1

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No. 2)

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

A Low-Delay Backward-Adaptive CELP Speech Coder

by

Leonardo G. Neumeyer

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

Ottawa-Carleton Institute for Electrical Engineering
Faculty of Engineering
Department of Systems and Computer Engineering
Carleton University

August, 1990

© Leonardo G. Neumeyer, 1990

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

ISBN 0-315-60461-1

The undersigned recommend to the Faculty of
Graduate Studies and Research acceptance of the thesis

**“A Low-Delay Backward-
Adaptive CELP Speech Coder”**

submitted by
Leonardo G. Neumeyer,
Electronic Engineer
in partial fulfillment of the requirements
for the degree of Master of Engineering

Thesis Supervisor and Chairman, Department
of Systems and Computer Engineering

Carleton University

August, 1990

Abstract

Code-excited linear prediction has become one of the most popular techniques for coding speech at bit-rates below 10 Kb/s. It uses an analysis-by-synthesis search to vector quantize the residual of the LPC model. In this study, we investigate different aspects of a CELP coder design at 8 Kb/s. A complete design procedure is presented. We also introduce a modification of the original CELP configuration that reduces the encoding delay by 80%. To accomplish that, a backward-adaptive short-term predictor is used. To control filter tracking, a novel mixed distortion measure is proposed. This distortion mixes noise-weighted mean-squared error with the log-likelihood distortion measure. Formal listening tests, based on mean opinion score measure, have shown that speech quality in the proposed and conventional coders are very similar (approximately 3.4 in a 5-point quality scale).

Acknowledgments

I would like to thank my thesis supervisor Dr. Samy Mahmoud for his guidance and encouragement during the course of this research.

I would also like to express my gratitude to Wilf LeBlanc for all his assistance, invaluable discussions, and for providing me with an extensive DSP software library.

I am grateful to Carleton University for the financial support that made my graduate studies possible.

Table of Contents

Abstract	iii
Acknowledgments	iv
List of Figures	vii
List of Tables	ix
List of Acronyms	x
List of Symbols	xii
Chapter 1 Introduction	1
1.1 A Historical Perspective	1
1.2 Digital Coding of Speech	2
1.3 Thesis Objectives	8
1.4 Thesis Organization	9
Chapter 2 Speech Production Models	10
2.1 Introduction	10
2.2 Speech Production Models	11
2.2.1 The Articulatory Model	11
2.2.2 The Digital Model	12
2.2.2.1 Excitation	13
2.2.2.2 Vocal Tract	13
2.2.2.3 Radiation	16
2.2.2.4 Complete Model	18
2.3 Linear Prediction of Speech	18
2.3.1 Formulation	18
2.3.2 LPC Block Algorithms	20
2.3.3 LPC Recursive Algorithms	23
2.3.4 The Long-Term Predictor	27
2.3.5 The Linear Prediction Model	27
2.4 Distortion Measures for Speech Processing	28
2.4.1 Signal-to-Noise Ratio	30
2.4.2 Log-Likelihood Ratio Distortion Measure	31
2.5 Speech Coding	33
2.5.1 Linear Prediction Coders	33
2.5.2 Forward vs Backward Adaptation	35

Chapter 3 Code Excited Linear Prediction Coders	39
3.1 Introduction	39
3.2 Design	40
3.2.1 Organization	42
3.2.2 Search Procedures	43
3.2.3 Complexity Reduction	51
3.2.3.1 Toeplitz Approximation	51
3.2.3.2 Shift Symmetric Codebooks	51
3.2.3.3 Sparse Codebooks	52
3.2.3.4 Summary	53
3.2.4 Long-Term Predictor	53
3.2.5 Excitation Codebook Optimization	56
3.2.6 Quantization of the LPC Parameters	59
3.2.6.1 Scalar Quantization of $A(z)$	59
3.2.6.2 Vector Quantization of $B(z)$	61
Chapter 4 Low-Delay Speech Coders	64
4.1 Backward Adaptive Coders at 16 Kb/s	64
4.2 A New Low-Delay 8 Kb/s Backward-Adaptive CELP Coder	66
4.2.1 Overview	66
4.2.2 The Mixed Distortion Measure	67
4.3 Computer Simulations and Results	71
Chapter 5 Summary and Suggestions for Future Research	82
5.1 Summary and Conclusions	82
5.2 Suggestions for Future Research	84
Appendix A Subjective Evaluation of Speech Quality	86
A.1 Introduction	86
A.2 Absolute Category Rating Test Method	87
A.2.1 Preparation of Source Tapes	87
A.2.2 Processing of the Stimulus Tapes	89
A.2.3 Testing Procedure	89
A.2.4 Presentation of the Results	90
References	92

List of Figures

Figure 1.1	Telephone circuits with echo cancellation. The delay is due to the speech coder and the communications channel.	6
Figure 2.1	Multi-tube model of the vocal-tract	15
Figure 2.2	Signal flow graph for lossless three-tube model of the vocal tract	17
Figure 2.3	Digital model of speech production	19
Figure 2.4	Linear prediction analysis/synthesis structure.	29
Figure 2.5	Computation of the residual energies in the log-likelihood ratio distortion measure.	32
Figure 2.6	Linear prediction speech synthesizers	34
Figure 2.7	Generic forward-adaptive LPC codec	37
Figure 2.8	Generic backward-adaptive LPC codec	38
Figure 3.1	CELP Encoder	41
Figure 3.2	The weighting filter is moved	46
Figure 3.3	Zero-input, zero-state separation	47
Figure 3.4	Simplified search	48
Figure 3.5	Update of the zero-input synthesis filters	50
Figure 3.6	Number of multiply-adds in search procedures	55
Figure 3.7	Pitch predictor optimization	57
Figure 4.1	Backward-adaptive CELP encoder.	68
Figure 4.2	Backward-adaptive CELP decoder.	69
Figure 4.3	Noise-weighted signal-to-noise ratio versus η	77
Figure 4.4	Noise-weighted signal-to-noise ratio for 15 iterations of the GLA algorithm.	78
Figure 4.5	Log-likelihood ratio versus η	80
Figure 4.6	% of suboptimum vectors versus η	81
Figure A.1	Distorted speech in terms of MNRUs.	88

Carleton University
Ottawa, Canada K1S 5B6

Text complete; leaf **viii** omitted in numbering.

List of Tables

Table 3.1	Complexity and storage requirements in search procedures	54
Table 4.1	Bit allocation and corresponding bit rate. The vector dimension is 26 samples and the sampling rate is 8 Khz.	70
Table 4.2	Bit allocation scheme for the forward-adaptive CELP coder. The frame size is 160 samples and the subframe size is 40 samples.	73
Table 4.3	Mean opinion score test.	75
Table A.1	Five-point quality scale.	91

List of Acronyms

ADPCM	- adaptive differential pulse code modulation
AM	- amplitude modulation
APC	- adaptive predictive coding
BA-CELP	- backward-adaptive CELP
b/s	- bits per second
CCITT	- international consultative committee for telephone and telegraph
CELP	- code-excited linear prediction
Codec	- encoder/decoder
dB	- decibel
DSP	- digital signal processor
FM	- frequency modulation
GLA	- generalized Lloyd algorithm
Hz	- Hertz
LAR	- log area ratio
LLR	- log likelihood ratio
LMS	- least mean square
LPC	- linear predictive coding
LSP	- line spectrum pairs
MFLOPS	- million floating point operations per second
MNRU	- modulated noise reference unit
MOS	- mean opinion score
ms	- milliseconds
MSE	- mean squared error
NWMSE	- noise weighted MSE
NWSNR	- noise weighted SNR
PCM	- pulse code modulation
RLS	- recursive least squares

SEGSNR	- segmental SNR
SNR	- signal to noise ratio
Vocoder	- voice coder
VQ	- vector quantization
zi	- zero input
zs	- zero state

List of Symbols

α	- residual energy
η	- mixing factor
ν	- no of pitch filter taps = $2\nu+1$
$\nabla_{\mathbf{a}}\alpha$	- gradient of α with respect to $\mathbf{a}$
$\mathbf{a}$	- long-term predictor coefficient vector
A_k	- cross-sectional area
$A(z)$	- formant filter
$\mathbf{b}$	- short-term predictor coefficient vector
$B(z)$	- pitch filter
$\mathbf{d}$	- synthesized residual vector
$\mathbf{D}$	- synthesized residual matrix
$d(n)$	- residual after pitch filter
d_{mix}	- mixed distortion
$e(n)$	- residual sequence
$E(z)$	- glottal filter
G	- gain
H	- entropy
$\mathbf{H}$	- impulse response matrix
$H(z)$	- speech production model
K	- subframe size
L	- codebook size
M	- pitch period
N	- frame size
n_c	- number of candidates
p	- short-term predictor order
Q	- attenuation factor in the MNRU
$\mathbf{r}$	- autocorrelation vector
$\mathbf{R}$	- autocorrelation matrix

r_0	- energy
$R(z)$	- radiation filter
S	- shifting matrix
$s(n)$	- speech sequence
t	- shifting factor
v	- excitation vector
$V(z)$	- vocal tract filter
$W(z)$	- weighting filter

Chapter 1

Introduction

1.1 A Historical Perspective

Before his son invented a machine capable of transmitting the speech wave through a pair of metallic wires, Alexander Melville Bell developed a universally applicable phonetic alphabet [1]. He called this system *Visible Speech*. Many other linguists had tried to write the universal alphabet by *analyzing* the way sounds were produced. Bell, instead, placed his vocal organs in systematically different positions and identified the sound produced in each case. By doing this extensive search he found the fundamental configurations and discarded the others. He needed 15 years, a systematic method, a fine ear and, probably, a lot of patience, to design a 34-symbol alphabet capable of describing almost any human sound. To test his invention, Bell taught his sons how to convert visible speech symbols into sounds. As another part of the experiment he asked a linguist to dictate for him words in all kind of exotic languages, dialects and accents. Melville's sons neither knew the languages nor attended the dictation sessions. However, they were able to decode visible speech format back to sounds in a slow, but faithful manner.

Bell's success was due to the idea of analyzing human speech by producing (or synthesizing) sounds in a systematic way. One century later, we are still applying the same basic principle. In modern speech coding research, we use digital instead of articulatory systems and codebooks instead of alphabets. Our objectives and motivations have slightly changed but the systematic and scientific method remain unchanged. An intriguing parallelism between most popular low bit-rate narrowband speech coders and Bell's work is the fact that both use an analysis-by-synthesis technique. The last generation of LPC digital speech coders use a large dictionary of vectors to produce synthetic speech. The encoder systematically searches for the vector that produces the best waveform and discards the others. Previous approaches, generally analyze the speech signal and extract its important features. The analysis method, although simpler, failed to satisfy quality standards at low bit-rates. In the last few years, almost all high-quality, low bit-rate speech coders are adopting the analysis-by-synthesis approach. This thesis is concerned with the realization of a version of this approach that satisfies specific requirements.

1.2 Digital Coding of Speech

Modern telecommunications systems such as telephone, radio, and satellite networks are being transformed into a new generation based on digital transmission systems. Thus, they require a digital representation of the speech wave. A discrete-time, discrete-amplitude representation will lead to economical, robust and secure transmission of the speech signal. Digital signals are easy to store, error-protect, encrypt, compress, multiplex and packetize. They can exploit the availability of reliable, low-cost microprocessors and integrated circuit technology. Moreover, the dramatic advances in digital signal processor technology is making computational and memory complexity a less critical problem.

In practice, the digitization of the speech wave introduces irreversible distortion. The Nyquist theorem [2] states that it is possible to sample a bandlimited signal without losing information by selecting a sampling frequency that is greater than twice the maximum frequency component of the signal. As a result, time discretization of a bandlimited signal is an information-lossless process. However, narrowband digital speech systems, such as those used in public telephone networks, are designed to work on the lower, and perceptually more important, range of the speech spectrum. Throughout this work we will use a standard 8 Ksamples/s sampling rate that limits the frequency spectrum to 4 KHz . Broader bandwidths are used in digital audio. This includes AM radio ($50\text{ Hz} - 7\text{ KHz}$), FM radio ($20\text{ Hz} - 15\text{ KHz}$) and compact disks ($10\text{ Hz} - 20\text{ KHz}$). On the other hand, amplitude discretization (or quantization) is always an information-lossy process. To minimize the quantization noise, a common practice is to exploit the statistics of the signal amplitude. This is done by enhancing the resolution at amplitude levels that are more likely to occur. Standards have been developed based on this approach, including the *A-law* and *μ -law* non-uniform quantizers [3].

The main disadvantage of digital representation is that signal transmission demands greater bandwidth than that required by analog representation. To make the digital signal spectrally efficient it is necessary to compress the information. By coding the speech signal, we try to achieve the minimum possible distortion for a given transmission rate. There are only two possible ways of reducing the transmission rate:

1. Some information is discarded and therefore lost. The speech coder eliminates irrelevant information that is not perceptible to the human ear.
2. Redundant information is removed at the encoder and placed back at the decoder.

If these two techniques were applied in a perfect way, there would be no coding distortion. However, practical systems are far from perfect and are subject to a number of constraints. A practical speech coding system can be characterized by the following

attributes:

- Bit rate.
- Speech quality.
- Encoding delay.
- Sensitivity to channel errors.
- Computational complexity.
- Memory complexity.

Bit rate. The common standard in digital telephony uses a transmission rate of *64 Kb/s* for a single voice channel. The *T1* carrier, for example, was designed to transmit 24 speech channels of *64 Kb/s* each [4]. Presently, submultiples such as 32, 16, and 8 *Kb/s* are multiplexed in a single *64 Kb/s* channel. Other practical bit-rates, evolved from data communications over telephone lines, are 9.6, 4.8, and 2.4 *Kb/s*. Very low bit-rates such as 2.4 *Kb/s* are used in tactical speech communications. At this rate, it is possible to encrypt the speech sequence and send it through a 4 *KHz* bandwidth telephone network or radio links. In such a system, speech quality is very poor; intelligibility is maintained at the expense of naturalness [3]. Low bit-rates in the order of 4 *Kb/s* are also a target in military and government applications such as secure telephony. Other applications where bandwidth is critical are cellular radio, mobile satellite communications (MSAT), international maritime satellite communications (INMARSAT) and storage. The second generation of cellular radio, for example, uses an 8 *Kb/s* speech coder for a total bit rate of 14 *Kb/s* including error protection. The bandwidth for speech channel will be reduced by a factor of three compared to the current 30 *KHz* FM system.

Speech quality. Speech quality is generally evaluated in terms of intelligibility and naturalness. At low bit-rates, objective measures such as the SNR are inaccurate. This is because too much irrelevant information has been dropped. Perceptual information is

not always visible in the waveform. As a result, more meaningful subjective distortion measures are necessary. Reasonable measurements of speech quality can only be obtained by conducting psychoacoustical studies such as the MOS test which is described in Appendix A.

Encoding delay. The encoding delay may become a major problem in two-way digital communications systems (not in broadcasting or storage applications). To compress the data, low bit-rate speech coders introduce a considerable amount of delay. This is in addition to the delays caused by international satellite communications, error-protection algorithms and so on. An excessive total delay may degrade the communication process in two different ways:

1. By causing annoying echoes.
2. By severely disrupting the quality of two-way voice communication between end users.

The echo may be generated at the hybrid transformers used for two-wire-to-four-wire conversion (figure 1.1). The hybrid converts the subscriber's single-path to a two-way-path. When the hybrid is not perfectly balanced, part of the energy is reflected back to the source producing the echo. For very low delays, the echoes are tolerable. According to CCITT standards some kind of echo cancellation is required when the two way delay exceeds 50 ms. The annoying effect of the echo can be reduced by using *half-echo suppressors* or *echo-cancellers* [5, 6]. Subjective studies show that even without echo 50% of the subscribers have problems when the two-way delay is 400 ms and 40% have problems with delays of 300 ms. Satellite communications already exceed the 300 ms boundary and mobile-to-mobile links may have a delay of 520 ms. Thus, more research effort is necessary to avoid additional delays.

Figure 1.1 Telephone circuits with echo cancellation. The delay is due to the speech coder and the communications channel.

Sensitivity to channel errors. Real communications channels are corrupted by receiver noise and other impairments. In digital communications, the effect of this noise is characterized by the probability of making an erroneous decision at the receiver. Random bit error probabilities can range from 10^{-6} in high quality channels to 10^{-2} in noisy channels. In some applications such as mobile communications errors can occur in bursts, making the problem even more severe. Forward error correction can reduce the bit error rate of the digital channel at the expense of bandwidth. Thus, speech coding algorithms have to be robust to channel errors. It is evident that as bit-rate decreases, the coder is required to eliminate more redundancy, and in the process the decoding decisions become more sensitive to channel errors. However, the perceptual effects of channel errors depend also on the structure of the algorithm. For example, if the decoder uses past data to reconstruct the speech sequence, it may propagate the error to future blocks of samples. Thus, the decoder's memory of past information must be limited in order to improve its robustness and reduce error propagation.

Complexity. Most speech coding applications require real-time computing facilities. During the last 10 years a new generation of programmable digital signal processors has been developed. DSPs are specialized microcomputers for real-time number crunching [7]. They differ from conventional microprocessors in several architectural aspects. A fast multiplier/accumulator is integrated into the data path. Independent memories are used for data and instructions. Parallel function units, hardwired design instead of microprogramming, and pipelining are some of the features. Commercial DSPs are capable of performing a floating-point multiply and add every 60-80 ns (i.e. 25-33 MFLOPS). Floating-point and fixed-point arithmetic are available in most DSP families. However, at present, floating-point DSPs are more expensive and slower than fixed-point versions by a factor of nearly 50% [7]. Hence the problem of precision and quantization

noise that must be addressed in the design of the algorithms is targeted for fixed-point DSPs.

1.3 Thesis Objectives

In section 1.2 we reviewed the main problems associated with digital coding in voiceband applications. The theoretical framework can be found in a branch of information theory known as *Rate-Distortion Theory* [8, 9, 10, 11]. However, speech coder design is often based on heuristic and intuitive approaches. This is because many variables are involved and the performance in terms of speech quality is highly subjective. Therefore, until a mathematical model for speech perception is developed it will not be possible to find a perfect speech coding scheme.

As mentioned earlier, coders that combine analysis-by-synthesis techniques with linear prediction coding have become very popular during the last few years. The first objective of this thesis is to survey the latest techniques used in code-excited linear prediction (CELP) coders in order to simulate a complete design of the system. The emphasis will be focused on the encoding delay. The main objective of this thesis is to examine the feasibility of modifying the conventional CELP structure in order to achieve shorter delays. The bit-rate will be in the order of 8 Kb/s. At this transmission rate, conventional CELP designs introduce a one-way delay of 40-60 ms [12]. This thesis will examine the design, simulation and testing of a version of the CELP coding algorithm that reduces this delay by approximately 80%. The basic approach consists in the use of backward adaptation to update the LPC parameters and the introduction of a new distortion measure to prevent filter mistracking. To the best of the author's knowledge this thesis represents the first attempt toward the realization of a partial backward-adaptive version of CELP at 8 Kb/s. Other aspects such as complexity and sensitivity to channel errors will be discussed. The coder will be evaluated in terms of delay and speech quality. To accomplish this, a conventional CELP coder will be

used as a reference. The total delay can be estimated from the design. Assessment of speech quality requires subjective tests which will be conducted according to standard CCITT recommendations [13].

1.4 Thesis Organization

This thesis can be divided in two main parts. The first part, (Chapters 1-3) introduces the reader in the design of a state-of-the-art CELP codec. The second part (Chapters 4-5) focuses on the theory and evaluation of the proposed backward-adaptive CELP codec.

Throughout Chapter 1, we discuss in general the main requirements and characterization of speech coding based on digital methods. Chapter 2 introduces several speech production models. In the last part of this chapter we present different aspects related to linear prediction of speech. In Chapter 3, we concentrate on CELP coders. The design process is presented in great detail. Complexity reduction methods and optimization procedures are also addressed. Chapters 2 and 3 then serve as detail background material. Chapter 4 is devoted to the delay problem and possible approaches for delay reduction. A novel backward-adaptive CELP design is presented. In the same chapter we compare the performance of the low-delay coder with a conventional CELP coder. Finally, Chapter 5 presents a summary of the results of this thesis along with suggestions for future research.

Chapter 2

Speech Production Models

2.1 Introduction

The natural speech communication process takes place within a short distance between the speaker's head and the listener's ear through the molecules of air. Although the physical process is essentially continuous, the communication process is discrete. To communicate we use a language formed by discrete words, sentences and structures studied in the field of *linguistics*. The basic linguistic element is called the *phoneme*. Each language has a set of phonemes (20-50) to describe its sound [14]. A word is a combination of these abstract representations of the human sounds. The physical realization of a phoneme is the *phone*. A phone may have an infinite acoustic variation depending on the speaker and the influence of neighboring phones. To make the communication process intelligible the listener has to recognize only the corresponding phonemes of the message. However, in order to receive a clear and natural spoken message, the listener needs much more information.

One of the basic questions in speech communication is how much information is in the speech signal. The lower bound could be the information contained in the phonemes. In English, for example, there are 42 phonemes with a probability of occurrence

$P(x_i)$ [14]. The entropy of the code is $H = -\sum P(x_i) \log_2 P(x_i)$ bits. Assuming independence among phonemes, the average information of the English phonemes can be represented by 5 bits. The normal rate in conversational speech is 10 phonemes/sec. Therefore the written equivalent of the rate of information is 50 bits/sec. On the other hand, we can find an upper limit considering a sampled signal of the speech wave. Considering a bandwidth of 3 KHz and a 6 bit quantizer, the capacity of the human channel is 36000 bits/sec. This is more than 600 times the lower limit [14]. The bit rate of interest depends on the application. In speech recognition low bit rates are enough to recover the phonetic information while in speech coding a representation of the physical wave is to be transmitted or stored and therefore higher bit rates are necessary.

2.2 Speech Production Models

2.2.1 The Articulatory Model

The human speech production system shares its elements with the respiratory, masticatory and olfactory apparatus. This fact makes the speech machinery very complex and "inefficient" in a communications sense. As a result, the speech wave has a lot of redundant information that can be removed in the transmission process without affecting the human perception of the message.

The human speech production system can be modeled using acoustic tubes of non-uniform cross-sectional areas. The vocal tract tube begins at the vocal cords (glottis) and ends at the lips (typical length of the vocal tract is approximately 17 cm for an adult male). A secondary tube, the nasal tract, is connected to the vocal tract through the velum and is terminated at the nostrils. The cross-section of the vocal tract is shaped by the articulators, namely the lips, jaw, tongue, velum, teeth and vocal cords [14, 15]. This model is quasi-stationary during short periods of time and can be interpreted as a time-invariant linear filter that amplifies and attenuates certain frequencies. The peaks

of the spectra or “formants” are the resonances of the acoustic tube corresponding to the poles of the filter.

Another important component of this model is the excitation caused by the flow of air generated at the sub-glottal system. The excitation is controlled by the articulators and can produce three different types of sounds.

- Voiced sounds (e.g. /a/, /i/, /æ/.) are originated at the glottis. The vocal cords are tense and in oscillation producing quasi-periodic waves. These signals are generally large with important low-frequency components. The spectrum has energy concentration at multiples of the fundamental frequency of the vocal cord vibration.
- Unvoiced or fricative sounds (e.g. /s/, /sh/, /f/.) are produced by turbulent flow of air caused by a small constriction at some place of the vocal tract. These noise-like sounds are weak and have a relatively broad and flat spectrum.
- Plosive sounds (e.g. voiced: /b/, /d/, /g/; unvoiced: /p/, /t/, /k/.) are the result of a pressure buildup at some point of closure and an abrupt release. These sounds may have voiced or unvoiced excitation and can be modeled as a step function; therefore their spectrum falls inversely with frequency.

Speech is the result of voluntary movement of articulators that rarely stay fixed for more than 40 ms [15]. Transitions between sounds are generally gradual. These characteristics will permit us to model the speech production system using linear system theory in a relatively simple and mathematically tractable way.

2.2.2 The Digital Model

Although a complete study of the acoustic theory of speech production is beyond the scope of this thesis, a general introduction is convenient in order to validate the use of an all-pole digital filter as a model of the vocal tract. A complete model of speech production must include the excitation, the vocal tract and the radiation effects. These

components of the system can be considered acoustically independent. The acoustic interaction between them is in most cases secondary.

2.2.2.1 Excitation Unvoiced excitation can be modeled by random noise with gaussian amplitude and flat spectrum. Therefore the spectrum of the speech signal will be dominated by the vocal tract frequency response.

Unlike unvoiced speech, voiced sounds play an important role in speech perception. Sonorants are excited by air pulses formed at the glottis. The shape of these pulses is similar to a half-rectified sine wave in which the closing phase is more rapid than the opening phase [15]. This source is generally modeled by a train of pulses exciting a glottal filter. The impulse response of the filter has the shape of a glottal pulse. In the frequency domain, the filter has a very low cutoff frequency and a fall off of -12 dB/octave. The spectrum of voiced speech is shaped by both the vocal tract and the glottal filters.

2.2.2.2 Vocal Tract In order to obtain the parameters of the vocal tract filter several simplifications have to be made. Very simple cases can be solved assuming planar propagation of the speech wave and applying the laws of physics, including conservation of mass, momentum and energy. A 17-cm-long tube with uniform cross-section filled with a zero-viscosity gas and non-elastic walls leads to a simple solution of the wave equations. When excited with a train of air pulses, the tube produces a sound similar to the neutral /UH/. The transfer function of this filter is $V(j\Omega) = 1/\cos(\Omega l/c)$ where l is the length of the tube, c is the speed of sound and Ω the frequency. In the frequency domain there are an infinite number of poles located on the imaginary axis at the frequencies $F_i = (2i - 1)c/4l = 500, 1500, 2500, \dots \text{Hz}$ [15]. These are the formants or resonances of the uniform lossless tube.

More complex shapes of the vocal tract can be approximated using more than one tube. The wave equation can be solved for the multi-tube case using the same assumptions of the one-tube model and applying the boundary conditions at the junctions of the tubes. In this case multiple reflections take place at the junctions. There is a direct analogy between the acoustic system and electric transmission lines.

Consider the multi-tube configuration shown in figure 2.1. There are N tubes of length Δx . The total length is l and the delay in one tube is $\tau = \Delta x/c$. Due to multiple reflections, the impulse response of this filter is a sequence of pulses. The first pulse of the response will appear at the output after an initial delay of $N\tau$, that is the time required by the wave to travel along the tubes. The rest of the pulses will come at a rate of 2τ , that is the time between two consecutive reflections at the boundaries of one section of the multi-tube model. Therefore the impulse response is,

$$v(t) = \delta(t - N\tau) + \sum_{k=1}^{\infty} \alpha_k \delta(t - N\tau - 2k\tau) \quad (2.1)$$

Since the response to an impulse is a sequence of impulses spaced with a fixed time, we can use a discrete-time variable. Note that we can define $t = 2n\tau$ and $T_s = 2\tau$ where T_s is the sampling frequency. Therefore,

$$v(n) = \sum_{k=0}^{\infty} \alpha_k \delta(n - N/2 - k) \quad (2.2)$$

which is the response of a discrete-time filter plus a fixed delay of $N/2$. In most practical cases the delay can be ignored without affecting the problem. The number of tubes for a given total length depends on the desired bandwidth of the speech signal, applying the sampling theorem [16],

$$B = \frac{1}{2T_s} = \frac{1}{4\tau} = \frac{Nc}{4l} \simeq \frac{N}{2} \text{ KHz} \quad (2.3)$$

As a result, a speech bandwidth of 4 KHz for an adult male requires a cascade model of 8 tubes.

Figure 2.1 Multi-tube model of the vocal-tract

A signal flow graph for three lossless tube model of the vocal tract is shown in figure 2.2, where,

$$r_k = \frac{A_{k+1} - A_k}{A_{k+1} + A_k} \quad (2.4)$$

are the reflection coefficients and A_k is the cross-sectional area of the k^{th} tube. For positive areas the range of the reflection coefficients is $-1 \leq r_k \leq 1$. The loading effects at the glottis and lips are included in r_L and r_G . The resulting transfer function of the vocal tract is,

$$V(z) = \frac{0.5(1 + r_G)z^{-N/2} \prod_{i=1}^N (1 + r_i)}{1 - \sum_{i=1}^N \alpha_i z^{-i}} \quad (2.5)$$

Poles generally occur in complex conjugate pairs that produce the formants. There are no zeros other than those at the origin which represent the fixed delay. As a consequence, we can say that the lossless multi-tube model of the vocal tract is a time-varying all-pole filter. There is a physical explanation for the absence of zeros. In the multi-tube model there is only a direct path between the source and the output, consequently there are no acoustic cavities that may trap the energy of the wave. This is not the real case for nasal or fricative sounds. Therefore to produce nasals and fricatives we must include zeros in the transfer function or achieve the same effect by including more poles [17].

2.2.2.3 Radiation The last part of the speech production process is the radiation at the lips (or at the nostrils for nasal sounds). It can be shown [14] that the radiation effect is like a highpass filter operation. Therefore a simple approximation is obtained using a first backward difference,

$$R(z) = R_0(1 - z^{-1}) \quad (2.6)$$

This corresponds to a +6 dB/octave spectrum.

Figure 2.2 Signal flow graph for lossless three-tube model of the vocal tract

2.2.2.4 Complete Model A complete discrete-time articulatory model of the speech production system must include all the previously discussed effects. A block diagram of the digital model of speech production is shown in figure 2.3. The shape of the spectrum is mainly defined by $V(z)$. The global trend in voiced speech is shaped by $E(z)R(z)$ with a spectrum of -12 dB/octave due to the glottal source plus 6 dB/octave due to the radiation at the lips resulting in a total -6 dB/octave falloff. In noisy speech the global spectrum is flat. Sometimes, it is convenient to integrate all the effects in a single transfer function,

$$H(z) = E(z)V(z)R(z) = \frac{G}{1 - \sum_{i=1}^p a_i z^{-i}} = \frac{G}{A(z)} \quad (2.7)$$

where p is the order of the filter and G is a constant.

2.3 Linear Prediction of Speech

Linear prediction is one of the most popular speech analysis techniques. In section 2.2.2 we discussed how an all-pole filter can approximate the human speech production mechanism. In the articulatory model the coefficients of the filter are related to the cross-sectional areas of the vocal tract. However, in speech analysis, we need a method to calculate the parameters of the model using the speech signal. A simple, straightforward, well understood, inexpensive solution of the all-pole filter is based on linear prediction techniques [18].

2.3.1 Formulation

The fundamental problem is to find the values of the parameters of the all-pole transversal filter using the speech signal. Assuming an all-pole model, a sample of speech is a linear combination of past samples plus an excitation term,

$$s(n) = \sum_{k=1}^p a_k s(n-k) + Gu(n) \quad (2.8)$$

Figure 2.3 Digital model of speech production

where $u(n)$ is the excitation sequence and p is the order of the filter. Since the excitation is not known we can only express a speech sample in terms of past samples,

$$\tilde{s}(n) = \sum_{k=1}^p \tilde{a}_k s(n-k) \quad (2.9)$$

$\tilde{s}(n)$ is the predicted sample and the $\tilde{a}_k$'s are the estimation of the filter coefficients. The error of the estimation is,

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^p \hat{a}_k s(n-k) \quad (2.10)$$

where $\hat{a}_0 = 1$ and $e(n)$ is the prediction error, generally called the *residual*. In the statistical literature it is called the *innovation* sequence because it contains the unpredictable part of the input signal. In the z domain,

$$\frac{E(z)}{S(z)} = 1 - \sum_{k=1}^p \tilde{a}_k z^{-k} = A(z) \quad (2.11)$$

$A(z)$ is called the *inverse* or *prediction error* filter. Note that if $\tilde{a}_k = a_k$ for all k then $e(n) = Gu(n)$ and $A(z)$ is the inverse filter of the model $H(z)$ in equation 2.7. Throughout this thesis, we will also refer to $A(z)$ as the *formant filter*. The optimum estimate of the a_k 's depends on the criteria used to minimize the prediction error.¹

2.3.2 LPC Block Algorithms

The excitation of the speech production model can be represented by either a deterministic train of pulses for sonorants or random noise for fricatives. This simplified model of the excitation sequence is used to solve the LPC problem. Due to the duality between a deterministic impulse and white noise, identical results are obtained whether statistical expectations or time averages are considered [18]. In this section we consider two deterministic solutions based on the method of *least squares*. This method can be viewed as the deterministic counterpart of Wiener filter theory [19, 20].

¹ In the remainder of this dissertation the tilde of the $\tilde{a}_k$'s will be dropped. It should be noted that the coefficients $\tilde{a}_k$'s are estimates.

In the least squares method, the objective is to find a set of parameters that minimizes the square error over a segment (or block) of speech samples,

$$\alpha = \sum_n e^2(n) \quad (2.12)$$

In matrix form,

$$e(n) = s(n) - \mathbf{a}^T \mathbf{s}(n) \quad (2.13)$$

where $\mathbf{a}^T = [a_1, a_2, \dots, a_p]$ and $\mathbf{s}^T(n) = [s(n-1), s(n-2), \dots, s(n-p)]$. Replacing $e(n)$ in equation 2.12 and leaving the limits of the summation undefined for the moment,

$$\begin{aligned} \alpha &= \sum_n \left[s(n)^2 - 2s(n)\mathbf{a}^T \mathbf{s}(n) + \mathbf{a}^T \mathbf{s}(n)\mathbf{s}^T(n)\mathbf{a} \right] \\ &= \sum_n s(n)^2 - 2\mathbf{a}^T \sum_n s(n)\mathbf{s}(n) + \mathbf{a}^T \left[\sum_n \mathbf{s}(n)\mathbf{s}^T(n) \right] \mathbf{a} \\ &= r_0 - 2\mathbf{a}^T \mathbf{r} + \mathbf{a}^T \mathbf{R} \mathbf{a} \end{aligned} \quad (2.14)$$

where r_0 is the energy of the speech segment, $\mathbf{r} = \sum_n s(n)\mathbf{s}(n)$ is the autocorrelation vector and $\mathbf{R} = \sum_n \mathbf{s}(n)\mathbf{s}^T(n)$ is the autocorrelation matrix. To minimize the error we take the gradient with respect to $\mathbf{a}$,

$$\nabla_{\mathbf{a}} \alpha = -2\mathbf{r} + 2\mathbf{R}\mathbf{a} = 0 \quad (2.15)$$

and thus we obtain the linear system,

$$\mathbf{R}\mathbf{a} = \mathbf{r} \quad (2.16)$$

replacing in equation 2.14 we get the minimum error,

$$\alpha_{min} = r_0 - \mathbf{a}^T \mathbf{r} \quad (2.17)$$

There are two approaches to define the limits of the summation in equation 2.12. In the *autocorrelation method* the speech segment is windowed. Samples outside the window are identically zero. We may thus express the error as,

$$\alpha = \sum_{n=-\infty}^{+\infty} e(n)^2 = \sum_{n=0}^{N+p-1} e(n)^2 \quad (2.18)$$

where N is the length of the window. Note that the error is expected to be larger at the p first samples of the window where speech samples are predicted from zero values. The same error occurs at the p last samples of the interval where zeros are predicted from non zero samples. For this reason windows that gradually decrease to zero at the ends of the interval are used. The elements of the correlation matrix $\mathbf{R}$ and the correlation vector $\mathbf{r}$ are given by,

$$r(|i-j|) = \sum_{n=0}^{N-1-|i-j|} s(n)s(n-|i-j|) \quad \begin{cases} i = 1, \dots, p \\ j = 0, \dots, p \end{cases} \quad (2.19)$$

and therefore the matrix is symmetric and Toeplitz. The system of linear equation can be solved using the efficient Levinson-Durbin recursion [17, 16, 10].

In the *covariance method* the error is computed over a finite segment as,

$$\alpha = \sum_{n=0}^{N-1} e(n)^2 \quad (2.20)$$

leading to a symmetric (but not Toeplitz) matrix. The elements of the correlation matrix and vector are given in this case by,

$$r(i, j) = \sum_{n=-j}^{N-j-1} s(n)s(n+j-i) \quad \begin{cases} i = 1, \dots, p \\ j = 0, \dots, p \end{cases} \quad (2.21)$$

The linear system, in this case, is generally solved using Choleski decomposition [17, 16, 10].

There are several aspects to consider before choosing the solution method of linear prediction. The computational complexity of the Levinson-Durbin algorithm,

for example, is p times smaller than Choleski decomposition. Storage requirements for the autocorrelation and covariance methods are similar.

For the autocorrelation method the inverse filter is guaranteed to be minimum phase leading to stable models. However, in numerical solutions, round-off noise and ill-conditioned correlation matrices may cause unstable filters. The round-off noise problem is solved using a large word length. Ill-conditioned matrices can be prevented by whitening the spectrum of voiced speech. This can be realized by pre-emphasizing the high frequencies of the spectrum with a first order filter [17]. A simple stability test can be performed using the reflection coefficients. The covariance method is more likely to produce unstable filters. To overcome this problem one can test for stability and reflect the poles inside the unit circle or simply use the modified covariance method (see [21] or appendix B in [10]). This method is a modification of the Choleski decomposition that guarantees the reflection coefficients to be between ± 1 and then calculates the filter parameters using the relation between a_k 's and r_k 's provided by the Levinson-Durbin recursion. The stability problem in the covariance method tends to decrease as the length of the speech segment increases. In the autocorrelation method the frame has to be long enough to reduce the distortion due to the tapering effect of the window.

2.3.3 LPC Recursive Algorithms

LPC solutions, such as the covariance and autocorrelation methods update the inverse filter in a block by block basis. Smooth variation of the LPC parameters is achieved by overlapping consecutive blocks of samples. These solutions are very popular in speech processing and have been used extensively in a wide range of applications. However, in certain speech coding designs, recursive (or sequentially adaptive) algorithms are preferable. Recursive algorithms continuously update the LPC parameters, i.e. in a sample by sample mode. The main advantage of this approach is that it does not introduce delay into the system. For some sequentially adaptive algorithms, complexity

is low compared to block algorithms. On the other hand, since the recursive algorithm involve feedback, they are subject to becoming unstable [19]. In addition to this, continuous adaptation of the LPC parameters does not present any advantage when this parameters have to be sent through a channel. This is because it would be very expensive to send LPC information in a sample by sample basis (more discussion about this in section 2.5.2).

The adaptive filtering operation involves two basic processes. One is the adaptive process in which the parameters of the filter are adjusted according to the adaptation algorithm. The other process is the filtering operation itself that produce the error (or residual) sequence. Some of the most popular recursive algorithms are based on adaptive transversal filters that use gradient vector estimation. As discussed before, the residual energy is given by,

$$\alpha(n) = r_0 - 2\mathbf{a}^T(n)\mathbf{r} + \mathbf{a}^T(n)\mathbf{R}\mathbf{a}(n) \quad (2.22)$$

where $\mathbf{a}(n)$ is the coefficient vector at iteration n . The function $\alpha = \mathcal{F}(\mathbf{a}(n))$ is a bowl-shaped surface (sometimes called error performance surface) with a unique minimum. Gradient algorithms seek the unique minimum of this p -dimensional function using an estimate of the gradient:

$$\mathbf{a}(n+1) = \mathbf{a}(n) + \frac{1}{2}\mu[-\nabla_{\mathbf{a}}\alpha(n)] \quad (2.23)$$

where μ is a positive constant called step size or weighting constant. In the method of *steepest-descent* the statistics of the signal are assumed to be stationary and known (i.e. $\mathbf{R}$, $\mathbf{r}$ and r_0 are known). A more practical approach is the *least-mean-square* (LMS) algorithm [22]. The LMS algorithm does not require a priori knowledge of the statistics of the signal. This makes it suitable for nonstationary environments. In addition to this, it does not require matrix inversion. As a result, it is simple to implement in a digital system. The gradient is estimated from the available data: $\mathbf{R}(n) = \mathbf{s}(n)\mathbf{s}^T(n)$

and $\mathbf{r}(n) = \mathbf{s}(n)s(n)$. Thus, replacing in equation 2.15 we obtain the gradient estimate given by,

$$\hat{\nabla}_{\mathbf{a}}\alpha(n) = -2\mathbf{s}(n)s(n) + 2\mathbf{s}(n)\mathbf{s}^T(n)\mathbf{a}(n) \quad (2.24)$$

replacing equation 2.24 in equation 2.23 leads to

$$\mathbf{a}(n+1) = \mathbf{a}(n) + \mu\mathbf{s}(n)\left[s(n) - \mathbf{s}^T(n)\mathbf{a}(n)\right] \quad (2.25)$$

which is the LMS algorithm. Despite the simplicity of the LMS algorithm, a statistical analysis of convergence is relatively complex. To make the analysis mathematically tractable, some assumptions about independence and gaussian distributions are necessary. This analysis is clearly developed in [19]. It can be shown that a necessary and sufficient condition for convergence of the average mean-squared error is,

$$0 < \mu < \frac{2}{\sum_{i=1}^p \lambda_i} \quad (2.26)$$

where the $\lambda_i, i = 1, \dots, p$ are the eigenvalues of the correlation matrix $\mathbf{R}$. Some practical systems use a time varying step size defined as,

$$\mu = \frac{K}{p(\text{input power})}, \quad K \leq 1 \quad (2.27)$$

The convergence of the LMS algorithm is governed by the step size μ , the number of taps p and the eigenvalues of the correlation matrix.

Further improvement in performance at the expense of large increase in computational complexity is possible by using the recursive least-squares (RLS) algorithm. It can be viewed as the deterministic counterpart of the Kalman filter theory [19]. The main advantage of the RLS algorithm is that it can use all the information contained in past data back to the initial iteration. As a result, the RLS algorithm converges faster

(one order of magnitude) than the LMS algorithm [19]. In the recursive solution of the least-squares method the objective is to minimize the accumulative error ϵ ,

$$\epsilon(n) = \sum_{i=1}^n \beta(n, i) |e(i)|^2 \quad (2.28)$$

where $e(i) = s(n) - \mathbf{a}^T(n)\mathbf{s}(i)$ and $\beta(n, i)$ is a weighting factor. The weighting factor can be used to *forget* distant data. This is necessary when the predictor is operating in a nonstationary environment. A common choice for the weighting factor is $\beta(n, i) = \lambda^{n-i}$, $i = 1, \dots, n$. Taking the gradient of the accumulative error with respect to $\mathbf{a}$ leads to the normal equation $\Phi(n)\mathbf{a}(n) = \Psi(n)$ where $\Phi(n)$ is a p -by- p correlation matrix and $\Psi(n)$ is a p -by-1 correlation vector given by,

$$\begin{aligned} \Phi(n) &= \sum_{i=1}^n \lambda^{n-i} \mathbf{s}(i) \mathbf{s}^T(i) \\ \Psi(n) &= \sum_{i=1}^n \lambda^{n-i} \mathbf{s}(i) s(i) \end{aligned} \quad (2.29)$$

Taking the terms corresponding to $i=n$ out of the summation we get the recursive relations:

$$\begin{aligned} \Phi(n) &= \lambda \Phi(n-1) + \mathbf{s}(n) \mathbf{s}^T(n) \\ \Psi(n) &= \lambda \Psi(n-1) + \mathbf{s}(n) s(n) \end{aligned} \quad (2.30)$$

To solve the normal equation for the coefficient vector $\mathbf{a}$, we have to determine the inverse of the correlation matrix. By using the matrix inversion lemma, we can get the recursive solution for $\mathbf{a}$ without computing the inversion [19]. As a result, the RLS algorithm for linear prediction is given by the following set of equations:

$$\begin{aligned} \mathbf{k}(n) &= \frac{\lambda^{-1} \mathbf{P}(n-1) \mathbf{s}(n)}{1 + \lambda^{-1} \mathbf{s}^T(n) \mathbf{P}(n-1) \mathbf{s}(n)} \\ \gamma(n) &= s(n) - \mathbf{s}^T(n) \mathbf{a}(n-1) \\ \mathbf{a}(n) &= \mathbf{a}(n-1) + \gamma(n) \mathbf{k}(n) \\ \mathbf{P}(n) &= \lambda^{-1} \mathbf{P}(n-1) - \lambda^{-1} \mathbf{k}(n) \mathbf{s}^T(n) \mathbf{P}(n-1) \end{aligned} \quad (2.31)$$

We said before that the computational complexity of the RLS algorithm is large. In the LMS algorithm the number of multiplications and additions per sample is in the order of $2p$. On the other hand the RLS requires approximately $3p^2 + p$ operations. The complexity can be reduced by using a fast algorithm. The *fast transversal filter* (FTF) [23], for example, requires $5p$ multiplications and $7p$ additions per sample to perform the linear prediction algorithm.

2.3.4 The Long-Term Predictor

A low order short-term predictor is capable of removing near sample redundancies of a speech signal. The spectrum of the model is a good fit of the global speech spectrum that corresponds to the formants of the vocal tract. However, the residual sequence of voiced speech is characterized by a sequence of pulses that correspond to the periodicity of the incoming signal. If we could, somehow, determine the pitch period M then we would be able to remove the *long-term* redundancies and increase the total prediction gain. The residual spectrum, in this case, would be more white after removing the global shape due to the vocal tract and the fine structure due to the periodicity of the signal.

The inverse filter of a long-term predictor is given by,

$$B(z) = 1 - \sum_{k=-\nu}^{\nu} b_k z^{-(M+k)} \quad (2.32)$$

Suitable values for ν are 0 or 1. In the last case we have three nonzero taps. A one tap predictor may be unable to work efficiently when the pitch period is not an integer multiple of the sampling period. On the contrary, a multi-tap predictor can interpolate the pitch lag between consecutive samples. In the speech coding literature, $B(z)$ is also referred as the pitch filter.

2.3.5 The Linear Prediction Model

A complete analysis/synthesis structure is shown in figure 2.4. The analysis stage has the ability to decorrelate the speech signal. That is, redundant information is removed

by the analysis filters and placed back by the synthesis filters. This is a desired feature in speech compression and will be exploited in the design of efficient voice codecs. To recover the speech signal the synthesis structure must have exactly the same parameters of the analysis filters. Assuming no round off noise, the synthetic and original signal should be identical, regardless of the filter parameters. Finally, note that there are two possible orders. The order adopted in this research work is the F-P cascade, that is, the formant analysis filter is followed by the pitch analysis filter. It was found in [24] that the F-P cascade outperforms the P-F cascade.

2.4 Distortion Measures for Speech Processing

In many speech processing applications it is necessary to determine how close two speech sequences are. To evaluate the performance of a speech codec, for example, the quality of the reconstructed speech has to be compared against the original one. In this case we want to assign a positive number to the corresponding difference in quality between the two signals by means of a distortion measure. The desired properties of such distortion measure can be listed as follows [25, 26],

1. It must correspond to psychological attributes of human perception.
2. It must be mathematically tractable and efficiently computable.
3. It must be a positive number or zero (infinite) when both frames are identical.

Note that the symmetry and triangular inequality are not included in the list. Hence the distortion measure is not required to be a metric and therefore it is not called a distance.

One of the most popular distortion measures is the squared error. Although it is mathematically tractable it is not a subjectively meaningful distortion in most low bit-rate speech processing applications. A method that is closely related to the linear prediction theory uses ratios of residual energies. This is the *log-likelihood ratio* distortion measure (d_{LLR}) [27, 28, 25, 26, 29, 30, 31].

Figure 2.4 Linear prediction analysis/synthesis structure.

2.4.1 Signal-to-Noise Ratio

Historically, the squared error is normalized with the energy of the signal and called signal-to-noise ratio,

$$SNR = \frac{\sum_n s(n)^2}{\sum_n [s(n) - \hat{s}(n)]^2} \quad (2.33)$$

A more subjectively meaningful measure is the segmental SNR (SEGSNR). The signal is divided in segments of 10 to 20 ms and the SNR is computed for each segment in dB. The SEGSNR is the average given by,

$$SEGSNR = \frac{1}{M_s} \sum_{m=1}^{M_s} SNR_{dB}(m) \quad (2.34)$$

where M_s is the number of segments. Silent periods can be eliminated from the computation if no noise is expected during these periods. The advantage of the SEGSNR is that it is more sensitive to segments of poor quality. In the conventional SNR these segments are generally masked by periods of low distortion.

Psychoacoustical studies show that the human auditory system can tolerate more noise in the formants of the speech spectrum than in the valleys. Therefore, a more subjective distortion measure can be obtained by weighting the spectrum of the error. Regions of the error spectrum that correspond to valleys between formants in the speech spectrum are de-emphasized and regions corresponding to the formants are emphasized. Using a weighting function $W(z)$ we can write,

$$\epsilon_w = \frac{1}{2\pi} \int_0^{2\pi} |S(e^{j\omega}) - \hat{S}(e^{j\omega})|^2 W(e^{j\omega}) d\omega \quad (2.35)$$

A general weighting filter is discussed in [32],

$$W(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)} \quad (2.36)$$

Listening tests showed that a good choice for the parameters is $\gamma_1=0.9$ and $\gamma_2=0.4$. However, in some coders, other combinations are used. For example $\gamma_1=1$ and $\gamma_2=0.8$ is used in CELP coders to simplify the search procedures. The noise-weighted signal-to-noise ratio (NWSNR) uses the noise-weighted mean-squared error (NWMSE) given by equation 2.35.

2.4.2 Log-Likelihood Ratio Distortion Measure

In linear prediction theory, the minimum residual energy for a particular speech frame is given by

$$\alpha = r_0 - \mathbf{a}^T \mathbf{r} \quad (2.37)$$

If the same frame is passed through a non-optimum inverse filter then the residual energy β must be greater than α , where

$$\beta = r_0 - 2\hat{\mathbf{a}}^T \mathbf{r} + \hat{\mathbf{a}}^T \mathbf{R} \hat{\mathbf{a}} \geq \alpha \quad (2.38)$$

with equality holding only for $\mathbf{a} = \hat{\mathbf{a}}$. The log-likelihood distortion measure is defined as

$$d_{LLR} = \log \left(\frac{\beta}{\alpha} \right) \quad (2.39)$$

and has proved to be subjectively meaningful [25, 30, 31].

Figure 2.5 shows the filtering operation. The two input sequences are $s(n)$ and $\hat{s}(n)$. The corresponding p^{th} order inverse filters are $A(z)$ and $\hat{A}(z)$. When one of the input sequences, say $s(n)$ is filtered through the filters, the resulting residual energies are α and β . A different distortion would be obtained by using $\hat{s}(n)$ instead of $s(n)$ as the input sequence. This fact shows the asymmetric nature of the likelihood ratio.

The d_{LLR} can also be used to compare two inverse filters. It has extensively been used in the design of vector quantizers in linear predictive coders. In this case a nearest neighbor algorithm has to choose the closest inverse filter from a codebook [33, 34].

Figure 2.5 Computation of the residual energies
in the log-likelihood ratio distortion measure.

2.5 Speech Coding

2.5.1 Linear Prediction Coders

Linear prediction theory has been used extensively in the design of speech coders for the last 50 years. A brief description of some LPC codecs is presented in this section. Note that there exist many possible variations. Our objective here is to show the evolution of the LPC codecs in a simple way. Five different speech synthesizers are shown in figure 2.6.

The *Vocoder* produces a synthetic waveform that is not closely related to the original speech sequence [16]. It preserves basic characteristics of the speech signal such as type of excitation, energy and formants. The excitation is either voiced with a certain pitch period or unvoiced. The result is poor speech quality at a very low bit rate in the order of 2 Kb/s or lower. Medium complexity is required to extract prediction parameters, voicing and pitch period for every frame.

The following two methods in the same figure are waveform decoders. The accuracy of the reconstructed speech waveform is preserved in a mean-squared-error sense. This leads to good speech quality at bit rates above 16 Kb/s. In Adaptive Differential Pulse Code Modulation (ADPCM) codecs [3], the encoder sends the residual through the channel. The parameters of $A(z)$ can be sent through the channel as side information or recovered at the other end of the channel based on synthesized speech. Both approaches will be discussed in section 2.5.2. For the case in which the LPC parameters are sent as side information, the synthesis filter is generally updated at intervals of 20 ms. Adaptive Predictive Coding (APC) [3] incorporates a long-term predictor which is generally updated several times in a single frame.

The last two approaches in figure 2.6 are called *hybrid coders* [35, 36, 37]. They combine the accuracy of waveform coders and the excitation modeling of vocoders. In these cases the excitation models are more sophisticated and complex. The encoders are

Figure 2.6 Linear prediction speech synthesizers

based on an analysis-by-synthesis search that finds the optimum excitation sequence. In a *Multipulse* LPC codec only a small number of pulses are sent through the channel, e.g. 8 pulses in a 20 ms frame. The encoder calculates the optimum amplitude and position of each pulse in order to obtain an accurate output. In Code Excited Linear Prediction (*CELP*) codecs the residual is vector quantized. The residual (or excitation) vector is chosen from a large codebook. The encoder seeks the index that correspond to the optimum excitation vector. Only the index (not the residual) is sent to the decoder along with the gain and linear prediction parameters.

2.5.2 Forward vs Backward Adaptation

All the codecs discussed in the last section use adaptive filters. The parameters of the filters must be available at both ends of the channel. There are two approaches for prediction adaptation. In the first one the parameters of the predictor are calculated at the encoder based on the original speech and sent to the decoder as side information. This method is usually called *Forward Adaptive* and is shown in figure 2.7. In the *Backward Adaptive* method, the coefficients are calculated based on data available at both ends of the channel, therefore no side information is necessary, see figure 2.8.

In forward adaptive codecs the incoming speech samples are buffered. Block algorithms such as the autocorrelation or covariance methods can be used to calculate the linear prediction coefficients. This computation is performed in a frame by frame basis. The encoding delay is proportional to the frame length and might be excessively high for certain communications networks. Note also that the predictor parameters must be quantized prior to transmission.

Backward adaptive coders send a single bit stream through the channel. No extra channel capacity for side information and no buffering of the original speech sequence is necessary. The linear predictor can be updated on a sample by sample basis using recursive algorithms such as the LMS.

A comparison of the two methods shows that at a high bit rate both approaches achieve similar SNR. Low delay is a clear advantage of backward adaptive codecs. However, sensitivity to channel errors is also expected to be higher. This is due to the propagation of errors in the backward adaptive process. Finally, the performance of forward adaptive codecs tends to be better as the bit rate decreases and the quantization effects are more pronounced.

Figure 2.7 Generic forward-adaptive LPC codec

ENCODER **DECODER****Figure 2.8 Generic backward-adaptive LPC codec**

Chapter 3

Code Excited Linear Prediction Coders

3.1 Introduction

Linear prediction waveform coders tend to perform poorly as the bit rate decreases below 10 kb/s, restricting its use to medium rate speech coders. To overcome this limitation, Atal and Schroeder [38] proposed an analysis-by-synthesis code-excited linear prediction (CELP) speech coder.

In high bit rate coders, the residual is generally quantized in a sample by sample manner. However, further bit-rate reductions can be achieved by using vector quantization. A fundamental result of rate-distortion theory is that coding vectors instead of scalars leads to better performance, even when the original data is uncorrelated or independent [39]. Vector quantization exploits four interrelated properties of the vectors [9]: linear dependency, nonlinear dependency, shape of the probability density function, and vector dimensionality. In CELP coders the residual is vector quantized using an analysis-by-synthesis search procedure. Low bit rates and good quality are achieved at the cost of high computational complexity and long encoding delays.

A CELP decoder is depicted in figure 2.6. Its structure is very similar to the APC decoder. The only difference is that instead of receiving a sequence of quantized residual samples, the CELP decoder receives an index corresponding to an excitation vector (or codevector). Two identical large codebooks containing L codevectors of dimension K are located at both ends of the channel. The decoder simply retrieves the corresponding codevector that will be used in the synthesis process. The search for the optimum excitation vector takes place at the encoder.

The basic operation of the CELP encoder is shown in figure 3.1. Assuming that the system has already calculated the filter parameters, it passes all the excitation vectors through the synthesis filters producing the synthesized speech vectors $\hat{s}_n^{(l)}$, $l = 0 \dots L - 1$. The coder compares the original and synthesized speech vectors using a noise-weighted mean-squared error distortion measure. After computing the distortion measure for each element of the L-level codebook, the coder chooses the codevector that minimizes the NWMSE. The encoder also has to perform linear prediction analysis, estimate pitch period and gain, and send all the data to the decoder.

There are two possible types of codebooks: stochastic and deterministic. We can create a stochastic codebook using white gaussian random numbers with unity variance. The gaussian assumption works properly for most quasi-stationary segments of speech. This assumption is not so accurate for stops and transitions from unvoiced or silence regions to voiced speech [38]. We can also design a deterministic codebook using an initial (stochastic) codebook, a big speech database and a VQ design procedure. This codebook optimization procedure can slightly improve the SNR and will be discussed in section 3.2.5.

3.2 Design

This section will describe in detail how to design a forward-adaptive CELP coder. The next chapter will use this design as the base for further improvements.

Figure 3.1 CELP Encoder

3.2.1 Organization

Let us organize the original speech sequence in two levels, the frame and the subframe levels. The frame is a segment of the original speech sequence in which the parameters of the signal are expected to be stationary in order to compute the short-term linear prediction analysis. The length of the frame has to be long enough to permit a good linear prediction analysis while maintaining a low bit rate. The shorter the frame, the more often we have to send linear prediction coefficients and therefore the higher the bit rate. A typical value for the frame size is 160 samples or 20 ms at a sampling rate of 8000 samples/s. The subframe is a shorter segment, and generally its length is an integer sub-multiple of the frame size. The length of the subframe defines the dimension of all the vectors that flow along the system and the rate at which certain parameters are updated such as the pitch period and the excitation vector index. In the speech coding literature the subframe is also called the “vector dimension” or simply the “block of samples”.

Three important decisions that have to be considered in the design process are the use of:

1. Open-loop vs closed-loop procedures.
2. Sequential vs joint optimization.
3. Forward vs backward adaptation.

In an open-loop procedure, the parameters are estimated without using any kind of feedback. For example, we can estimate the pitch period of the frame by performing a cepstral [40] or autocorrelation analysis on the input speech. On the other hand, a closed-loop procedure computes the synthesis parameters by minimizing the distortion between original and synthesized subframes. In our previous example we would choose the pitch period that minimizes the MSE. Ideally all the parameters of the coder can be optimized in a closed-loop procedure. However, the short-term predictor would lead

to a highly non linear set of equations with no direct solution [41]. A practical trade-off is to compute the short-term linear prediction coefficients in an open-loop manner based on the original speech. The remaining parameters (gain, pitch lag and pitch filter coefficients) can be computed using closed-loop procedures. Note that the closed-loop approach will lead to a more accurate synthesized speech signal at the cost of more computational complexity in the encoding process.

The parameters that will be calculated in a closed loop manner can be optimized using either sequential or joint methods. The optimum solution (in a MSE sense) jointly solves the set of equations that minimizes the error. However suboptimum (or sequential) solutions can further reduce complexity and still produce good results. For example, we can search for the pitch period and pitch filter coefficients sequentially as will be described in a later section.

Our final consideration is whether to compute a certain parameter in a backward or forward adaptive mode. It should be noted that only those parameters that are calculated using open-loop procedures can be transmitted in a backward-adaptive mode. The reason for this is that closed-loop procedures require the original speech which is not available at the decoder.

3.2.2 Search Procedures

The computational complexity of the CELP encoder is extremely large. Many simplified methods are based on a separation of the zero-input (zi) and zero-state (zs) components of the synthesis filters. These techniques lead to exact results or close approximations of the original full complexity method [42]. In order to simplify the calculations in the forward-adaptive case we define $\gamma_1=1$ and $\gamma_2=\gamma$.

In figure 3.2 the weighting filter $W(z)=A(z)/A(z/\gamma)$ is moved from its original location to both of its input branches. The formant synthesis filter is combined with the weighting filter forming a weighted formant synthesis filter. The poles of this filter are moved

inward causing a more damped impulse response. At the other branch, the input speech is passed through the weighting filter. The output of the synthesis filters has two components: the zero-state response due to the current input sequence and the zero-input response that is caused by the "memory" of the filter. Given a discrete-time filter, the output sequence can be written as,

$$y(n) = \sum_{k=-\infty}^{+\infty} x(n)h(n-k) \quad (3.1)$$

where $x(n)$ and $y(n)$ are the input and output sequences respectively and $h(n)$ is the impulse response. For a block of samples starting at n_0 we can break the input sequence and get the zi and zs output sequences,

$$\begin{aligned} y_{zi}(n) &= \sum_{k=-\infty}^{n_0-1} x_{zi}(n)h(n-k) & x_{zi}(k) &= \begin{cases} x(k) & k < n_0 \\ 0 & k \geq n_0 \end{cases} \\ y_{zs}(n) &= \sum_{k=n_0}^{+\infty} x_{zs}(n)h(n-k) & x_{zs}(k) &= \begin{cases} 0 & k < n_0 \\ x(k) & k \geq n_0 \end{cases} \\ \Rightarrow y(n) &= y_{zi}(n) + y_{zs}(n) & & \\ &= \sum_{k=-\infty}^{+\infty} [x_{zi}(k) + x_{zs}(k)]h(n-k) & & \\ &= \sum_{k=-\infty}^{+\infty} x(k)h(n-k) & & \end{aligned} \quad (3.2)$$

The zero-state sequence can be obtained by resetting the memory of the filter to zero prior to the filtering operation. To generate the zero-input component, a null sequence is passed through the filter.

In figure 3.3 the filters are separated into their zi and zs parts. A direct consequence of this division is that the transfer function of the zero-state pitch synthesis filter is unity when the minimum pitch period is greater than the subframe size. To show this property we can consider a three-tap filter given by,

$$d(n) = e(n) + b_{-1}d(n - (M - 1)) + b_0d(n - M) + b_1d(n - (M + 1)) \quad (3.3)$$

for the zero-state component, the input is,

$$e_{zs}(n) = \begin{cases} 0 & n < n_0 \\ e(n) & n \geq n_0 \end{cases} \quad (3.4)$$

and the output can be expressed as,

$$d_{zs}(n) = \begin{cases} 0 & n < n_0 \\ e_{zs}(n) & n_0 \leq n < n_0 + (M - 1) \\ e_{zs}(n) + \sum_{k=-1}^{+1} b_k d(n - (M + k)) & n \geq n_0 + (M - 1) \end{cases} \quad (3.5)$$

observing the second case where $d_{zs}(n) = e_{zs}(n)$ we conclude that if the minimum pitch period is always greater than the speech subframe, then the transfer function of the zero-state pitch filter is unity,

$$M_{min} \geq K + 1 \Rightarrow \frac{1}{B(z)} = 1 \quad \text{zero state} \quad (3.6)$$

In some practical coders, the subframe size is larger than the minimum pitch period of normal speech (specially for female utterances). However, large long-term prediction gains can still be obtained by using multiples of the original pitch period. Therefore, condition 3.6 can be used at the cost of small degradation in performance.

The formant synthesis filter that follows the zero-input pitch filter in figure 3.3 can also be split into *zi/zs* components. Assuming that condition 3.6 holds, we get the block diagram shown in figure 3.4.

The search for the optimum codevector is now based on the residual speech after subtracting the memory from previous speech subframes. The convolution of the l^{th} codevector with the impulse response of the weighted synthesis filter $1/A(z/\gamma)$ can be written in matrix form as,

$$\hat{\mathbf{z}}(l) = G(l)H\mathbf{v}(l) \quad (3.7)$$

Figure 3.2 The weighting filter is moved

Figure 3.3 Zero-input, zero-state separation

Figure 3.4 Simplified search

where $\mathbf{H}$ is a lower triangular Toeplitz matrix containing the impulse response of $1/A(z/\gamma)$ in its first column,

$$\mathbf{H} = \begin{bmatrix} h_0 & 0 & 0 & \cdots & 0 \\ h_1 & h_0 & 0 & \cdots & 0 \\ h_2 & h_1 & h_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{K-1} & h_{K-2} & h_{K-3} & \cdots & h_0 \end{bmatrix} \quad (3.8)$$

The NWMSE is given by,

$$\begin{aligned} \epsilon_w(l) &= \|\mathbf{z} - \hat{\mathbf{z}}(l)\|^2 \\ &= \|\mathbf{z}\|^2 - 2G(l)\mathbf{z}^T \mathbf{H}\mathbf{v}(l) + G^2(l)\mathbf{v}^T(l)\mathbf{H}^T \mathbf{H}\mathbf{v}(l) \end{aligned} \quad (3.9)$$

Taking the derivative with respect to $G(l)$ we get the minimum NWMSE and the optimum gain for the l^{th} codevector,

$$\begin{aligned} \frac{\partial \epsilon_w(l)}{\partial G(l)} &= -2\mathbf{z}^T \mathbf{H}\mathbf{v}(l) + 2G(l)\mathbf{v}^T(l)\mathbf{H}^T \mathbf{H}\mathbf{v}(l) = 0 \\ \Rightarrow G(l) &= \frac{\mathbf{z}^T \mathbf{H}\mathbf{v}(l)}{\mathbf{v}^T(l)\mathbf{H}^T \mathbf{H}\mathbf{v}(l)} \\ \Rightarrow \epsilon_w(l) &= \|\mathbf{z}\|^2 - \frac{(\mathbf{z}^T \mathbf{H}\mathbf{v}(l))^2}{\mathbf{v}^T(l)\mathbf{H}^T \mathbf{H}\mathbf{v}(l)} \end{aligned} \quad (3.10)$$

In order to find the optimum excitation vector out of the L -level codebook it is necessary to maximize the second term of the minimum error:

$$\text{find } l_{\text{opt}} \Rightarrow \max_{l=0 \dots L-1} \frac{(\mathbf{z}^T \mathbf{H}\mathbf{v}(l))^2}{\mathbf{v}^T(l)\mathbf{H}^T \mathbf{H}\mathbf{v}(l)} \quad (3.11)$$

This procedure seeks the index of the optimum codevector and gain for a single subframe. Before proceeding with the next block of samples, the state of the zero-input filters must be updated. To accomplish this, the weighted synthesis structure shown in figure 3.5 is used. The optimum codevector is passed through the filters using the optimum gain. The new "memory" of these filters correspond to the state of the updated zero-input filters. To prepare the zero-state filters for the next block of samples, their buffers must be reset to zero.

Figure 3.5 Update of the zero-input synthesis filters

3.2.3 Complexity Reduction

In order to reduce the computational complexity even more, some techniques are discussed in this section. The most expensive operation is given by the product $\mathbf{v}^T \mathbf{H}^T \mathbf{H} \mathbf{v}$ in equation 3.11 which has to be computed L times per subframe.

3.2.3.1 Toeplitz Approximation This method assumes that the product $\mathbf{H}^T \mathbf{H}$ can be approximated to a matrix that is symmetric and Toeplitz. This approximation is possible when the impulse response of the weighted synthesis filter dies out in a period that is much shorter than the subframe size. This can be achieved by using long subframes or by reducing the weighting factor γ . The first column of the approximated toeplitz matrix is equal to the first column of $\mathbf{H}^T \mathbf{H}$ and can be written as follows,

$$\mathbf{H}^T \mathbf{H} \simeq \begin{bmatrix} r_{hh}(0) & r_{hh}(1) & \cdots & r_{hh}(K-1) \\ r_{hh}(1) & r_{hh}(0) & \cdots & r_{hh}(K-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{hh}(K-1) & r_{hh}(K-2) & \cdots & r_{hh}(0) \end{bmatrix} \quad (3.12)$$

where,

$$r_{hh}(i) = \sum_{j=i}^{K-1} h(j)h(j-i) \quad (3.13)$$

After some algebra we get the final expression,

$$\mathbf{v}^T \mathbf{H}^T \mathbf{H} \mathbf{v} \simeq r_{hh}(0)r_{vv}(0) + 2 \sum_{i=1}^{K-1} r_{hh}(i)r_{vv}(i) \quad (3.14)$$

The autocorrelation coefficients of the codevectors $r_{vv}(i)$ can be precomputed and stored in memory along with the codebook. The autocorrelation vector of the impulse response is calculated only once per frame.

3.2.3.2 Shift Symmetric Codebooks Another strategy to reduce the number of operations is to use a shift symmetric codebook. That is, given a sequence of random numbers $\{v(n)\}$, the l^{th} codevector can be expressed as $\mathbf{v}^T(l) =$

$[v(tl), v(tl + 1), \dots, v(tl + K - 1)]$. Consecutive vectors overlap in all but t elements. When this codebook is used $Hv(l)$ can be calculated in a recursive manner and unlike the toeplitz approximation the result is exact. The dependence between neighboring vectors may deteriorate performance. This degradation becomes rapidly negligible for values of $t=2$ or greater [43]. Another advantage is that shift-symmetric codebooks require almost K times less memory than conventional stochastic codebooks.

To compute the gain and the minimum NWMSE it is convenient to define: $c(l) = Hv(l)$ where $c(l)$ is the following K by 1 vector,

$$c(l)^T = [c_l(0) \ c_l(1) \ \dots \ c_l(K - 1)] \quad (3.15)$$

This vector can be computed recursively. For the case $t=1$ we have,

$$c(l+1) = Hv(l+1) = \begin{bmatrix} c_l(1) - h(1)v(l) \\ c_l(2) - h(2)v(l) \\ \vdots \\ c_l(K-1) - h(K-1)v(l) \\ h(K-1)v(l+1) + h(K-2)v(l+2) + \dots + h(0)v(l+K) \end{bmatrix} \quad (3.16)$$

where $c_{l(j)}$ is the j^{th} element in vector $c(l)$.

3.2.3.3 Sparse Codebooks Further reductions in computational complexity are possible by using sparse vectors in the excitation codebook. Note that Multipulse and CELP coders only differ in how the excitation sequence is generated. In multipulse LPC, the optimum excitation sequence has a few pulses per frame (typically 10-20 % of the samples are not zeros). This excitation sequence minimizes the NWMSE between original and synthesized speech. On the other hand, CELP coders use a codebook of random white gaussian vectors. Both approaches can be combined using sparse codebooks [44]. Sparse codebooks only have a few non zero elements in each codevectors. This type of codebook leads to efficient real-time implementations of the search procedure in terms the number of operations and storage requirements.

3.2.3.4 Summary The computational complexity and memory requirements of the different methods are shown in table 3.1 and figure 3.6. Shift-symmetric codebooks have a reduction factor of 5 when a subframe of 40 samples is used. Similar reductions can be obtained using the toeplitz approximation but with the restrictions discussed before.

3.2.4 Long-Term Predictor

The long-term predictor parameters can be calculated using a closed-loop procedure that maximizes the prediction gain between $y(n)$ and $z(n)$ in figure 3.4. An expanded diagram for a three-tap pitch synthesis filter is shown in figure 3.7. In order to reduce computational complexity, pitch period and LPC coefficients are computed sequentially. A practical way of estimating the pitch period is to minimize the squared-error using a one-tap filter. Note that this problem is identical to the search procedure with a shift-symmetric codebook. In this case a buffer is used instead of a fixed codebook. For this reason the pitch predictor is sometimes called *adaptive codebook*. The number of “adaptive codevectors” is equal to the difference between the maximum and minimum pitch periods. The optimum pitch period is calculated using the following search procedure:

$$\text{find } M \Rightarrow \max_{m=M_{\min} \dots M_{\max}} \frac{(\mathbf{y}^T \mathbf{H} \mathbf{d}(n-m))^2}{\mathbf{d}^T(n-m) \mathbf{H}^T \mathbf{H} \mathbf{d}(n-m)} \quad (3.17)$$

where $\mathbf{d}^T(n) = [\tilde{d}(n), \tilde{d}(n+1), \dots, \tilde{d}(n+K-1)]$. Once we estimated the pitch lag M , the next step is to calculate the predictor coefficients of the three-tap predictor. Applying the least-squares method, the objective is to minimize the error given by,

$$\epsilon_z = \|\mathbf{y} - \hat{\mathbf{y}}\|^2 \quad (3.18)$$

where,

$$\hat{\mathbf{y}} = \mathbf{H} \mathbf{d} \quad (3.19)$$

Method	Multiply-Adds per sample	Floating-Point Numbers
Direct	$(K/2+3)L$	KL
Toeplitz Approx	$K^2/N+4L$	KL
Shift-symmetric	5L	2K+L

Table 3.1 Complexity and storage requirements in search procedures

Figure 3.6 Number of multiply-adds in search procedures

and

$$\mathbf{d} = \mathbf{D}\mathbf{b} = \begin{bmatrix} \mathbf{d}(n-M-1) & \mathbf{d}(n-M) & \mathbf{d}(n-M+1) \end{bmatrix} \begin{bmatrix} b_1 \\ b_0 \\ b_{-1} \end{bmatrix} \quad (3.20)$$

Replacing in equation 3.18 we get,

$$\begin{aligned} \epsilon_z &= \|\mathbf{y} - \mathbf{H}\mathbf{D}\mathbf{b}\|^2 \\ &= \|\mathbf{y}\|^2 - 2\mathbf{b}^T \mathbf{D}^T \mathbf{H}^T \mathbf{y} + \mathbf{b}^T \mathbf{D}^T \mathbf{H}^T \mathbf{H} \mathbf{D} \mathbf{b} \end{aligned} \quad (3.21)$$

The error is minimized by taking the gradient with respect to $\mathbf{b}$,

$$\nabla_{\mathbf{b}} \epsilon_z = -2\mathbf{D}^T \mathbf{H}^T \mathbf{y} + 2\mathbf{D}^T \mathbf{H}^T \mathbf{H} \mathbf{D} \mathbf{b} = 0 \quad (3.22)$$

and the optimum coefficients $\mathbf{b}_{opt}$ are obtained by solving the linear system,

$$\mathbf{b}_{opt} = \left[\mathbf{D}^T \mathbf{H}^T \mathbf{H} \mathbf{D} \right]^{-1} \mathbf{D}^T \mathbf{H}^T \mathbf{y} \quad (3.23)$$

An efficient solution of the pitch coefficients can be implemented using Choleski decomposition on the 3-by-3 linear system.

3.2.5 Excitation Codebook Optimization

Although a large stochastic codebook is a rich source of excitation vectors, further improvements are possible by using the Generalized Lloyd Algorithm (GLA) [10, 45, 46]. This vector quantization codebook design procedure can be used to find a local minimum of the error. Usually, to design a codebook, a large database (or training sequence) is used to "teach" the system what the optimum codevectors are. The average distance (distortion) between the training vectors and the codevectors converges to a local minimum error after several iterations. The minimum error is local because its value depends on the initial codebook. There is no algorithm to find a global minimum. In CELP, the GLA does not even guarantee convergence to a local minimum. This is due to the feedback through the filter memory. That is, the error for a particular codevector

Figure 3.7 Pitch predictor optimization

depends on the current and previous codevectors. As a consequence, the global error decreases for the first iterations and then oscillates around a certain value [46].

A general procedure for shift-symmetric codebooks is presented in [46]. The conventional codebook (no overlap) can be considered as a particular case of the shift-symmetric codebook. Two consecutive codevectors have all but t elements in common. Where $t=1$ in the highly overlapped case and $t=K$ in the conventional (no overlap) case. Let us express the l^{th} codevector as,

$$\mathbf{v}(l) = \mathbf{S}(l)\mathbf{u} \quad (3.24)$$

where $\mathbf{u}^T = [v_0, v_1, \dots, v_{K+t(L-1)}]$ is a long vector containing random numbers and

$$\mathbf{S}(l) = [\mathbf{0}_{K,t} | \mathbf{I}_{K,K} | \mathbf{0}_{K,L-t}] \quad (3.25)$$

is a K by $K+t(L-1)$ shifting matrix where $\mathbf{0}_{K,n}$ is a K by n matrix of zeros and $\mathbf{I}$ is the identity matrix.

The objective is to minimize the sum of the errors between the original speech (after subtracting the memory due to previous subframes) and the synthesized speech. This can be expressed as,

$$\bar{\epsilon}_w = \sum_{m=0}^{M_T-1} \|\mathbf{z}_m - G_m \mathbf{H}_m \mathbf{v}(l_m)\|^2 \quad (3.26)$$

where M_T is the total number of subframes in the training sequence. The subscript m corresponds to the m^{th} subframe. Substituting equation 3.24 in equation 3.26 we get,

$$\begin{aligned} \bar{\epsilon}_w &= \sum_{m=0}^{M_T-1} \|\mathbf{z}_m - G_m \mathbf{H}_m \mathbf{S}(l_m) \mathbf{u}\|^2 \\ &= \bar{\sigma}_z^2 - 2\bar{\mathbf{c}}^T \mathbf{u} + \mathbf{u}^T \bar{\mathbf{R}} \mathbf{u} \end{aligned} \quad (3.27)$$

where,

$$\bar{\sigma}_z^2 = \sum_{m=0}^{M_T-1} \mathbf{z}_m^T \mathbf{z}_m \quad (3.28)$$

$$\bar{\mathbf{c}} = \sum_{m=0}^{M_T-1} G_m \mathbf{S}^T(l_m) \mathbf{H}_m^T \mathbf{z}_m \quad (3.29)$$

$$\bar{\mathbf{R}} = \sum_{m=0}^{M_T-1} G_m^2 \mathbf{S}^T(l_m) \mathbf{H}_m^T \mathbf{H}_m \mathbf{S}(l_m) \quad (3.30)$$

The error is minimized by taking the gradient with respect to $\mathbf{u}$. The optimized codebook is thus given by,

$$\mathbf{u}_{opt} = \bar{\mathbf{R}}^{-1} \bar{\mathbf{c}} \quad (3.31)$$

3.2.6 Quantization of the LPC Parameters

In order to transmit the short- and long-term predictors through a communications channel, it is necessary to represent the LPC coefficients with a limited number of bits. An efficient quantization process achieves minimum distortion for a given bit rate. Alternate representations of the LPC parameters have different quantization characteristics. Since a profound study of the problem is beyond the scope of this thesis, we will present suitable quantization schemes for our CELP coder design.

One of the choices we have to make is whether to use scalar or vector quantization of the LPC parameters [9, 39, 33]. Vector quantization is a more effective solution in terms of distortion for a certain number of bits. However, the computational cost grows exponentially with the order of the vector. For this reason we adopt a scalar quantization solution for the short-term LPC coefficients. On the other hand, since the order of the long-term predictor is small ($p=3$), we can quantize $B(z)$ using the VQ approach.

3.2.6.1 Scalar Quantization of $A(z)$ Alternate representations of the inverse filter can be found in the LPC and speech literature [18, 15, 47, 48]. Among others, the most used parameters are:

- Impulse response of the inverse filter $A(z)$: a_i , $1 \leq i \leq p$.

- Impulse response of the all-pole model $H(z)$: $h_i, 0 \leq i \leq p$.
- Autocorrelation of the a_i 's: $r_{aa}(k), 0 \leq k \leq p$.
- Autocorrelation of the h_i 's: $r_{hh}(k), 0 \leq k \leq p$.
- Spectral coefficients of either autocorrelation $r_{aa}(k)$ or $r_{hh}(k)$.
- Cepstral representation of either a_i or h_i .
- Poles of $H(z)$ or zeros of $A(z)$.
- Reflection coefficients of the acoustic model: $r_i, 1 \leq i \leq p$.
- Log-area ratios (LAR): $g_i, 1 \leq i \leq p$.
- Inverse sine functions: $g_i, 1 \leq i \leq p$.
- Line spectrum pairs (LSP).

All the above representations require either $p+1$ coefficients or p coefficients plus the *gain*. In the context of scalar quantization, the desired properties of the parameters are:

1. Low distortion.
2. Filter stability upon quantization.
3. Natural ordering of the parameters.
4. Low complexity.

A direct representation of the coefficients of $A(z)$ is inappropriate. They have a relatively large dynamic range and are difficult to control for stability. From the alternate representations only the poles and zeros do not present natural ordering. A natural ordering will permit us to exploit the statistics of the individual parameters. For example, by assigning different number of bits to each parameter according to dynamic range and perceptual importance. Reflection coefficients not only satisfy property 3 but also are always stable and easy to compute. However, their quantization characteristics become poor for values close to ± 1 . To overcome this problem, a nonlinear transformation can be used. The inverse sine functions, for example, can be expressed as,

$$g_i = \sin^{-1} r_i \quad (3.32)$$

where r_i is the i^{th} reflection coefficient. A widely used transformation is the log-area ratio given by,

$$g_i = \log \frac{1 + r_i}{1 - r_i} = 2 \tanh^{-1} r_i \quad (3.33)$$

A more efficient representation that allows 30% lower bit rate than with reflection coefficients is given by the line spectrum pairs. LSPs have low dynamic range, are stable and more efficient than other representations. The cost is paid with more computational complexity than the LARs. To obtain the parameters, we extend the order of the filter to $p+1$ by adding a new reflection coefficient $r_{p+1}=1$ and $r_{p+1}=-1$. This is like adding an open and a closed tube to the acoustic model. The extended filters are given by,

$$\begin{aligned} P(z) &= A(z) + z^{-(p+1)} A(z^{-1}) \\ Q(z) &= A(z) - z^{-(p+1)} A(z^{-1}) \end{aligned} \quad (3.34)$$

and

$$A(z) = [P(z) + Q(z)]/2 \quad (3.35)$$

Three important properties of $P(z)$ and $Q(z)$ are [47],

1. All zeros of $P(z)$ and $Q(z)$ are on the unit circle.
2. Zeros of $P(z)$ and $Q(z)$ are interlaced with each other.
3. Stability is easily preserved after quantization.

From property 1 we conclude that all the zeros are given by $z_i = e^{j\omega_i}$. The ω_i 's are the LSP frequencies. Several methods have been developed to quantize the LSP frequencies [47, 48, 49].

3.2.6.2 Vector Quantization of $B(z)$ In this section we will present a method for VQ of $B(z)$. The procedure is similar to that presented in section 3.2.5. In section 3.2.4 we discussed a closed-loop procedure to calculate the pitch predictor parameters. However,

the coefficients of $B(z)$ have to be quantized before transmission. To accomplish that, we will use a codebook $\mathbf{p}$ that contains L_p three-dimensional coefficient vectors. Let us define the $3L_p$ by 1 codebook as,

$$\mathbf{p}^T = [\mathbf{b}^T(0), \mathbf{b}^T(1), \dots, \mathbf{b}^T(L_p - 1)] \quad (3.36)$$

Therefore, the l^{th} vector is given by,

$$\mathbf{b}(l) = \begin{bmatrix} b_1(l) \\ b_0(l) \\ b_{-1}(l) \end{bmatrix} = \mathbf{S}(l)\mathbf{p} \quad (3.37)$$

where

$$\mathbf{S}(l) = [\mathbf{0}_{3,3M} | \mathbf{I}_{3,3} | \mathbf{0}_{3,L_p-3M}] \quad (3.38)$$

is a 3 by $3L_p$ shifting matrix similar to that presented in section 3.2.4. To find the optimum coefficient vector out of the L_p -level codebook, we have to minimize equation 3.21. Since $\|\mathbf{y}\|^2$ is constant for a certain block of samples, we can write the search as,

$$\text{find } l \quad \min_{l=0 \dots L_p-1} -2\mathbf{b}^T(l)\mathbf{D}^T\mathbf{H}^T\mathbf{y} + \mathbf{b}^T(l)\mathbf{D}^T\mathbf{H}^T\mathbf{H}\mathbf{D}\mathbf{b}(l) \quad (3.39)$$

The pitch codebook is optimized using the General Lloyd Algorithm. The initial codebook is implemented with a random set of coefficient vectors. That is, we run the coder on a rich speech database and pick L_p unquantized vectors at random. The optimized codebook is obtained after minimizing the accumulative error given by,

$$\bar{\epsilon}_z = \sum_{j=0}^{M_T-1} \|\mathbf{y}_j - \mathbf{H}_j\mathbf{D}_j\mathbf{b}(l_j)\|^2 \quad (3.40)$$

where $\mathbf{b}(l_j)$ is the optimum coefficient vector for the j^{th} block of samples. Thus, substituting equation 3.37 in equation 3.40 we get,

$$\begin{aligned} \bar{\epsilon}_z &= \sum_{j=0}^{M_T-1} \|\mathbf{y}_j - \mathbf{H}_j\mathbf{D}_j\mathbf{S}(l_j)\mathbf{p}\|^2 \\ &= \bar{\sigma}_y^2 - 2\bar{\mathbf{c}}^T\mathbf{p} + \mathbf{p}^T\bar{\mathbf{R}}\mathbf{p} \end{aligned} \quad (3.41)$$

where,

$$\bar{\sigma}_y^2 = \sum_{j=0}^{M_T-1} y_j^T y_j \quad (3.42)$$

$$\bar{\mathbf{c}} = \sum_{j=0}^{M_T-1} \mathbf{S}^T(l_j) \mathbf{D}_j^T \mathbf{H}_j^T y_j \quad (3.43)$$

$$\bar{\mathbf{R}} = \sum_{j=0}^{M_T-1} \mathbf{S}^T(l_j) \mathbf{D}_j^T \mathbf{H}_j^T \mathbf{H}_j \mathbf{D}_j \mathbf{S}(l_j) \quad (3.44)$$

The optimized pitch codebook is simply,

$$\mathbf{p}_{opt} = \bar{\mathbf{R}}^{-1} \bar{\mathbf{c}} \quad (3.45)$$

which can be solved using Choleski decomposition. This procedure has to be used iteratively until the error converges to its local minimum.

Chapter 4

Low-Delay Speech Coders

4.1 Backward Adaptive Coders at 16 Kb/s

The International Consultative Committee for Telephone and Telegraph (CCITT) is considering a new standard for low-delay network quality speech at 16 kb/s [35]. It follows the G.721 standard for 32 Kb/s ADPCM which has extensively been used in public and private telecommunications since 1986 [35]. The most important desired characteristic of the new standard is that the one-way encoder/decoder delay should not exceed 5 ms (the objective is ≤ 2 ms). The CCITT standard also requires that the distortion at bit error rates of 10^{-3} and 10^{-2} should be not worse than G.721.

The delay requirements prevent the speech coder design from using a conventional forward-adaptive short-term predictor. This means that the linear prediction analysis must be performed in a backward-adaptive mode, using past synthesized speech. Two candidates for the new standard are presented in [12] and [50]. Both designs are modifications of the original CELP coder in which only the excitation vector index is transmitted through the channel and the rest of the parameters are obtained in a backward-adaptive mode. In [12] the 16 Kb/s are covered using a subframe size of 5 samples and an excitation codebook of 10 bits. Thus, at a sampling frequency of 8

K samples/s the total bit rate is $8 K\text{samples/s} \times 10 \text{ bits} / 5 \text{ samples} = 16 \text{ Kbits/s}$. The other design uses a subframe size of 4 samples and therefore a codebook of 8 bits.

The design presented in [12] uses a single all-pole synthesis filter. The linear prediction analysis is performed using the autocorrelation method on past quantized speech. A backward pitch predictor is considered very sensitive to channel errors and therefore, discarded. The elimination of the long-term predictor affects female speech quality. It is shown that linear prediction gains for females saturate when the order of the predictor reaches 50. As a result, a single high-order ($p=50$) short-term predictor is adopted. It is possible to update the predictor every 5 samples (i.e. every subframe) but the complexity would be too high. As a result, the filter is updated at a lower rate (in the range of 2.5 to 5 ms) with little degradation in speech quality. This CELP coder also uses a different approach for the weighting filter. Note that the weighting process only takes place at the encoder. Therefore, there is no need to use the backward-adaptive predictor parameters in the weighting filter. We can use a more accurate linear prediction analysis based on the original speech. This approach, although more complex, can only improve the subjective quality of the speech signal. Consequently, the coder performs two parallel LPC analyses. One based on the synthesized speech (order 50) for $A(z)$ and the other one based on unquantized speech (order 10) for $W(z)$. In this case, there is no reason for using the simplified weighting filter (i.e. $\gamma_1=1$, $\gamma_2=0.8$). Instead, the general weighting filter discussed in chapter 2 is used.

The low-delay coder discussed in [50] uses a three-tap pitch predictor and a pole-zero formant predictor. The linear prediction coefficients are calculated in a sample-by-sample mode using a gradient adaption algorithm. In the same article, a comparison between recursive and block algorithms for the short-term linear prediction analysis is presented. The recursive approach is considered the best option due to its lower computational complexity and lower sensitivity to channel errors. The recursive approach clearly

outperforms the block method in the presence of transmission errors.

4.2 A New Low-Delay 8 Kb/s Backward-Adaptive CELP Coder

4.2.1 Overview

In this section we will discuss a new low-delay 8 Kb/s CELP coder (BA-CELP). The delay is reduced by sending the short-term predictor in a backward adaptive mode. At 8 Kb/s the quantization effects are more pronounced than in the 16 Kb/s backward-adaptive coders discussed in section 4.1. Therefore, the tracking of the filter tends to be poor during rapid changes of the speech spectrum. To prevent filter mistracking a new distortion measure is introduced. This distortion combines the conventional NWMSE used in forward-adaptive CELP with the log-likelihood ratio distortion measure. As a result, the speech quality of the new coder is almost as good as in the conventional (large-delay) CELP coder.

The main features of the proposed coder are:

- Backward-adaptive formant filter to reduce the delay.
- General weighting filter based on original speech to enhance perceptual quality.
- Three-tap, forward-adaptive, closed-loop pitch filter to preserve the fine structure of the spectrum.
- Mixed distortion measure to prevent formant filter mistracking.
- Shift-symmetric, optimized excitation codebook to reduce computational and memory complexity.

A diagram of the encoder is shown in figure 4.1. Note that $W(z) = A(z/\gamma_1)/A(z/\gamma_2)$ is not combined with $1/\hat{A}(z)$. This is because, unlike the conventional case, the weighting filter and the formant synthesis filter use different LPC parameters. The computational complexity of the search procedures is almost not affected by this fact. Once the impulse

response of $H(z)=[1/\hat{A}(z)]W(z)$ is computed, the search procedures are identical to the conventional case. However, more computation is required for the extra LPC analysis.

The block diagram of the decoder is shown in figure 4.2. The parameters of the short-term predictor are computed based on the output speech in a backward-adaptive manner. The rest of the parameters (i.e. codebook index, gain, and pitch filter) are received from the channel. At the beginning, the initial conditions of $\hat{A}(z)$ are identical at the encoder and decoder. A simple way of doing this is by setting $\hat{A}(z)=1$. The buffer that contains past synthesized speech is reset to zero. The first subframe is computed using the forward-adaptive parameters and the initial $\hat{A}(z)=1$. Once the first subframe is produced, it is stored in the synthesized speech buffer. The second subframe is computed based on the second set of forward-adaptive parameters and the backward-adaptive $\hat{A}(z)$ which is computed using the recently updated synthesized speech buffer. Note that the LPC analysis is performed on the last four subframes. Therefore the length of the synthesized speech buffer is four times the subframe size. The convergence of the synthetic speech to the original speech is guaranteed by the analysis-by-synthesis procedure as long as the synthesis filters are stable. As a result, the implementation of the forward- and backward-adaptive decoders only differ in how the short-term predictor is updated.

In the BA-CELP coder, all the parameters are updated at the end of each subframe. The bit allocation scheme is shown in table 4.1. The subframe size is 26 samples and the sampling rate is 8 KHz. Consequently, the total delay (typically 4 times the vector dimension) is $4 \times 26 \text{ samples} / (8000 \text{ samples/s}) = 13 \text{ ms}$. This represents a reduction in coding/decoding delay of almost 80% (typical delay of the forward-adaptive coder is 60 ms).

4.2.2 The Mixed Distortion Measure

The critical problem in the BA-CELP coder is how to improve the spectral match

Figure 4.1 Backward-adaptive CELP encoder.

Figure 4.2 Backward-adaptive CELP decoder.

Parameter	bits/block	Kbits/sec
Formant filter	0	0.0
Pitch filter	5	1.5
Pitch lag	7	2.2
Gain	5	1.5
Excitation vec.	9	2.8
Total	26	8.0

Table 4.1 Bit allocation and corresponding bit rate. The vector dimension is 26 samples and the sampling rate is 8 KHz.

in the formant filter. This is solved, partially, by using an accurate three-tap pitch filter which sends information in a forward adaptive manner. The long-term predictor not only preserves the fine structure but also helps in global shaping of the speech spectrum. Further improvement in filter tracking can be achieved by using a *mixed distortion measure* in the excitation vector search procedure. The optimum codevector is searched in two sequential steps. First, a conventional search algorithm finds the best n_c excitation vectors that minimize the NWMSE. The best n_c candidates are used in the second stage in order to minimize the mixed distortion measure given by,

$$d_{mix}^{(i)} = d_{LLR} \left(A(z), \tilde{A}_i(z) \right) + \eta \log \frac{\epsilon_w^{(i)}}{\epsilon_w^{(min)}} \quad (4.1)$$

$$i = 1 \dots n_c$$

$$\eta \geq 0$$

where,

- $A(z)$ is the LPC model of the original speech sequence.
- $\tilde{A}_i(z)$ is the LPC model of the synthesized speech sequence for the next frame, assuming that the i^{th} candidate was chosen.
- $\epsilon_w^{(min)}$ is the minimum NWMSE out of the L-level codebook.
- $\epsilon_w^{(i)}$ is the NWMSE corresponding to the i^{th} candidate.
- η is a mixing factor to be optimized in subjective tests.

In equation 4.1, the first term on the right side is the log-likelihood ratio distortion of the next frame caused by the current i^{th} candidate vector. In the second term, the NWMSE is normalized with $\epsilon_w^{(min)}$. Therefore the logarithm of the quotient is greater than (or equal to) zero. The mixing factor η regulates the relative weigh of each term. As η increases, the mixed distortion measure approaches the NWMSE criterion. On the other hand, as η approaches zero, the LPC distortion of future frames decreases at the cost of accuracy in the current block of samples.

4.3 Computer Simulations and Results

Simulations of the BA-CELP coder were run on a Sparc workstation under the UNIX operating system. The code was written in the "C" language. In this section we present some of the results that will help to evaluate the performance of the coder.

Gain and pitch filter coefficients were quantized using scalar and vector quantizers respectively. The modified covariance method clearly outperformed the autocorrelation solution of linear prediction. The order of the short-term predictor was $p=10$ in all the cases. The mixing factor was adjusted to $\eta=1$ and the number of candidates was $n_c=16$. These values were selected by conducting informal listening tests. The excitation, pitch and gain codebooks were optimized with the GLA and a 0-minute speech database. In the long-term predictor, the pitch range was $27 < M < 156$ samples.

The BA-CELP coder was evaluated in terms of:

- Subjective speech quality. (Mean opinion score tests)
- Objective speech distortion. (NWMSE)
- Formant filter tracking. (Log-likelihood ratio distortion measure)

Speech quality assessment involves psychological attributes of human perception. As a result, formal subjective listening tests were conducted to evaluate speech quality using the "Mean Opinion Score" (MOS) test. Our objective was to compare the output speech quality of the forward-adaptive coder with the BA-CELP coder. In order to obtain meaningful results, a forward-adaptive CELP coder designed independently by W.P. LeBlanc was used as a reference. The forward-adaptive CELP coder used buffered speech to compute the LPC analysis with the autocorrelation method. The pitch filter had only one tap and the LPC coefficients were quantized using LSPs. The bit allocation scheme for the forward-adaptive version is shown in table 4.2. Note that the formant filter $A(z)$ is updated at the frame level (every 160 samples) and the rest of the parameters are updated at the subframe level (every 40 samples).

Parameter	bits/block	Update rate (per frame)	Kbits/sec
Formant filter	48	1	2.4
Pitch filter	5	4	1.0
Pitch lag	7	4	1.4
Gain	6	4	1.2
Excitation vec.	10	4	2.0
Total			8.0

Table 4.2 Bit allocation scheme for the forward-adaptive CELP coder.
The frame size is 160 samples and the subframe size is 40 samples.

In the MOS test, several subjects have to classify a speech stimulus material according to a five-point scale (see appendix A). The subjective tests followed the CCITT recommendations on subjective performance assessment of digital processes using the modulated noise reference unit with the following minimum variations:

1. No studio quality tape recorder was used.
2. The experiment was not conducted in a quiet room with controlled noise levels.
3. Headphones were used instead of telephone handsets.
4. Power level was constant throughout the experiment but not calibrated.

In the experiment, 20 listeners evaluated speech quality under five different conditions: 2 coders and 3 references. The reference conditions consisted of the original speech (PCM 64 Kb/s) and speech corrupted with random noise which had amplitude proportional to the instantaneous signal amplitude. The distorted speech was specified according to the "Modulated Noise Reference Unit" (MNRU) which is discussed in more detail in appendix A. The speech material consisted of 3 male and 3 female speakers under 5 different conditions (original, forward-adaptive coder, backward-adaptive coder, MNRU with $Q = 20$ dB and MNRU with $Q = 15$ dB). Mean opinion scores and 95% confidence intervals are shown in table 4.3. The results are classified by gender. In all the cases, female speakers received lower scores than male speakers. This is attributable to the original source where male speakers have 0.5 points more than female speakers. Average score for male speakers are identical in both coders. On the other hand, the response of the forward-adaptive coder is 0.2 points greater than in the BA-CELP case when female speech is processed. Global results show that speech quality in the conventional CELP coder is only 0.1 points better than the BA-CELP coder.

Figure 4.3 shows NWSNR as a function of the mixing factor η for a 30-second segment of speech. The dashed line represents the NWSNR in the conventional search, i.e. when the NWMSE criterion is used to find the optimum codevector. Note that

Condition	Speakers	Mean	Error
PCM 64 Kb/s (Original speech)	Female	3.97	0.20
	Male	4.50	0.21
	All	4.24	0.15
Forward Adaptive Coder	Female	3.13	0.29
	Male	3.71	0.27
	All	3.42	0.20
Backward Adaptive Coder	Female	2.95	0.29
	Male	3.71	0.19
	All	3.33	0.19
MNRU Q = 20 dB	Female	2.11	0.26
	Male	2.68	0.20
	All	2.39	0.18
MNRU Q = 15 dB	Female	1.42	0.18
	Male	1.95	0.28
	All	1.68	0.17

Table 4.3 Mean opinion score test.

according to equation 4.1 as η approaches infinity the NWSNR must converge to the dashed line. Observe that for values of η between 0.2 and 3 the global NWSNR was increased by using the suboptimal (in a NWMSE sense) mixed distortion measure. In other words, an increase in the error of individual blocks of samples helps in filter tracking and therefore improves the overall performance of the coder.

As we mentioned above, the shift-symmetric codebook was optimized according to the procedure discussed in section 3.2.5. In figure 4.4 we can see the improvement in NWSNR for 15 iterations of the GLA. A single 10-minute speech database was used as the training sequence. The shift-symmetric codebook optimized in one iteration was used as the starting codebook in the next one. The original codebook was a sequence of gaussian random numbers with unity variance. Note that we are evaluating NWSNR inside the training sequence, i.e. the codebook was optimized for that particular training database. However, since the speech database contained a large number of voices, it was expected to improve the performance of the coder outside the training sequence as well. After 13 iterations, the NWSNR was increased by 0.8 dB. Observe that the error did not converge to a local minimum due to the feedback from previous frames (see section 3.2.5). To evaluate the improvement in performance in terms of NWSNR outside the training sequence, we computed the NWSNR for an ensemble of random and optimized codebooks. The ensemble of optimized codebooks was obtained by running the GLA on several initial random codebooks. The same 10-minute speech database was used in the optimization process. In the second part of the experiment we computed the NWSNR of a 30-second speech sequence for all the codebooks. The statistical results showed an average NWSNR of 18.85 dB for BA-CELP with random codebooks and 19.58 for optimized codebooks. The error of the experiment was ± 0.2 dB with a confidence of 95%.

The tracking of the formant filter is closely related to the LLR distortion between the LPC models based on original and reconstructed speech. Perfect tracking implies zero

Figure 4.3 Noise-weighted signal-to-noise ratio versus η .

**Figure 4.4 Noise-weighted signal-to-noise ratio
for 15 iterations of the GLA algorithm.**

distortion. As a result, we can estimate the improvement in filter tracking by measuring the LLR distortion. Figure 4.5 shows the LLR distortion measure as a function of η . The dashed line corresponds to the LLR value for the conventional search. For $\eta=1$ the distortion is reduced by 4 dB.

The last experiment shows the quantitative influence of the mixed distortion measure. Given a 30-second segment of speech, the encoder chooses the excitation vector that minimizes the mixed distortion measure in each subframe. At the same time we check if the selected codevector (using the mixed distortion measure) corresponds to the optimum vector (if we were using NWMSE distortion). As a result, we can count the number of times in which the chosen vectors differ from the minimum NWMSE. Figure 4.6 indicates the percentage of suboptimum excitation vectors as a function of η . For values of η above 3, 10% of the blocks were not optimum in a NWMSE sense. For $\eta=1$ the coder uses suboptimum vectors in 25% of the cases. This means that the mixed distortion measure was “active” in 25% of the subframes.

Figure 4.5 Log-likelihood ratio versus η .

Figure 4.6 % of suboptimum vectors versus η .

Chapter 5

Summary and Suggestions for Future Research

5.1 Summary and Conclusions

In this thesis, we have found a new approach for reducing the delay of a CELP codec with almost no degradation in speech quality. This has been possible by using a backward-adaptive formant filter to avoid buffering of the input speech sequence. Pitch filter and gain coefficients, however, are transmitted in a forward-adaptive mode.

The tracking of the backward-adaptive filter has been found to be poor, when conventional MSE is used as distortion measure. The tracking problem is more profound when rapid changes of the spectrum occur. To alleviate this problem, a mixed distortion measure has been proposed. It combines conventional MSE minimization with look-ahead coding techniques. That is, it reduces the LPC distortion between original and backward-adaptive models one subframe into the future. It has been shown that the mixed distortion measure helps in filter tracking.

To mask the BA-CELP coder distortion, a generalized weighting filter has been used. This filter is based on original rather than synthesized speech. The weighting filter is essential for high quality speech coding. In addition, we have chosen a three-tap

Figure A.1 Distorted speech in terms of MNRUs.

forward-adaptive pitch filter to improve speech quality. The pitch filter parameters are computed in a closed-loop procedure and vector quantized. Informal listening tests have shown that the use of either a weighting filter based on synthesized speech or a one-tap predictor, clearly degrades speech quality.

In order to reduce computational complexity, we have adopted a shift-symmetric codebook. Further improvements have been possible after optimizing the codebook with a 10-minute database. An additional advantage of this codebook is that it has the same structure as the one-tap long-term predictor used to find the optimum pitch period. Therefore, it is possible to use the same algorithm to search for the optimum excitation vector and pitch lag. The resemblance between both search procedures may eventually simplify the practical implementation.

Formal listening tests have been conducted on the basis of CCITT recommendations on reference [13]. Results showed that 20 subjects did not find perceptible differences between forward- and backward-adaptive CELP coders. This is an encouraging result that indicates the feasibility of developing practical low-delay coders at 8 K b/s .

The impressive reduction in encoding delay has been gained at the expense of computational complexity and more sensitivity to transmission errors. The computational complexity is not viewed as a serious problem since continuous advances in processing technology are likely to provide more computational power at a lower cost. However, the proposed mixed distortion requires an additional LPC analysis for each excitation vector candidate. In this work, we used $n_c=16$ which leads to 16 extra LPC covariance method analyses per subframe. More investigation is necessary to reduce the number of computations.

In addition to the complexity problem, there is a need to investigate the performance of the coder under more realistic conditions. The BA-CELP coder is more sensitive to channel errors than the forward-adaptive coder. This is because the formant filter is

updated based on past reconstructed data. The propagation of the errors can severely affect the accuracy of the short-term predictor and therefore degrade speech quality.

5.2 Suggestions for Future Research

Further research is required before considering a practical implementations of the BA-CELP algorithm. The mixed-distortion measure has to be implemented in a more efficient way. Since the speech frames used in LPC analyses are highly overlapped, it is not practical to use block algorithms, such as the covariance method, to calculate the short term predictors. An interesting alternative would be to consider the use of recursive algorithms instead of block algorithms to eliminate redundant calculations.

A fast RLS algorithm, for example, can be used to implement the LPC computation. An exponential weight can be used to weigh the latest samples more than the oldest samples of the frame. This approach may improve tracking during periods in which the spectrum changes rapidly. The exponential weighting works as a forgetting factor that will make the algorithm more robust against channel errors.

Further reductions of the delay are possible by exploiting the correlation of the gain. In normal speech, the energy of the signal is expected to vary smoothly from one subframe to the next. Consequently, we can reduce the number of bits reserved for the gain by including a gain predictor. This predictor can be fixed or adaptive. The adaptive gain predictor would require side information to be transmitted through the channel. However, we can overcome this problem by sending gain information in a backward-adaptive manner. This would lead to shorter subframes and hence, lower delay.

The minimum possible delay can be achieved by sending all the parameters in a backward-adaptive mode. In this case all the parameters are derived from the excitation vector and no side information is required. The pure backward-adaptive CELP algorithm can reduce the encoding delay to 5 ms. However the codec will

become extremely sensitive to transmission errors which will affect speech quality in the practical implementation.

Appendix A

Subjective Evaluation of Speech Quality

A.1 Introduction

Subjective performance assessment is required in the design process of low bit-rate speech coders. This is because there is no replacement of the human perception mechanism. As a result, researchers have to conduct time consuming experiments to evaluate speech quality in terms of naturalness and intelligibility. The procedures used in the experiments have to be specified in great detail to prevent biased results. However, before conducting formal experiments it is preferred to evaluate the coder with objective measurements and informal subjective tests. Informal tests play an important role in the design of speech processing devices. They allow the designer to choose the systems that will be used in the formal experiment. In this appendix we will describe one of the methods recommended by the CCITT called *The Absolute Category Rating Test Method* [13]. In this method an ensemble of subjects classify a speech stimulus on a 5-point quality scale. Typical results show that the *Mean Opinion Score* ranges from 4-5 for high quality speech to 1-2 for very distorted speech. The stimulus material includes the processed speech and references. The references generally include high quality speech

like PCM-64 Kb/s and degraded speech. For the degraded signals we use the *Modulated Noise Reference Unit* [51]. Figure A.1 shows a circuit for the production of degraded speech in terms of MNRUs.

A.2 Absolute Category Rating Test Method

The objective of this method is to assign a subjective variable to the performance of a digital process. In this case we want to assign Q values (or modulated noise reference units) to a certain speech sample. The Q value in dB is a signal-to-multiplicative-noise ratio. As a result, we can measure the speech quality produced by a given coder under certain conditions in Q units. In this thesis, we have used Q values as a low-quality reference.

Signals processed with multiplicative noise are more appropriate than signals with additive noise as references. Signals processed with multiplicative noise are perceptually very similar in character to signals processed through digital systems. In addition to this they are simple to implement and considerable experience and information have been accumulated.

Several variables are involved in the experiments. Speech material, talker voice characteristic, presentation order, timing, signal level, environmental noise and so on may bias the outcome of the experiment unless they are controlled. In the next sections we summarize some of the procedures described in the CCITT recommendation.

A.2.1 Preparation of Source Tapes

- **Tape Recorder.** Use studio quality audio equipment. Use A/D converter and a television cassette recorder for storing high quality source and test tapes.
- **Calibration Tones.** Used to enable checking of the sensitivity/frequency characteristics.
- **Sending System.** Specifies the telephony system.

Figure A.1 Distorted speech in terms of MNRUs.

- **Recording Environment.** Specifies environmental noise level and spectrum.
- **Speech Samples.** The source tape should include all the samples. The total number of units is equal to *No of talkers* times *No of conditions*. Where the number of talkers should be at least 4. Conditions are training, reference and test samples. Each sample should comprise 2 or 3 sentences separated by 1 second. The total duration of a sample should be 6-10 seconds and the separation between samples should be 5 seconds.
- **Talkers.** Use at least 4 different talkers with different voice characteristics.

A.2.2 Processing of the Stimulus Tapes

- **Test Conditions.** Specify type of codec, bit-rate, parameters, etc.
- **Reference Conditions.** Include Q values in the range 5 dB to 25 dB. Include other references such as PCM.
- **Stimulus Tape.** The stimulus tape should begin with 12 practice conditions and then the test and reference conditions. Include all the talkers in the practice conditions. The test and reference part will be *No of talkers* times *No of conditions*. The order of the units should be random. There should be at least 2 stimulus tapes with different random order. It would be better to have a different order for each subject.

A.2.3 Testing Procedure

- **Listeners.** The preferred number of listeners is 32 with a minimum of 12. The subjects should represent a typical user population in terms of gender, ages, hearing, etc.
- **Listening Environment.** The ambient room noise level should be ≤ 40 dBA simulating a quiet office or a living room.
- **Test Instructions.** The subjects will be provided with written instructions which will also be read before the practice conditions. Subjects should not be instructed

that the practice conditions represent the full range of quality to be encountered in the test. Between the practice and test conditions the administrator should answer all possible question asked by the subjects. The subjects should be instructed to rate the conditions according to a 5-point quality scale shown in table A.1.

A.2.4 Presentation of the Results

Mean opinion scores, standard deviations, 95% confidence intervals and other statistics should be reported. Technical data of the codecs and test environment may also be included. Description of the talkers and listeners may be necessary in certain cases.

Score	Quality Ratings
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table A.1 Five-point quality scale.

References

- [1] R. V. Bruce, *Bell: Alexander Graham Bell and the Conquest of Solitude*. Little, Brown and Company, 1973.
- [2] J. G. Proakis, *Digital Communications*. McGraw-Hill, 2nd ed., 1989.
- [3] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Prentice-Hall, 1984.
- [4] E. A. Lee and D. G. Messerschmitt, *Digital Communications*. Kluwer Academic Publishers, 1988.
- [5] R. H. Moffet, "Echo and Delay Problems in Some Digital Communication Systems," *IEEE Communications Magazine*, pp. 41-47, 1987.
- [6] K. Murano, S. Unagami, and F. Amano, "Echo Cancellation and Applications," *IEEE Communications Magazine*, pp. 49-55, 1990.
- [7] E. A. Lee, "Programmable DSP Architectures," *IEEE ASSP Magazine*, pp. 4-19, October 1988.
- [8] C. E. Shannon, "A Mathematical Theory of Communications," *Bell Systems Technical Journal*, vol. 27, pp. 379-423, 623-656, 1948.
- [9] J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech," *Proceedings of the IEEE*, vol. 73, pp. 1551-1588, November 1985.
- [10] W. P. LeBlanc, "An Advanced Speech Coder Based on a Rate-Distortion Theory Framework," Master's thesis, Carleton University, Ottawa, ON, Canada, March 1988.
- [11] T. Berger, *Rate-Distortion Theory*. Prentice-Hall, 1971.
- [12] J. H. Chen, "A Robust Low-Delay CELP Speech Coder," in *IEEE Global Communications Conference*, pp. 1237-1241, 1989.
- [13] CCITT Supplement No. 14 to Recommendation P.81, Blue Book, *Subjective Performance Assessment of Digital Processes Using the Modulated Noise Reference Unit*, 1988.
- [14] J. L. Flanagan, *Speech Analysis Synthesis and Perception*. Springer-Verlag, 1972.
- [15] D. O'Shaughnessy, *Speech Communication*. Addison-Wesley, 1987.
- [16] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*. Prentice-Hall, 1978.
- [17] J. D. Markel and A. H. Gray, *Linear Prediction of Speech*. Springer-Verlag, 1976.
- [18] J. Makhoul, "Linear Prediction: A Tutorial Review," *Proceedings of the IEEE*, vol. 63, pp. 561-580, April 1975.
- [19] S. Haykin, *Adaptive Filter Theory*. Prentice-Hall, 1986.
- [20] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 2nd ed., 1984.

- [21] 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, June 1979.
- [22] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, "Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter," *Proceedings of the IEEE*, vol. 64, pp. 1151-1162, August 1976.
- [23] J. M. Cioffi and T. Kailath, "Fast, Recursive-Least-Squares Transversal Filters for Adaptive Filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, pp. 304-337, April 1984.
- [24] R. Ramachandran and P. Kabal, "Pitch Prediction Filters in Speech Coding," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-37, pp. 467-478, April 1989.
- [25] A. H. Gray and J. D. Markel, "Distance Measures for Speech Processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-24, pp. 380-391, October 1976.
- [26] R. M. Gray, A. Buzo, A. H. Gray, and Y. Matsuyama, "Distorsion Measures for Speech Processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-28, pp. 367-376, August 1980.
- [27] F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-23, pp. 145-150, Febraury 1975.
- [28] Y. Matsuyama and R. M. Gray, "Voice Coding and Tree Encoding Speech Compression Systems Based Upon Inverse Filter Matching," *IEEE Transactions on Communucations*, vol. COM-30, pp. 711-720, April 1982.
- [29] B. H. Juang, "On the Hidden Markov Model and Dynamic Time Wrapping for Speech Recognition-A Unified View," *AT&T Bell Laboratories Technical Journal*, vol. 63, September 1984.
- [30] B. H. Juang, "On Using the Itakura-Saito Measures for Speech Coder Performance Evaluation," *AT&T Bell Laboratories Technical Journal*, vol. 63, pp. 1477-1498, October 1984.
- [31] R. E. Crochiere, J. M. Tribolet, and L. R. Rabiner, "An Interpretation of the Log Likelihood Ratio as a Measure of Waveform Coder Performance," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-28, pp. 318-323, June 1980.
- [32] J. Chen, G. Davidson, and K. Zeger, "Speech Coding for the Mobile Satellite Experiment," in *IEEE International Conference on Communicaions*, (Seattle, Washington), pp. 756-763, 1987.
- [33] A. Buzo, A. H. Gray, R. M. Gray, and J. D. Markel, "Speech Coding Based Upon Vector Quantization," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-28, pp. 562-574, October 1980.

- [34] B. H. Juang, D. Y. Wong, and A. H. Gray, "Distortion Performance of Vector Quantization for LPC Voice Coding," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-30, pp. 294–303, April 1982.
- [35] N. S. Jayant, "High-Quality Coding of Telephone Speech and Wideband Audio," *IEEE Communications Magazine*, pp. 10–20, 1990.
- [36] B. S. Atal, "High-Quality Speech at Low Bit Rates: Multi-pulse and stochastically Excited Linear Predictive Coders," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, (Tokio, Japan), pp. 1681–1684, 1986.
- [37] B. S. Atal and J. R. Remde, "A New Model of LPC Excitation for Producing Natural-Sounding Speech at Low Bit Rates," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, (Paris, France), pp. 614–617, 1982.
- [38] M. R. Schroeder and B. S. Atal, "Code-Excited Linear Prediction (CELP): High-Quality Speech at Very Low Bit Rates," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, (Tokio, Japan), pp. 937–940, 1985.
- [39] R. M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, pp. 4–29, April 1984.
- [40] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*. Prentice Hall, 1989.
- [41] P. Kabal, J. L. Moncet, and C. C. Chu, "Synthesis Filter Optimization and Coding: Applications to CELP," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, (New York), pp. 147–150, 1988.
- [42] I. M. Trancoso and B. S. Atal, "Efficient Procedures for Finding the Optimum Innovation in Stochastic Coders," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, (Tokio, Japan), pp. 2375–2378, 1986.
- [43] W. B. Kleijn, D. J. Krasinski, and R. H. Ketchum, "Improved Speech Quality and Efficient Vector Quantization in SELP," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, (New York), pp. 155–158, 1988.
- [44] G. Davidson and A. Gersho, "Complexity Reduction Methods for Vector Excitation Coding," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, (Tokio, Japan), pp. 3055–3058, 1986.
- [45] Y. Linde, A. Buzo, and R. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications*, vol. COM-28, pp. 84–95, January 1980.
- [46] W. P. LeBlanc and S. A. Mahmoud, "Structured Codebook Design in CELP," in *Second International Mobile Satellite Conference*, June 1990.
- [47] F. K. Soong and B. H. Juang, "Line Spectrum Pair (LSP) and Speech Data Compression," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1.10.1–4, 1984.
- [48] F. K. Soong and B. H. Juang, "Optimal Quantization of LSP Parameters," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, (New York), pp. 394–397, 1988.

- [49] P. Kabal and R. Ramachandran, "The Computation of Line Spectral Frequencies using Chebyshev Polynomials," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, December 1986.
- [50] V. Cuperman, A. Gersho, R. Pettigrew, J. J. Shynk, and J. H. Yao, "Backward Adaptation for Low Delay Vector Excitation Coding of Speech at 16 Kbit/s," in *IEEE Global Communications Conference*, pp. 1242-1246, 1989.
- [51] CCITT Supplement No. 14 to Recommendation P.81, Blue Book, *Modulated Noise Reference Unit (MNRU)*, 1988.

END

1 2 0 4 9 1

CIN