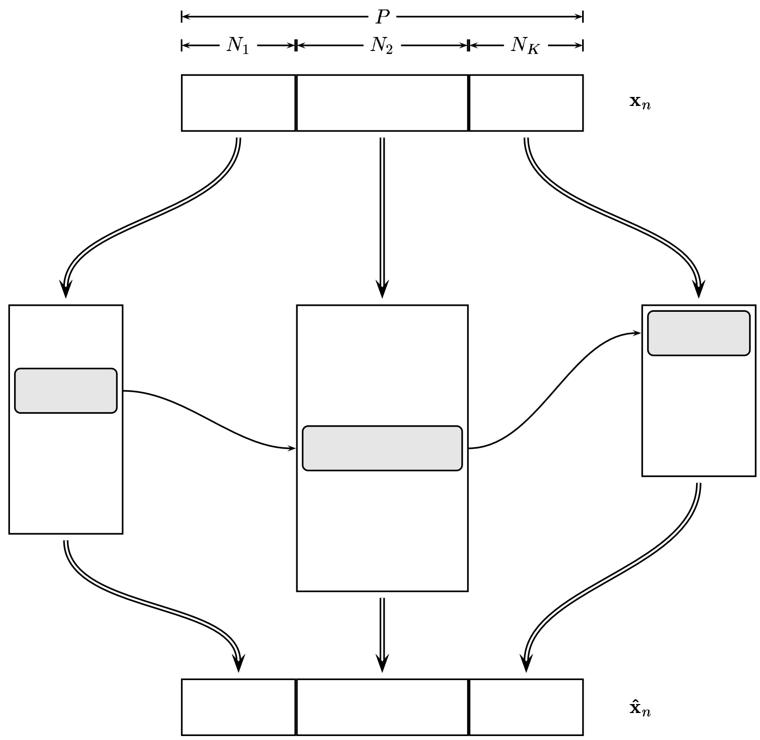

Figure 4.10: A split vector quantizer. Several separate codebooks are combined by concatenation of the sub-vectors to produce the final codevector.

List of Figures

List of Tables