SPEECH CODING FOR PACKETIZED NETWORKS

by

Aamir Husain

B.Sc. (Hons) Loughborough University of Technology, U.K, 1987

M.S.E.E. California Institute of Technology, U.S.A, 1990

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in the School
of
Engineering Science

SIMON FRASER UNIVERSITY

July 1996

ISBN 0-612-16925-1

PARTIAL COPYRIGHT LICENSE

Title of Thesis/Project/Extended Essay

"Speech Coding for Packetized Networks"

Author:

(signature)

M.A. Husain
(name)

July 15, 1996
(date)

APPROVAL

Name: Aamir Husain

Degree: Doctor of Philosophy

Title of thesis : Speech Coding for Packetized Networks

Examining Committee: Dr. S. Stapleton, Chairman
Professor, Engineering Science, SFU

Senior Supervisor: Dr. Vladimir Cuperman
Professor, Engineering Science, SFU

Supervisor: Dr. Paul Ho
Associate Professor, Engineering Science, SFU

Supervisor: Dr. Jacques Vaisey
Assistant Professor, Engineering Science, SFU

Internal Examiner: Dr. Steve Hardy
Professor, Engineering Science, SFU

External Examiner: Dr. Bishnu Atal
Dept Head, AT&T Bell Labs, U.S.A

Date Approved: 15 July 1996

Abstract

The objective of this thesis is speech coding for packet networks. A common aspect of speech transmission through packetised networks is the need to consider the discarded (missing) packets as a result of error detection or network overload. The missing packets and the possible mistracking that results in the speech decoder lead to significant quality degradation. In this thesis, we introduce a packet recovery technique for code excited linear prediction (CELP) based speech coders.

The proposed technique independently extrapolates the excitation signal and the short-term synthesis filter. A recovery strategy based on speech classification (voiced, unvoiced, transition, silence) is discussed. The extrapolation of the short-term filter uses a least-squares fading memory polynomial filter applied to reflection coefficients.

In network environment, we have many possible sources of delay, hence low-delay coders may be required. In this research, two approaches towards achieving a high quality low-delay speech coder at 8 kb/s have been developed: a backward 8 kb/s coder and a partially-forward 8 kb/s coder. The effect of coefficient adaptation on speech quality under clean and noisy channel conditions is also investigated, using various driving signals.

The quality of the low-delay codecs developed are comparable to the 8 kb/s VSELP codec in clean conditions. At bit error rates of 10^{-3}, both systems achieve MOS scores which were within 0.2 on the MOS scale from the scores obtained in clean conditions. The backward system, by making use of a different driving signal, results in a robust codec which achieves good subjective quality, even at BER as high as 10^{-2}.

Objective and subjective quality evaluations of the missing packet recovery model applied to the low-delay CELP (LD-CELP) G.728 standard and a variable-rate CELP (VR-CELP) system for random and burst block erasures are presented. The results indicate that the packet recovery system is robust up to a block erasure rate of 10%. Very little degradation in quality was observed at erasure rates up to 3%.

To my family for their love and support always
and to the memory of my dear friend Navid.

Acknowledgements

I would like to express my gratitude to my supervisor Dr. Vladimir Cuperman for his guidance throughout the course of this research as well as providing useful suggestions during the writing of this thesis. I would also like to thank the various organizations that provided financial assistance during my studies. They include NSERC, Rockwell International, Simon Fraser University and the Centre for Systems Science.

My thanks also to Mr. Mike Callender for providing me with information on standardization activity as well as providing the initial idea towards packet reconstruction techniques through discussions with him. I would also like to thank Dr. Vijay Varma of Bellcore for providing me with a packet loss model and Dr. Spiros Dimolitsas for providing LD-CELP verification software and standardization activity information. I would like to thank Drs. Paul Ho and Jacques Vaisey for useful comments during presentations of my research material.

A sincere word of thanks to my colleagues especially Bhaskar, Peter and Yingbo, in the speech research lab. They not only provided me with useful ideas and suggestions but were there to provide assistance when the chips were down (or for that matter when the code was bugged). A general thanks to all those who volunteered for speech evaluation tests done during the course of this research. A word of thanks to Sanjay Gupta for printing the colour plots in this thesis.

A very special thanks to Brigitte Rabold who has been a great help during my stay here and has always there to help keep my chin up. Thanks Brigitte for your constant support. I would also like to thank the office staff (Marilyn, Lesley and Jackie) for going the extra mile in helping me out. My thanks also to the computing support staff especially Chao, Ken and Frank for their help in keeping the code running.

Finally, a special thanks to my friends for providing lighter moments in an otherwise hectic research endeavour especially my cricketing buddies in keeping the weekends busy over the summer.

Contents

List of Tables

List of Figures

List of Symbols

E	Expectation operator
pdf	probability density function
$R(D)$	Rate distortion function
$D(x,y)$	Average distortion
$A(z)$	short-term prediction (inverse) filter
$A_p(z)$	symmetrical m+1 order polynomial
$A_q(z)$	antisymmetrical m+1 order polynomial
$B(z)$	long-term prediction filter
$H(z)$	weighted short-term synthesis filter
H	linear prediction filter operation
$\mathbf{H}_w$	weighted linear prediction impulse response matrix
$P(z)$	Post filter
$W(z)$	Perceptual weighting filter
$W_1(z)$	10-th order transversal all zero filter
W	weighting filter operation
$\mathbf{W}$	weighting matrix
a_j	j-th short-term prediction coefficient
b_j	j-th long-term prediction coefficient
k_j	j-th order reflection coefficient
e_j	j-th order forward prediction error
r_j	j-th order backward prediction error
μ	step size in predictor adaptation
ν	softening factor in backward block adaptation
λ	leakage factor in predictor adaptation
α	numerator in lattice filter adaptation

β	denominator in lattice filter adaptation
υ	exponential fading memory term in lattice adaptation
σ	signal variance in predictor adaptation
δ	running average memory
γ_1	numerator bandwidth expansion coefficient
γ_2	denominator bandwidth expansion coefficient
Γ	highpass spectral tilt factor in post filter
ρ	normalised autocorrelation function
m	predictor (short- or long-term) order
$x(n)$	input (or original) signal
$\tilde{x}(n)$	quantized or predicted signal
$c(n)$	codevector excitation signal
$e(n)$	residual error signal
$u(n)$	quantized residual error signal
$w(n)$	quantized residual error signal with pitch periodicity
$p(n)$	quantized pitch periodic error signal
$y(n)$	output (or reconstructed) signal
$s(n)$	windowed input signal
r_k	autocorrelation function at lag k
$\mathbf{r}_x$	autocorrelation function of x (vector form)
$\mathbf{R}_{xx}$	autocorrelation matrix of x
ψ	autocovariance function of s (vector form)
$\mathbf{\Psi}$	autocovariance matrix of s
k_p	pitch period
g_p	pitch predictor or adaptive codebook gain
g	excitation gain
$\mathbf{t}$	target vector for 1st (usually adaptive) codebook search
$\tilde{\mathbf{t}}$	target vector for subsequent codebook search
$\mathbf{e}_w$	error vector after subtracting 1st stage contribution
$\mathbf{e}_{pw}$	error vector after subtracting subsequent stage contribution
$\mathbf{z}$	unscaled adaptive codebook contribution through weighted short-term filter

$\mathbf{y}_{wzir}$	weighted ZIR reconstruction vector
$\mathbf{y}_{wzsr}$	weighted ZSR reconstruction vector

List of Abbreviations and Acronyms

ADPCM	Adaptive Differential Pulse Code Modulation
ATM	Asynchronous Transfer Mode
A-by-S	Analysis-by-Synthesis
BER	Bit Error Rate
BLR	Block Erasure (Loss) Rate
BISDN	Broadband Integrated Services Digital Network
CDMA	Code Division Multiple Access
CELP	Code Excited Linear Prediction
CS-ACELP	Conjugate Structure Algebraic Code Excited Linear Prediction
DoD	Department of Defense
DSI	Digital Speech Interpolation
ETSI	European Telecommunications Standard Institute
FEC	Forward Error Correction
FER	Frame Erasure Rate
FIR	Finite Impulse Response
FPLMTS	Future Public Land Mobile Telecommunications System
GSM	Global System for Mobile Telecommunications
HDLC	High-Level Data Link Control
ITU	International Telecommunications Union
ITU-T	International Telecommunications Union Telecommunications Sector
ITU-R	International Telecommunications Union Radiocommunications Sector
LD-CELP	Low Delay Code Excited Linear Prediction
LD-VXC	Low-Delay Vector Excitation Coding

LLD-VXC	Lattice Low-Delay Vector Excitation Coding
LMS	Least Mean Squares
LPC	Linear Prediction Coding
LSF	Line Spectral Frequencies
LSP	Line Spectral Pairs
MBE	Multi-Band Excitation
MSE	Mean-Square Error
MOS	Mean Opinion Score
PCM	Pulse Code Modulation
PCS	Personal Communications Services
PRM	Packet Recovery Model
PRMA	Packet Reservation Multiple Access
PSI-CELP	Pitch Synchronous Innovation Code Excited Linear Prediction
PSTN	Public Switched Telephone Network
PVP	Packet Voice Protocol
QCELP	Qualcomm Code Excited Linear Prediction
SNR	Signal-to-Noise Ratio
SegSNR	Segmental Signal-to-Noise Ratio
STC	Sinusoidal Transform Coder
TASI	Time Assignment Speech Interpolation
TDMA	Time Division Multiple Access
TIA	Telecommunications Industry Association
VAD	Voice Activity Detection
VAR-CELP	Variable Rate Code Excited Linear Prediction
VQ	Vector Quantization
VSELP	Vector Sum Excited Linear Prediction
VXC	Vector Excitation Coding
ZIR	Zero Input Response
ZSR	Zero State Response

Chapter 1

Introduction

Speech coding has recently seen a renewed explosion of interest and activity. This is primarily attributed to the rapid development of very large scaled integrated circuit technology that has enabled cost effective implementation of new technology. Wireless personal communications offers a significant role for speech coding technology to mature, as a result of the tremendous increase in demand for personal communication services (PCS), which is expected to continue well into the next millennium.

Part of the challenge in planning future wireless systems is to determine the services they will be required to support. This has been the major thrust of the International Telecommunications Union (ITU), which is defining future public land mobile telecommunications systems (FPLMTS).

Fast packet switching systems for T1 transmission and asynchronous transfer mode (ATM) for broadband integrated services digital network (BISDN) systems have motivated interest in packetized speech transmission in wired networks. Next generation personal communication networks will be required to co-exist with fibre-optic based broadband communication networks (such as B-ISDN/ATM) in a transparent and efficient manner.

1.1 Speech Coding

Speech coding involves the transformation of a continuous time and continuous amplitude speech signal into a discrete time and discrete amplitude signal. This is a lossy transformation, whereby the signal samples are never recovered exactly after

decoding. The objective of the speech coding system is to reduce the bandwidth required to transmit a speech signal in digital form, and this is achieved by decreasing the bit-rate whilst attempting to preserve speech quality.

There are various criteria involved in the design of a speech coding system. Some of the most important criteria include:

- perceptual speech quality
- transmission bit rate
- computational complexity
- communications delay
- robustness to channel impairments

The aim of the speech coding researcher is to design a system, suitably trading-off these criteria depending on the desired application of the speech coding system. In this thesis, we pay particular attention to codec robustness and delay whilst attempting to preserve speech quality.

Communication delay has become an important performance criterion for speech coders, as efforts are being intensified to achieve toll quality at rates as low as 4 kb/s, to replace existing higher rate systems. Delay may necessitate the use of echo cancellation, and in some applications it remains an impairment even after echo cancellation is performed. For example, in a network environment there are many possible sources of delay (*i.e.* transmission, propagation, queuing *etc*), hence low-delay coders may be required. A critical consideration in low-delay coding has been codec robustness in noisy channels.

In this research, two approaches towards achieving a high quality low-delay speech coder at 8 kb/s have been developed: a backward 8 kb/s coder, which makes use of a 3-tap hybrid backward adaptive open-loop pitch predictor and a partially-forward scheme, which uses a 3-tap forward adapted long-term adaptive codebook. Also investigated, is the effect of coefficient adaptation on speech quality under clean and noisy channel conditions using various synthesis driving signals.

The development of a low-rate high quality low-delay speech coder is a precursor to the primary motivation of this thesis: the study of packet recovery techniques in code

excited linear prediction (CELP) based speech coders. After having examined the effect encoder modifications have on codec robustness under noisy and clean conditions, the thesis addresses the issue of: how can the decoder be better designed to handle errors, both random as well as bursts, for blocks of speech information.

1.2 Packetized Speech Coding

Packet techniques offer significant benefits for voice as well as for data communications. The integration of digital voice with data in a common packet-switched system offers potential cost savings through sharing of switching and transmission resources, as well as enhanced services for users who require access to both voice and data communications. Packet networks also provide a system environment for effective exploitation of variable bit-rate voice transmission techniques, either to reduce average end-to-end bit-rate or to dynamically adapt voice bit-rate to network conditions.

Block erasures may appear in packetized systems due to network overload conditions. Speech blocks may be lost due to buffer overflow or they may be dropped in the network nodes by the congestion control mechanism. The congestion control operates as follows: newly arriving packets are blocked if the number already present at an input queue exceed some threshold. A distinction being made between input packets (packets newly arriving at a node) and transmit packets (those already in the network and arriving from another node). Another source of block erasures could be fixed delays due to transmission and propagation, as well as statistically varying delays, such as long queuing delay in nodes, which result in packets not arriving at their destination within a prescribed time and being considered as lost. Additional varying delays components are caused by packet retransmissions to compensate for errors in delivery. For real-time voice communication with a low-delay constraint (*i.e.* packet retransmissions being minimized), some reliability needs to be sacrificed by tolerating a small percent of lost packets [102].

Portable radio channels experience signal fluctuations (fades) caused by multi-path signal additions from different propagation paths. The effect of fading is to produce errors in bursts when the received signal envelope fades below some noise related threshold, resulting in a channel that is either very good (no errors) or very bad (error bursts). Fading leads to a situation where error detection may be preferable to

forward error correction (FEC) and the corresponding channel may be characterized by block erasures.

The telecommunication standardization sector of the International Telecommunication Union (ITU-T) has recently standardized a 16 kb/s speech coding algorithm. The system chosen was low-delay code excited linear prediction (LD-CELP). The ITU-T is currently in the process of standardizing an 8 kb/s speech coding algorithm. Personal communications services (PCS) is a major application envisioned for the 8 kb/s standard. The 16 kb/s LD-CELP algorithm is likely to be employed in the early phase of PCS, until lower rate standards are fully established.

Codecs can be evaluated using a random errors channel model which is an acceptable method for wired networks. However, speech coders operating in a personal communications environment also need to address bursty errors in order to achieve acceptable performance in applications [21]. The random error assumption can still be applied if burst errors can be randomized by using techniques such as bit interleaving. In the vehicular mobile environment, bit interleaving to randomize bursty errors may be achieved within acceptable delay due to short fade and inter-fade durations. On the other hand, in PCS systems, due to the slow fading nature of the channel, a substantial amount of bit interleaving would be required to randomize errors and this results in unacceptable delay.

In this thesis, missing packet recovery techniques based on speech classification and spectral extrapolation are examined. The recovery system extrapolates independently the excitation signal and the short-term synthesis filter using an extrapolation strategy based on speech classification (voiced, unvoiced, transition, silence). The extrapolation of the short-term filter uses a least-squares fading memory polynomial filter applied to reflection coefficients.

1.3 Thesis Objectives

The objective of this thesis was speech coding for packet networks. The main objective was to examine codec robustness under block erasures on speech quality in CELP-based speech coders as well as to develop a high-quality low-delay coder that is robust under bit errors.

The main contributions of the research are:

- Recovery techniques based on speech classification and spectral extrapolation are developed for CELP-based speech coders [41, 40, 42, 43]. A recovery strategy based on speech classification (voiced, unvoiced, transition, silence) was developed. Recovery techniques which extrapolated independently the excitation signal and the short-term synthesis filter were developed. Transition recovery was improved by developing a model to recalculate the extrapolated excitation, which due to packet errors was corrupted as a result of absent or misplaced pitch pulses.

- Two approaches towards achieving a high quality speech coder at 8 kb/s have been developed, trading off communication delay and speech quality. Also the effect of coefficient adaptation on speech quality under clean and noisy channel conditions is investigated [38, 39].

1.4 Thesis Outline

This thesis is organized into six chapters. Chapter 2 provides a brief overview of speech coding fundamentals which are considered important concepts necessary to facilitate a better understanding of the thesis. In Chapter 3, low-delay coding is discussed in detail with particular emphasis being placed on 8 kb/s lattice low-delay vector excitation coding (LLD-VXC) studies undertaken and a review of the 16 kb/s LD-CELP speech coder and earlier work on low-delay vector excitation coding (LD-VXC) and LLD-VXC at 16 kb/s.

Chapter 4 discusses some critical issues in packet based speech coders. Also presented is an overview of variable-rate speech coding together with a brief description of a variable-rate CELP (VR-CELP) coder tested under the packet recovery model. Packet recovery techniques in CELP based speech coders are presented in Chapter 5. Finally, conclusions are presented in Chapter 6.

Chapter 2

Speech Coding Fundamentals

This chapter provides a brief overview of speech coding fundamentals which are considered important concepts necessary to facilitate a better understanding of the thesis. The chapter is organized into six sections. Section 2.1 provides a concise introduction to rate distortion theory. In Section 2.2, a summary of scalar and vector quantization is provided. Section 2.3 presents the linear prediction model for speech. In Section 2.4, quality measures useful for measuring the subjective quality of speech are discussed. Speech coding systems are reviewed briefly in Section 2.5, with an overview of the state of the art coding techniques being presented for completeness to give the reader a feel for the latest coding strategies being actively studied. Finally, the CELP coding technique is presented in Section 2.6.

2.1 Rate Distortion

By applying the sampling theorem, the output of an analog source is converted to an equivalent discrete-time sequence of samples. The samples are then quantized and encoded. Quantization of the amplitudes of the sampled signal results in data compression but also introduces some distortion of the waveform or loss in fidelity. By the term "distortion" we mean some measure of the difference between the actual source x_k and the corresponding quantized values $\tilde{x}_k$, which we denote by $d\{x_k, \tilde{x}_k\}$. For example a commonly used distortion measure is the *squared-error distortion*,

defined as

$$d(x_k, \tilde{x_k}) = (x_k - \tilde{x_k})^2 \tag{2.1}$$

Once a distortion measure has been chosen and if $d\{x_k, \tilde{x_k}\}$ is the distortion measure per letter, the distortion between a sequence of n samples $\mathbf{x}_n$ and the corresponding n quantized values $\tilde{\mathbf{x}_n}$ is the average over the n source output samples

$$d(\mathbf{x}_n, \tilde{\mathbf{x}_n}) = \frac{1}{n} \sum_{k=1}^{n} d(x_k, \tilde{x_k}) \tag{2.2}$$

The source output is a random process, and, hence, the n samples in $\mathbf{x}_n$ are random variables. Therefore, $d(\mathbf{x}_n, \tilde{\mathbf{x}_n})$ is a random variable. Its expected value is defined as the average distortion

$$D = E[d(\mathbf{x}_n, \tilde{\mathbf{x}_n})] = \frac{1}{n} \sum_{k=1}^{n} E[d(x_k, \tilde{x_k})] = E[d(x, \tilde{x})] \tag{2.3}$$

where the last step follows from the assumption that the source output process is stationary.

Suppose we have a memoryless source with a continuous-amplitude output $\mathbf{x}$ that has a probability density function (*pdf*) $p(x)$, a quantized amplitude output alphabet $\tilde{\mathbf{x}}$, and a per letter distortion measure $d(x, \tilde{x})$, where $x \in \mathbf{x}$ and $\tilde{x} \in \tilde{\mathbf{x}}$. Then, the minimum rate in bits per source output that is required to represent the output $\mathbf{x}$ of the memoryless source with a distortion less than or equal to D is called the rate-distortion function $R(D)$ and is defined as

$$R(D) = \min_{p(\tilde{x}_i|x): E[d(\mathbf{x}, \tilde{\mathbf{x}})] \leq D} I(\mathbf{x}, \tilde{\mathbf{x}}) \tag{2.4}$$

where $I(\mathbf{x}, \tilde{\mathbf{x}})$ is the average mutual information between $\mathbf{x}$ and $\tilde{\mathbf{x}}$ and is given by

$$I(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{i=1}^{L} \int_{\infty}^{\infty} p(\tilde{x}_i|x) P(x) log \frac{p(\tilde{x}_i|x)}{p(\tilde{x}_i)} dx \tag{2.5}$$

L here determines the number of outcomes of the output $\tilde{\mathbf{x}}$, and $p(\tilde{x}_i|x)$ is the probability of x being reproduced as $\tilde{x}_i$. $I(\mathbf{x}, \tilde{\mathbf{x}})$ is a function of $p(\tilde{x}_i|x)$. In general, the rate $R(D)$ decreases as D increases or, conversely, $R(D)$ increases as D decreases.

The rate distortion function $R(D)$ of a source is associated with the following basic source coding theorem in information theory attributed to Shannon.

> "There exists an encoding scheme that maps the source output into codewords such that for any given distortion D, the minimum rate $R(D)$ bits per sample is sufficient to reconstruct the source output with an average distortion that is arbitrarily close to D"

It is clear that the rate distortion function $R(D)$ for any source represents a lower bound on the source rate that is possible for a given level of distortion. In deriving a rate-distortion function, sources are assumed stationary and to have a given *pdf*. Speech is a non-stationary process, its rate-distortion function for a general fidelity criterion is still an unsolved problem. Perceptual-based criteria which are important in low-rate speech coding are difficult to apply to the mathematical workframe of the rate-distortion theory. The implications of which are an inability to accurately obtain a lower bound on the source rate to achieve a given level of distortion . For more information on rate-distortion theory and entropy, the reader can refer to [49, 83, 5].

2.2 Quantization

As discussed earlier, quantization is a mapping procedure which results in a loss of information. In this section, scalar and vector quantization will be briefly reviewed. For more detailed information Jayant and Noll provide an excellent review of earlier work on quantization [49] and Gersho and Gray gives a comprehensive overview of the state of the art in vector quantization [28].

2.2.1 Scalar Quantization

A scalar quantizer is a many-to-one mapping of the real axis into a finite set of real numbers $y_k, k = 1, 2, ..., L$. If the input signal is x, the quantizer mapping Q, and the quantizer output points y_k, the quantizer equation becomes

$$Q(x) = y \quad \text{where} \quad y \in y_1, y_2, ..., y_L \tag{2.6}$$

The choice of output points are done so as to minimize a distortion criteria $d(x, y)$. The quantizer equation now becomes

$$Q(x) = y_k \quad \text{where} \quad k = ARGMIN_j d(x, y_j) \tag{2.7}$$

where, $ARGMIN_j$ returns the value of the argument j for which a minimum is obtained.

The design of optimal scalar quantizers is performed using the Lloyd-Max algorithm [66, 71], which is discussed in detail in most texts on quantization [49, 28]. Scalar quantizers can be optimized in such a way that for a given input probability density function and a given number of levels, they give a minimal quantization error variance. These quantizers are called optimal quantizers. The quantization error is given by

$$e = x - Q(x) \tag{2.8}$$

with the variance of the quantization error as

$$\sigma_e^2 = E(e^2) = E(x - Q(x))^2 \tag{2.9}$$

The necessary conditions for the minimum are obtained by straightforward differentiation with respect to y_k and the interval boundary values. The necessary conditions for optimality can be shown to be

$$x_k = \frac{1}{2}(y_k + y_{k+1}) \quad \text{for} \quad k = 1, 2, .., L-1 \tag{2.10}$$

$$y_k = E\{x | x \in [x_{k-1}, x_k]\} \quad \text{for} \quad k = 1, 2.., L \tag{2.11}$$

Equation 2.10 states that the interval boundaries must lie halfway between the reconstruction values. Note that equation 2.10 is a particular case of equation 2.7 in the general case. Equation 2.11 means that the quantization level is the centroid of the quantization interval. Equation 2.11 can be written as

$$y_k = \frac{\int_{x_{k-1}}^{x_k} x p_y(x) dx}{\int_{x_{k-1}}^{x_k} p_y(x) dx} \quad \text{for} \quad k = 1, 2, .., L \tag{2.12}$$

Analytical solutions for equations 2.10 and 2.11 can only be found for $L = 2, 3$. Lloyd proposed an iterative algorithm that can be used to obtain a numerical solution. The key steps involved are:

1. For a given centroids set, obtain the optimal decision boundaries that satisfy equation 2.10.

2. For a given set of decision regions, obtain the optimal centroids satisfying equation 2.12.

Lloyd's technique as described above is not suitable for speech signals as the *pdf* used in computing equation 2.12 is not known. Max, however, introduced a technique for signals represented by training sequences, based on computing the conditional expectation as a simple average, clustering all input points that lay in any given interval, thus not requiring knowledge of the *pdf*.

2.2.2 Vector Quantization

The vector quantization (VQ) problem is part of the general pattern-recognition problem of classification of data into a number of categories that optimize some fidelity criterion. Indeed, in the design of vector quantizers, one often employs well known techniques from pattern recognition. The theoretical foundation of vector quantization comes from Shannon's theory on rate-distortion, which suggests that better compression can be achieved by coding vectors instead of scalars.

Vector quantization involves the joint quantization of a block of signal parameters or samples. If the input signal is an n-dimensional vector $\mathbf{x} = [x_1 x_2 ... x_n]$ with real-valued, continuous amplitude components $\{x_i, 1 \leq i \leq n\}$ that are described by a joint *pdf* $p(\mathbf{x})$, the quantizer mapping is Q and the quantizer output is an n-dimensional vector $\mathbf{y}$ with components $\{y_i, 1 \leq i \leq n\}$, the quantizer equation becomes

$$Q(\mathbf{x}) = \mathbf{y} \tag{2.13}$$

The n-dimensional space is partitioned into L cells $C_k, 1 \leq k \leq L$. All input vectors that fall in cell C_k are quantized to the vector $\mathbf{y}_k$.

Quantization introduces a distortion $d(\mathbf{x}, \mathbf{y})$. The average distortion over the set of input vectors $\mathbf{x}$ is

$$D = \sum_{k=1}^{L} P(\mathbf{x} \in C_k) E[d(\mathbf{x}, \mathbf{y}_k) | \mathbf{x} \in C_k] \tag{2.14}$$

$$= \sum_{k=1}^{L} P(\mathbf{x} \in C_k) \int_{\mathbf{x} \in C_k} d(\mathbf{x}, \mathbf{y}_k) p(\mathbf{x}) d\mathbf{x} \tag{2.15}$$

where $P(\mathbf{x} \in C_k)$ is the probability that the vector $\mathbf{x}$ falls in the cell C_k and $p(\mathbf{x})$ is the joint *pdf* of the n random variables. As in scalar quantization, D is minimized by

selecting the L cells $\{C_k, 1 \leq k \leq L\}$ and corresponding centroids $\mathbf{y}_k$ for a given *pdf* $p(\mathbf{x})$. There are two conditions for optimality. The first is that the optimal quantizer employ a nearest neighbour selection rule, which can be expressed as

$$C_k = \{\mathbf{x} : d(\mathbf{x}, \mathbf{y}_k) \leq d(\mathbf{x}, \mathbf{y}_j) \text{ for } 1 \leq j \leq L\} \tag{2.16}$$

This condition states that all input vectors that lie in the k-th cell are as close to $\mathbf{y}_k$ as to any other codevector. Note that when the distortion between an input vector and two or more codevectors is equal, the input vector is on a cell boundary and must be assigned to a unique codevector (tie breaking rule). The second condition for optimality is that each output vector $\mathbf{y}_k$ be chosen to minimize the average distortion in cell C_k. So $\mathbf{y}_k$ is that vector $\mathbf{y}$ which minimizes

$$D_k = E[d(\mathbf{x}, \mathbf{y}) | \mathbf{x} \in C_k] = \int_{\mathbf{x} \in C_k} d(\mathbf{x}, \mathbf{y}) p(\mathbf{x}) d\mathbf{x} \tag{2.17}$$

where $\mathbf{y}_k$ is the centroid of the cell C_k and can be written as $\mathbf{y}_k = \text{Cent}(C_k)$. These two conditions are generalizations of the optimal scalar quantization solution presented earlier, to solve for the n-dimensional problem.

The analysis above is predicated on the assumption that the joint *pdf* $p(\mathbf{x})$ of the data vector is known. However, the *pdf* $p(\mathbf{x})$ is not known in practice. One method of codebook design is the *K-means algorithm* (attributed to Forgey), an iterative clustering technique used widely in pattern recognition [67]. Lloyd, in an unpublished paper in 1957, had developed the same algorithm but for the scalar quantization problem and a known distribution [66]. The application of this algorithm to a training sequence and the VQ case is termed as the generalized Lloyd algorithm (GLA) [34]. The algorithm is also known as the LBG algorithm based on the work of Linde, Buzo and Gray [65].

The algorithm divides the set of M training vectors $\mathbf{x}$ into L clusters, where $K = L$ in the design problem and $M \gg L$. q is the iteration index and $C_k(q)$ is the k-th cluster at iteration q. The GLA algorithm is as follows:

Step 1 *Initialization:* Set q = 0. Choose a set of output vectors $y_k(0)$, $1 \leq k \leq L$

Step 2 *Classification:* Classify the set of training vectors $\{\mathbf{x}(m), 1 \leq m \leq M\}$ into clusters C_k using the nearest neighbour rule

$$\mathbf{x} \in C_k(q), \text{ iff } d(\mathbf{x}, \mathbf{y}_k) \leq d(\mathbf{x}, \mathbf{y}_j) \quad \forall \quad k \neq j \tag{2.18}$$

Step 3 *Code Vector Updating:* set q to $q+1$. Update the code vector of every cluster by computing the centroid of the training vectors in each cluster.

$$\mathbf{y}_k(q) = \text{Cent } (C_k(q)) \quad 1 \leq k \leq L \tag{2.19}$$

For the mean-square error (MSE) distortion measure $\mathbf{y}_k(q)$ is given by

$$\mathbf{y}_k(q) = \frac{1}{M_k} \sum_{\mathbf{x} \in C_k} \mathbf{x}(m) \quad 1 \leq k \leq L \tag{2.20}$$

Also, compute the resulting distortion $D(q)$ at the q-th iteration.

Step 4 *Termination Test:* If the decrease in the overall distortion $D(q)$ at iteration q relative to $D(q-1)$ is below a certain threshold, stop; otherwise go to step 2. A reasonable choice of threshold is 0.001 [29].

The algorithm above has been shown to converge to a local optimum [65]. Note, however, that any such solution is not unique. Global optimization can be obtained by reinitializing the codevectors to all possible initializations and repeating the algorithm, and then selecting the codebook that results in a minimum overall distortion.

The choice of a distortion measure is an important consideration in the vector quantizer design problem. Some measures used, include the MSE (Euclidean distance between vectors) and the weighted MSE. In weighted MSE, unequal weights are introduced to render certain contributions to the distortion more important than others. A weighted measure, where $\mathbf{W}$ is a positive-definite weighting matrix, can be defined as

$$D(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \mathbf{W} (\mathbf{x} - \mathbf{y}) \tag{2.21}$$

2.2.3 Vector vs Scalar Quantization

Vector quantization is a process of redundancy removal that makes effective use of four inter-related properties of vector parameters in proper placement of the code vectors [67]. The four parameters are: linear dependency, nonlinear dependency, *pdf* shape and vector dimensionality. How a vector quantizer chooses code vector placements and cell shapes is the critical aspect in VQ design.

Redundancy in compression implies dependence amongst parameters. There are two types of dependence: linear and nonlinear. Linear dependency can also be considered as correlation. Two random variables that are correlated are linearly dependent. If they are uncorrelated then they are linearly independent even though they may be statistically dependent. The dependency that remains after the linear dependence is removed is called nonlinear dependence.

Consider random variables x_1 and x_2 that have a two-dimensional joint *pdf* as shown in Figure 2.1. It is trivial to show that x_1 and x_2 are dependent. If scalar quantizers are used to quantize x_1 and x_2 independently, the resulting scheme is equivalent to a vector quantization code that is the product of the scalar quantization codes for x_1 and x_2. Such VQ codes are known as *product codes* and from the example are clearly wasteful of bits as there are regions of zero probability that have been assigned some bits (unshaded region in Q). By a simple rotation of 45 degrees, the random variables can be transformed so as to be uncorrelated and therefore save bits in quantization.

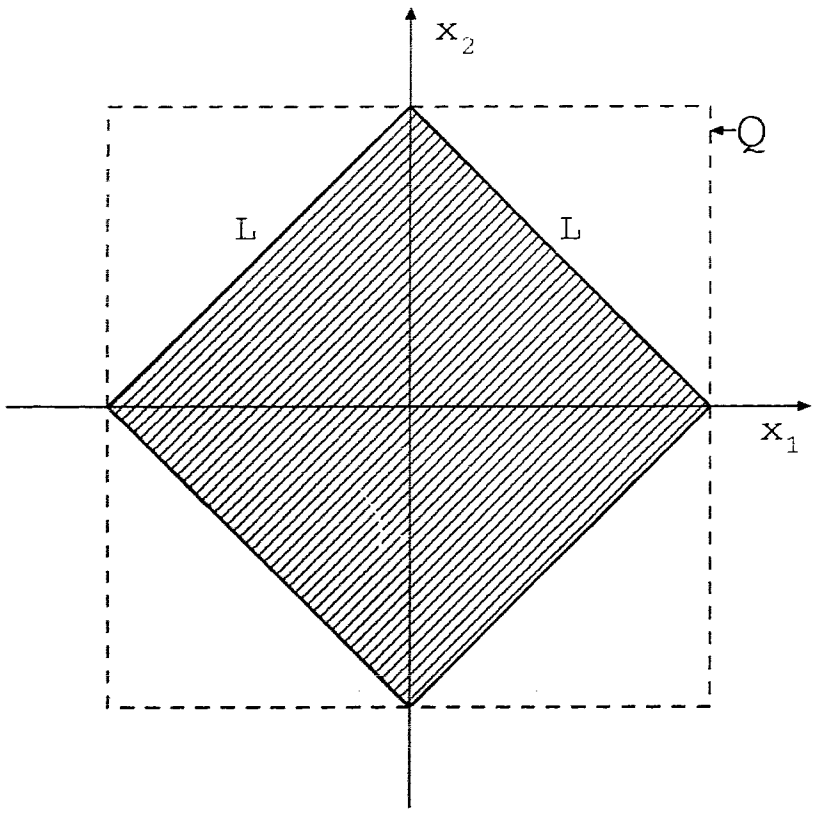


Figure 2.1: 2-d joint probability density function showing linear dependence

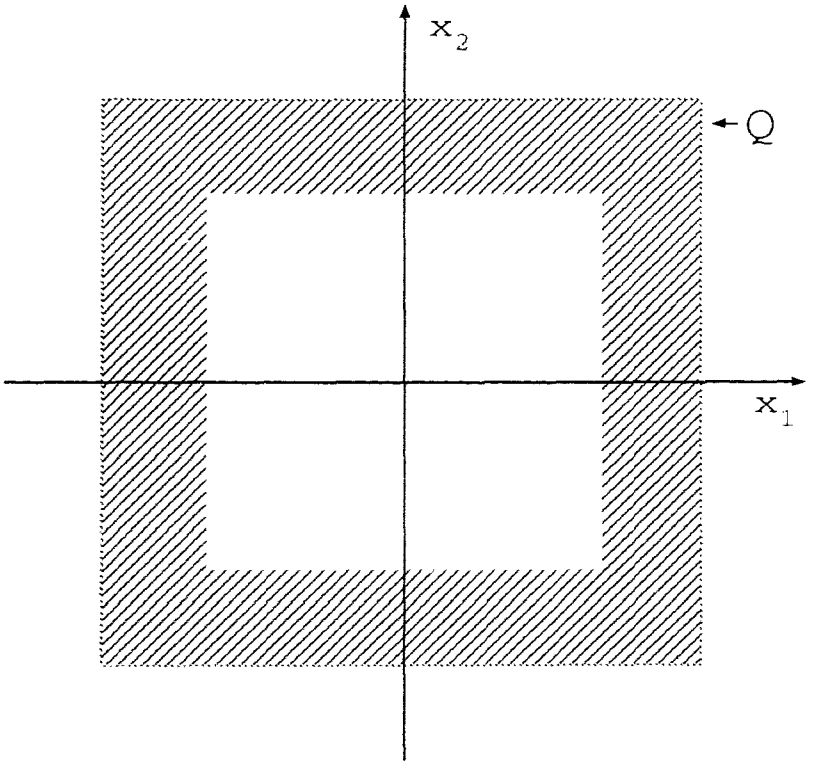


Figure 2.2: 2-d joint probability density function showing nonlinear dependence

In the example presented above, the random variables are also independent after rotation. If, however, the random variables were uncorrelated but statistically dependent, by modifying the joint *pdf* to that of Figure 2.2, savings in bits could be attained by ensuring that partitions excluded the area inside the smaller rectangle, where the probability was zero.

Dimensionality is another parameter that decides judicious choice of code placement. The vector quantizer examples shown in Figure's 2.1 and 2.2 assume as a square the shape of all their cells. One nice feature of vector quantizers for higher dimensions is the freedom to choose different cell shapes as opposed to being limited to the natural extension of the square in higher dimensions as required in scalar quantization.

In discussing the vector quantizer example, the cells are assumed to have the same shape and size in a quantization region. For non-uniform *pdf* shapes, one would expect that not only would there be unequal sizes of quantization regions to take full advantage of the unequal *pdf* distribution, but also that different cells shall have different shapes for optimal performance.

2.2.4 Sub-Optimal Vector Quantization

Having designed a codebook, one can then use it to quantize the input vector of dimension n by computing the distortion between the input vector and each of the code vectors, then choosing the code vector with the minimum distortion as the quantized value. This is a full search procedure since all code vectors are tested in obtaining the quantized vector. For an L-level quantizer, the number of distortion computations needed to quantize a single input vector is L. The computational cost for quantizing each input vector is proportional to nL. If the average number of bits per dimension is R, then $\log_2 L = Rn$, where $\log_2 L$ is the total number of bits used in quantizing the input vector. The computational cost is therefore proportional to $n2^{Rn}$, so complexity grows exponentially with vector dimension. Storage costs also grow exponentially with vector dimension.

Reducing the vector dimension often sacrifices the possibility of effectively exploiting the statistical dependency that exists in a set of samples. The solution to this complexity limitation is obtained through applying constraints in the structure of the vector quantizer so as to tradeoff complexity with quality judiciously. It is often possible to reduce the complexity by orders of magnitude while paying only a slight penalty in average distortion.

Some of the contrained VQs are:

- Lattice VQ: A lattice quantizer is a quantizer whose codewords form a subset of a lattice. Extensive papers on these forms of VQs developed by Gersho, Conway *et al.* are presented in a good reference source by Abut [1].

- Tree-Structured VQ: The codebook search is performed in stages. In each stage a substantial subset of candidate code vectors is eliminated from the search procedure. The procedure is attributed to Buzo *et al.* [6, 67].

- Product code VQ (Split VQ): Reduces the quantization procedure by splitting the codevector into subvectors, each to be quantized by a separate codebook. This technique is attributed to Sabin and Gray [88].

- Shape-Gain VQ: A product code technique that decomposes the problem into that of coding a scalar and a vector based on extracting the root mean-square

value of the vector components. This quantity is called the gain and acts as a normalizing factor. The normalized input vector is hence called the shape. Technique was first developed by Sabin and Gray [88].

- Multi-Stage VQ: The basic idea is to divide the quantization problem into successive stages, where the first stage performs a crude approximation of the input vector. Then, a second stage quantizer operates on the error difference between the original input vector and the first stage quantized output. Subsequent stages are used for refining the quantization procedure [51].

2.3 Linear Prediction

The most common use of prediction is to estimate a sample of a stationary random process from observations of several past samples. Linear prediction was first introduced in speech processing by Atal and Schroeder. This section presents a brief overview of linear prediction theory and the computation and quantization of linear prediction coefficients. For more detailed information, the reader is referred to the following references [69, 49, 28, 24, 93].

2.3.1 Linear Prediction for Stationary Signals

Consider a stationary random process $\{x(n)\}$ with zero mean and an autocorrelation function $r_k = E\{x(n)x(n-k)\}$. The linear prediction of the current sample $x(n)$, is obtained as a linear combination of past samples

$$\hat{x}(n) = -\sum_{k=1}^{m} a_k x(n-k) \tag{2.22}$$

where a_k are the linear prediction coefficients and m is the order of the linear predictor. The minus sign is included for convenience in expressing the error. The prediction error is defined as

$$e(n) = x(n) - \hat{x}(n) = x(n) + \sum_{k=1}^{m} a_k x(n-k) \tag{2.23}$$

The coefficients a_k are chosen to minimize the variance of the prediction error

$$\sigma_e^2 = E\{e(n)^2\} = E\{(x(n) - \hat{x}(n))^2\} \tag{2.24}$$

Taking the derivative of equation 2.24 with respect to a_k, the following condition for optimality is found

$$E\{e(n)x(n-k)\} = 0 \quad k = 1, 2, .., m \tag{2.25}$$

which states that a linear predictor is optimal (in the MSE sense) if and only if the resulting errors $e(n)$ are orthogonal to the observations $x(n)$. This is referred to as the *orthogonality principle*. By replacing $e(n)$ in equation 2.25 by 2.23 and 2.22, the following system of equations can be obtained

$$\sum_{j=1}^{m} a_j r_{|j-k|} = -r_k \quad k = 1, 2, .., m \tag{2.26}$$

This is a system of m linear equations with m unknowns a_j, which are called the Yule-Walker equations or Weiner-Hopf equations. Equation 2.26 can be written in vector form as

$$\mathbf{R}_{xx}\mathbf{a} = -\mathbf{r}_x \tag{2.27}$$

where $\mathbf{a} = [a_1, a_2, .., a_m]^T$ is the linear prediction coefficient vector, $\mathbf{r}_x = [r_1, r_2, .., r_m]^T$ and $\mathbf{R}_{xx}$ is the autocorrelation matrix given by

$$\mathbf{R}_{xx} = \begin{bmatrix} r_0 & r_1 & \cdots & r_{m-1} \\ r_1 & r_0 & \cdots & r_{m-2} \\ \cdots & \cdots & \cdots & \cdots \\ r_{m-1} & r_{m-2} & \cdots & r_0 \end{bmatrix} \tag{2.28}$$

Assuming $\mathbf{R}_{xx}$ is positive definite and therefore nonsingular, the following solution can be obtained for the optimal linear prediction coefficients

$$\mathbf{a} = -\mathbf{R}_{xx}^{-1}\mathbf{r}_x \tag{2.29}$$

$\mathbf{R}_{xx}$ is Toeplitz and symmetrical, hence computationally efficient techniques can be used for performing the matrix inversion such as the Levinson-Durbin algorithm [69, 24].

The linear prediction model can be obtained by considering equation 2.23 as a filtering operation where the filter system function is $A(z)$ with input $x(n)$ and output $e(n)$. From equation 2.23 it follows immediately that

$$A(z) = \sum_{k=0}^{m} a_k z^{-k} \tag{2.30}$$

$A(z)$ is known as the prediction filter. It can also be referred to as the whitening filter, based on the property that an infinite order optimal linear predictor can transform a stationary signal into a white noise process. The proof is as follow: From the orthogonality condition, $e(n)$ is uncorrelated with all past values $x(n-j)$ for $j \geq 1$ of the input sequence $x(n)$. But $e(n-i)$ is fully determined by a linear combination of past input values $x(n-i-j)$ for $j \geq 0$. Hence, $e(n)$ is uncorrelated with $e(n-i)$ for $i > 0$. Since $e(n)$ is stationary, it is therefore white.

The filter $1/A(z)$ is known as the synthesis filter, because given the residual error signal $e(n)$, it can reconstruct the original signal $x(n)$. The filter $A(z)$ is commonly referred to as the inverse filter in speech coding systems.

Pitch Prediction

The discussion has so far been limited to short-term prediction. Pitch prediction (long-term prediction) plays a crucial role in predictive systems for voice by attempting to predict the periodicity in voiced speech. In pitch predictors, the current sample is predicted from a combination of past samples, k_p samples in the past, where k_p is the pitch period. Generally, any lag value that has significant correlation is appropriate.

The pitch predictor equation is given by

$$\hat{x}(n) = \sum_{k=-(m-1)/2}^{(m-1)/2} b_k x(n - k_p - k) \tag{2.31}$$

where m relates to the pitch predictor order, k_p is the pitch period, and b_k are the predictor coefficients. A one-tap predictor corresponds to $m = 1$, while a three-tap predictor corresponds to $m = 3$. A three-tap predictor proves to be desirable, as the use of a one-tap predictor results in a loss in performance due to a lower prediction gain. Also, since the pitch period may not be an exact integer number of samples, by making use of multiple taps, non-integer pitch values can be allowed for by adjusting the tap gains to ensure the autocorrelation peaks lie at these non-integer points. Fractional pitch filters are an alternative way of achieving the same effect (Kroon *et al.* [59]), while overcoming the bit constraints imposed by higher order predictors. Studies undertaken by Veeneman showed that multi-tap predictors outperformed fractional pitch filters [98].

The transfer function of the pitch prediction filter is given by

$$1 - B(z) = 1 - \sum_{k=-(m-1)/2}^{(m-1)/2} b_k z^{-(k_p+k)} \tag{2.32}$$

2.3.2 Adaptive Prediction: Block vs Recursive

In reality, speech is not a stationary process, so using a fixed predictor based on long time estimates for the autocorrelation function results in significant degradation. A substantial improvement in performance can be obtained by using an adaptive linear prediction coefficient set according to the time-varying speech statistics.

The adaptation can be classified into two different categories. First, the predictor coefficients can be computed using the original signal (forward adaptation) or can be derived from the past reconstructed signal (backward adaptation). Second, the adaptation can be done on a block-by-block basis (block adaptation) or on a sample-by-sample basis (recursive adaptation).

In speech coders which utilize forward adaptation, block adaptation is preferred, which results in buffering delays. Since the linear prediction coefficients need to be sent to the receiver as side information, block adaptation reduces the coding rate of the parameters. In backward adaptive systems, as the filter coefficients need not be transmitted, they can be updated more frequently, lending themselves to recursive adaptation. However, the performance of the linear predictor is degraded somewhat as a result of the adaptation on a signal that contains quantization noise. Both recursive and block adaptation are used in backward adaptive systems. More details are presented in Subsections 2.3.3 and 2.3.4.

To enable a better understanding of the signals that can possibly be used in adaptation, block diagrams of the input-output relationship for a typical encoder and decoder configuration are presented in Figures 2.3 and 2.4 respectively.

2.3.3 Block Adaptation

Suppose we give up now on the stationarity assumption. This leads to two alternative formulations of linear predictor design: the *autocorrelation method* and the *covariance method.*

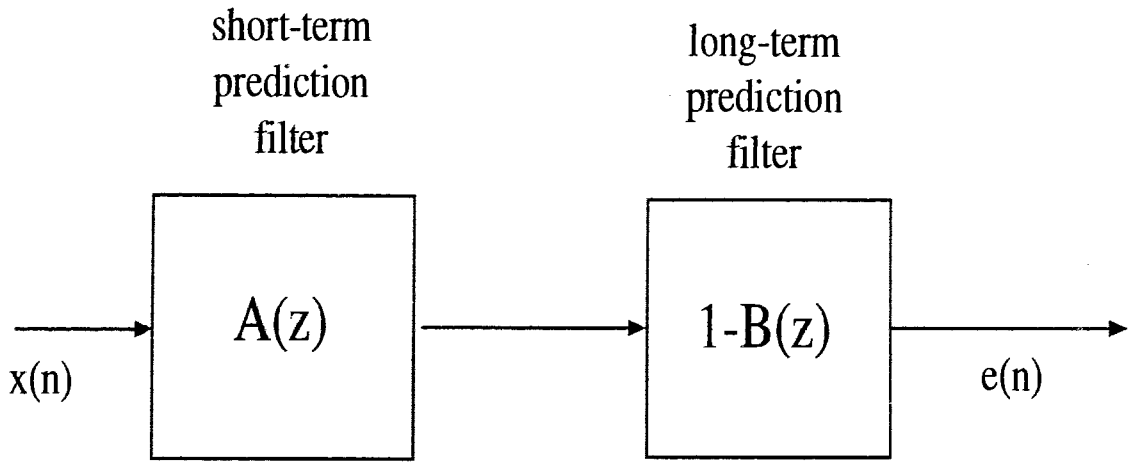


Figure 2.3: Encoder configuration of short- and long-term predictor

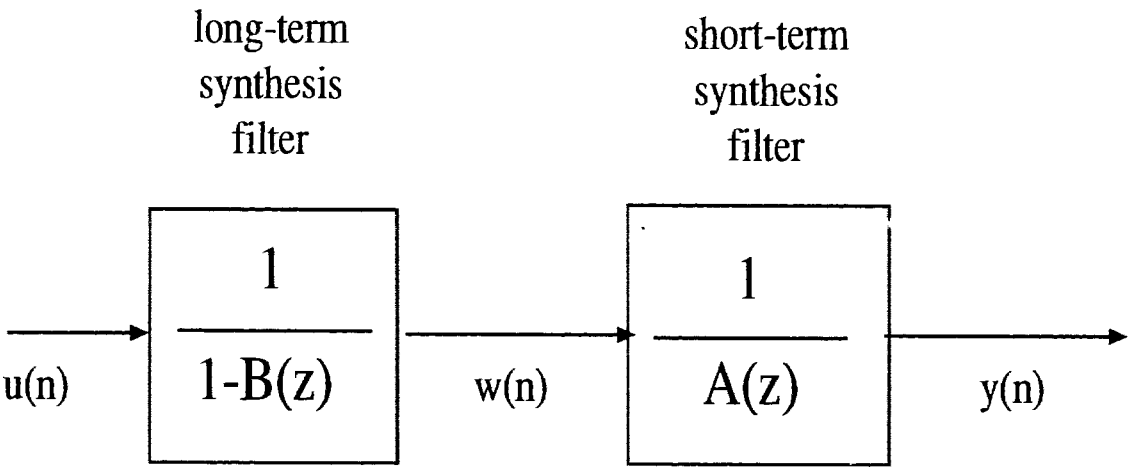


Figure 2.4: Decoder configuration of short- and long-term predictor

If the analysis presented in Section 2.3.1 was performed with a window function introduced to differentiate between the part of the original process $x(n)$ that is observable and that which is not, $s(n)$ will be a truncated version of the original random process, with $s(n) = win(n)x(n)$ and $n \in \mathcal{M} \equiv \{0, 1, .., M\}$. $win(n)$ is nonzero in a finite interval $\mathcal{M}$ of observable events. By minimizing the prediction error in a yet unspecified finite interval $\mathcal{N}$, equation 2.26 can be written as

$$\sum_{j=1}^{m} a_j \phi_{j,k} = -\phi_{0,k} \quad k = 1, 2, .., m \tag{2.33}$$

where

$$\phi_{j,k} = \sum_{n \in \mathcal{N}} s(n-j)s(n-k) \quad 1 \leq j, k \leq m \tag{2.34}$$

In the autocorrelation method, the error $e(n)$ though minimized over a finite interval $\mathcal{N}$ given by the set of all integers, is equivalent to minimizing over the interval $[0, M+m]$ as $e(n)$ is zero outside this region because $s(n)$ is zero outside the region $n \in \mathcal{M}$. Equation 2.34 can now be written as

$$\phi_{j,k} = \sum_{n=0}^{M-|j-k|} s(n)s(n+|j-k|) = r_{|j-k|} \quad 1 \leq j, k \leq m \tag{2.35}$$

Hence, equation 2.33 can be rewritten to be nothing but equation 2.26, with the notation 'x' being replaced by 's' to signify the windowed signal in the autocorrelation computation.

The autocorrelation method lends itself towards an efficient way of solving for the linear prediction coefficients using the Levinson-Durbin recursion, which is presented in Markel and Gray [69]. Also, it always results in a stable inverse filter, which is necessary in analysis-by-synthesis coders. However, the prediction error is going to be very large in predicting the first few and last few samples of the process $s(n)$, *i.e.* for $n = \{[0, m-1] \text{ and } [M+1, M+m]\}$. A tapered window will help reduce the effect of these meaningless components on the quantity to be minimized. Of course, for $m \ll M$, the end effects will be relatively small and meaningful results will still be achieved. The autocorrelation method has a slight drop in performance over the covariance approach.

The covariance method on the other hand performs the error minimization over an interval $n \in \mathcal{N} \equiv [m, M]$, this way minimization is not performed over meaningless components and all $M - m + 1$ samples used in the covariance calculation are

observable. Equation 2.34 can be written as

$$\phi_{j,k} = \sum_{n=m}^{M} s(n-j)s(n-k) \quad 1 \leq j,k \leq m \tag{2.36}$$

with the system of equations governing the solution, given by equation 2.33. This can be written in a matrix form as

$$\mathbf{\Phi a} = -\phi \tag{2.37}$$

In the covariance method the matrix $\mathbf{\Phi}$ is not Toeplitz, so the Levinson-Durbin recursion can not be used in matrix inversion, resulting in computational complexity. This complexity can be significantly reduced using the Choleskey decomposition [69]. However, computationally the Choleskey decomposition is still considerably less efficient than the Levinson-Durbin recursion. For a 10-th order short-term filter, the Choleskey decomposition is approximately three times as computationally complex [24]. Another drawback of the covariance method is the need to stabilize the inverse filter, as it is not guaranteed to result in a stable filter. However, the covariance method may achieve slightly better performance over the autocorrelation method [69].

Pitch Prediction

Assuming knowledge of the pitch period, the predictor coefficients can be computed by solving the Weiner-Hopf equations as given in equation 2.27. For a three-tap predictor, the predictor delays become $k_p - 1, k_p, k_p + 1$, and equation 2.27 can be written as

$$\begin{bmatrix} r_0 & r_1 & r_2 \\ r_1 & r_0 & r_1 \\ r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} b_{-1} \\ b_0 \\ b_{+1} \end{bmatrix} = \begin{bmatrix} r_{k_p-1} \\ r_{k_p} \\ r_{k_p+1} \end{bmatrix} \tag{2.38}$$

In the pitch predictor, neither the autocorrelation or covariance method guarantees stability of the resulting pitch synthesis filter. Stability constraints need to be derived and checked to ensure the stability of the filter. For a one-tap predictor, the constraint is simply $|b_0| < 1$. For a three-tap predictor, these constraints are derived in [85].

In block adaptation of the pitch predictor in backward adaptive systems, the prediction parameters are computed from a previous block of reconstructed speech, under

the assumption that the parameters are varying slowly enough that the parameters obtained from the previous reconstructed signal are close to those for the current block, thus suggesting smaller sized blocks for better performance. However, the analysis frame has to be large enough to be able to compute the predictor coefficients accurately. These conflicting requirements results in a complicated system (overlapped frame structure) to solve for the frame size tradeoff.

2.3.4 Recursive Adaptation

An alternative to block adaptation is recursive adaptation, which is used in backward adaptive systems. The recursive adaptation is of lower complexity, enabling the adaptation to be done on a sample by sample basis.

Recursive algorithms attempt to minimize the mean squared prediction error $\epsilon^2 = E[e^2(n)]$ as a function of the predictor coefficients $\mathbf{a}$. This is done using the *gradient algorithm* as a starting point for solving the prediction error minimization problem [37].

Let the j-th iteration of the coefficient vector be $\mathbf{a}_j$. The gradient of the error $\nabla E[e^2(n)]$ is a vector in the direction of maximum increase of the error. Thus, by subtracting off a term that is the proportional to the gradient, the resulting vector should be closer to the optimal predictor coefficient. Denoting by μ the adaptation stepsize, the gradient algorithm is written as

$$\mathbf{a}_{j+1} = \mathbf{a}_j - \frac{\mu}{2}\nabla E[e^2(n)] \tag{2.39}$$

Taking the partial derivative, equation 2.39 can be written as

$$\mathbf{a}_{j+1} = \mathbf{a}_j + \mu E[e(n)\mathbf{x}(n)] \tag{2.40}$$

The gradient algorithm will converge to the optimal set of coefficients provided the stepsize μ satisfies the constraint

$$0 < \mu < \frac{2}{\lambda_{max}} \tag{2.41}$$

where λ_{max} is the maximum eigenvalue of the autocorrelation matrix $\mathbf{R}_{xx}$.

The *stochastic gradient* algorithm or the *least mean squares* (LMS) algorithm overcomes the problem of computing the expectation, which requires knowledge of the

ensemble statistics. The problem is circumvented by replacing the ensemble average by a time average. Equation 2.40, ignoring the expectation term, can be written as [37]

$$\mathbf{a}_{j+1} = \mathbf{a}_j + \mu e(n)\mathbf{x}(n) \tag{2.42}$$

where $\mathbf{x}(n) = [x(n-m), x(n-m+1), .., x(n-1)]^T$ is the original speech vector.

Another issue is that the signals $x(n)$ and $e(n)$, which are the original speech and error signal respectively, are not available at the receiver. To circumvent this problem, they are replaced by the signals $y(n)$ and $u(n)$, which are the reconstructed speech and quantized error signal respectively. The signal replacement is not a problem as long as the quantization is relatively accurate. The adaptation equation now becomes

$$\mathbf{a}_{j+1} = \mathbf{a}_j + \mu u(n)\mathbf{y}(n) \tag{2.43}$$

This can be written in scalar form as

$$a_k^{j+1} = a_k^j + \mu u(n)y(n-k) \tag{2.44}$$

where k corresponds to the filter coefficient index.

Pitch Prediction

Long-term predictors can also be adapted in a recursive fashion [82, 16]. The output of the long-term predictor is given by

$$w(n) = u(n) + \sum_{k=-(m-1)/2}^{(m-1)/2} b_k w(n-k_p-k) \tag{2.45}$$

By making use of the LMS algorithm for recursive adaptation of the long-term predictor, an adaptation equation can be obtained to be as follows

$$b_k^{j+1} = b_k^j + \mu u(n)w(n-k_p-k) \tag{2.46}$$

The analysis is done for a backward adaptive system, hence the use of the decoder signal $w(n)$ instead of a corresponding signal at the encoder.

As a result of constraints in stability checks for both long- and short-term predictors in the transversal filter form, an alternative (equivalent) filter known as the *lattice filter* offers significant advantages in implementation. The lattice structure is introduced in the next subsection and an adaptation algorithm for the lattice filter is also presented.

2.3.5 Lattice Filters

Linear prediction was seen to be an effective estimation procedure in modeling the accoustical tube of the vocal tract [69]. The reflection coefficients which uniquely define the area ratios of the accoustic tube model of the vocal tract can be obtained directly from linear prediction analysis of the speech waveform. The reflection coefficients k_m are also called *partial correlation* or *PARCOR* coefficients. For a detailed insight into the accoustic tube and speech production models, the reader is referred to Markel and Gray [69]. A detailed presentation of lattice filters is given in [69, 37, 24].

The prediction error equations giving the j-th order forward and backward prediction errors, $e_j(n)$ and $r_j(n)$ respectively, are

$$e_j(n) = x(n) + \sum_{k=1}^{j} a_k(j)x(n-k) \tag{2.47}$$

$$r_j(n) = x(n-j) + \sum_{k=1}^{j} a_{j-k+1}(j)x(n-k+1) \tag{2.48}$$

Due to the symmetry of the autocorrelation function (*i.e.* $r_k = r_{-k}$), the optimal backward predictor coefficients are the mirror image of the optimal forward predictor coefficients [37]. The different orders of the prediction errors are mutually orthogonal. The proof of which makes use of the projection theorem and is presented in [37].

The *order update equations* that relate the higher order prediction errors to lower order prediction errors are given by

$$e_{j+1}(n) = e_j(n) - k_{j+1}^{(n)} r_j(n-1) \tag{2.49}$$

$$r_{j+1}(n) = r_j(n-1) - k_{j+1}^{(n)} e_j(n) \tag{2.50}$$

where, $k_j^{(n)}$ is the j-th order lattice filter coefficient at time n. Equations 2.49 and 2.50 define the structure of a lattice filter. $k_j^{(n)}$ can also be known as the partial correlation coefficient, which is given by

$$k_j = \frac{E[e_j(n)r_j(n)]}{E[e_j(n)^2]E[r_j(n)^2]} \tag{2.51}$$

This equation means that the partial correlation coefficient is the correlation between the forward and backward prediction errors. The proof of equations 2.49 to 2.51 is presented in [37].

Equations 2.49 and 2.50 are the recursion equations for the all-zero lattice filter. From equations 2.49 and 2.50 it follows that

$$\begin{aligned} A_{j+1}(z) &= A_j(z) - k_{j+1}z^{-1}B_j(z) \\ B_{j+1}(z) &= z^{-1}B_j(z) - k_{j+1}A_j(z) \end{aligned} \tag{2.52}$$

where $A_j(z)$ and $B_j(z)$ are the transfer functions of the lattice filter from the input to the j-th order forward and backward prediction errors respectively. $B_j(z)$ here should not be confused with the notation to denote a pitch predictor used elsewhere in the thesis. It is possible to implement an all-pole filter as a lattice filter by reversing the signal propagation direction of $e_j(n)$, as illustrated in Figure 2.6. The all-zero lattice filter and the corresponding all-pole lattice filters are shown in Figures 2.5 and 2.6.

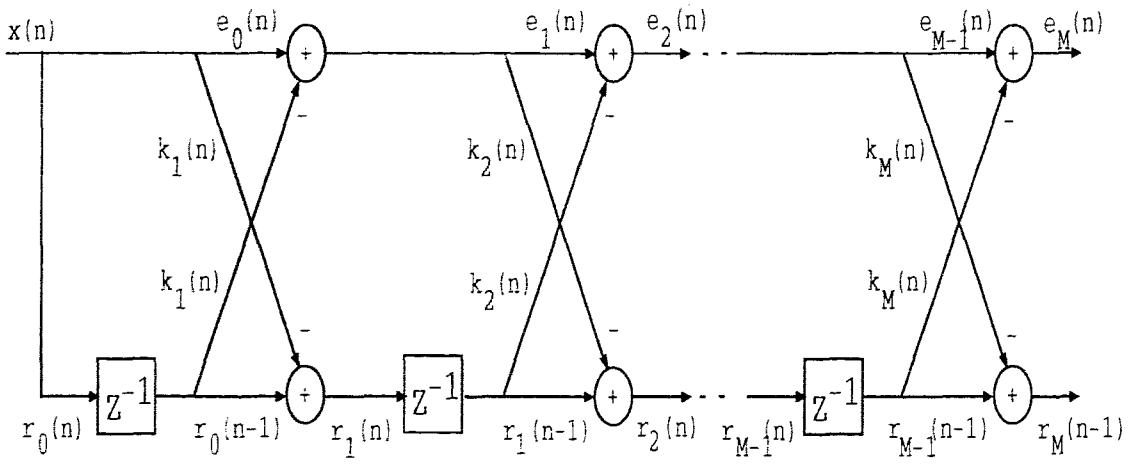


Figure 2.5: All-zero lattice filter representation

Conversion between Lattice and Transversal forms

Given a transversal representation, one needs to be able to obtain a lattice equivalent. This is a by-product of the Levinson-Durbin recursion, which computes both transversal and lattice filter coefficients given the autocorrelation coefficients. Let k_j be the j-th reflection coefficient and $a_j(m)$ be the j-th transversal coefficient of the m-th order filter. Given the set of transversal filter coefficients the lattice coefficients are now obtained as follows, the proof of which is presented in [69, 37]:

Start with the m-th order predictor and work backwards to obtain the corresponding lower order predictors.

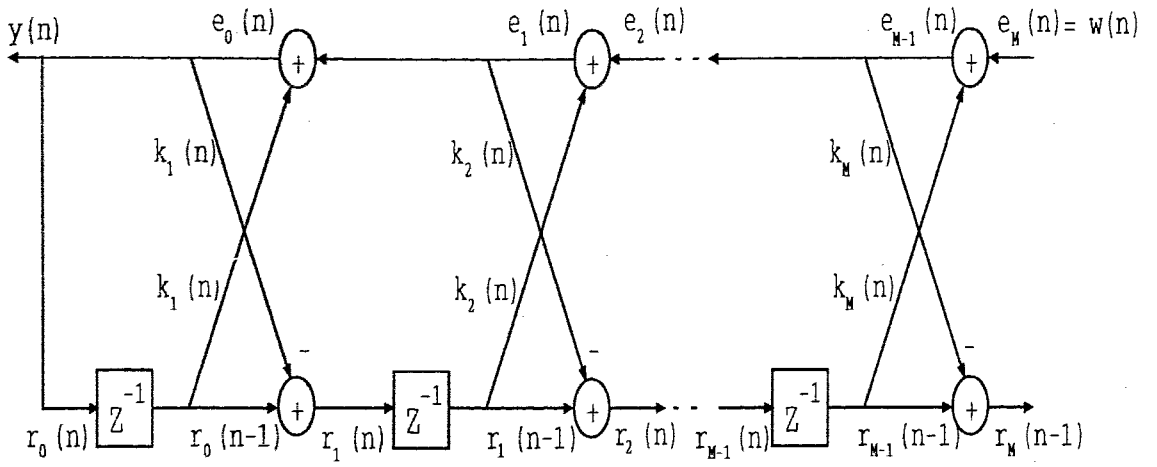


Figure 2.6: All-pole lattice filter representation

1. Set $j = m$, then compute

$$k_j = a_j(j) \tag{2.53}$$

$$a_k(j-1) = \frac{k_j a_{j-k}(j) + a_k(j)}{1 - k_j^2} \text{ for } k = 1, 2, .., j-1 \tag{2.54}$$

2. decrement j to $j-1$ and repeat procedure 1. till $j = 1$.

The lattice filter coefficients are independent of the order of the filter as a result of the orthogonality of the prediction errors at each stage in the lattice filter. The orthogonalization of the lattice filter stages speeds up the adaptation of subsequent stages. Finally, the all-pole lattice filter is stable provided that the reflection coefficients $|k_j| \leq 1$ for $j = 1, 2, .., m$ where m is the filter order.

Lattice Adaptation

Lattice filter adaptation can be done in either block or recursive fashion. In block adaptation, the lattice coefficients are simply obtained as a by-product of the Levinson-Durbin algorithm for solving the Weiner-Hopf equations, resulting in a computationally efficient implementation.

The lattice filters have been found to converge more rapidly then their corresponding transversal realization [37]. Adaptation algorithms for lattice filters are generally more complex, so the discussion is limited to the simpler gradient based lattice adaptation algorithms. Recursive adaptation of the lattice filter is presented in Reininger

and Gibson [87].

The reflection coefficients are updated according to the LMS algorithm as [87]

$$k_j(n+1) = \frac{\alpha_j(n+1)}{\beta_j(n+1)} \quad (2.55)$$

$$\alpha_j(n+1) = \upsilon\alpha_j(n) + \lambda r_j(n-1)e_j(n) \quad (2.56)$$

$$\beta_j(n+1) = \upsilon\beta_j(n) + e_j^2(n) \quad (2.57)$$

where the term λ is a damping factor introduced to improve performance in noisy channel conditions and υ is the exponential fading memory term.

Quantization

Quantization of the short-term predictor coefficients is a critical aspect of speech coding systems. The transversal coefficients have a very wide dynamic range and do not guarantee synthesis filter stability, making them unsuitable for quantization. This in turn leads to the conversion to lattice coefficients. These coefficients have a much smaller dynamic range and have been suitably quantized using both scalar and vector quantization procedures. An alternative way to represent short-term predictor coefficients is by using the line spectral pairs (LSP) approach. A good reference source for early work in short-term predictor quantization is [24].

2.3.6 Line Spectral Pairs

LSPs have become the most widely studied procedure for representing and quantizing the linear prediction coefficients. A good reference source on LSP quantization is the paper by Gersho [27] which cites most recent research efforts in spectral quantization. The conversion of linear prediction coefficients (transversal form) to line spectral frequencies (LSF) or LSPs is presented using the Chebyshev polynomials in [52]. A good description of the LSP model is presented in Furui [24].

Given the inverse prediction filter $A(z)$, a symmetrical polynomial $A_p(z)$ and an anti-symmetrical polynomial $A_q(z)$ can be obtained as

$$\begin{aligned} A_p(z) &= A(z) + z^{-(m+1)}A(z^{-1}) \\ A_q(z) &= A(z) - z^{-(m+1)}A(z^{-1}) \end{aligned} \quad (2.58)$$

Using equation 2.52, the polynomials $A_p(z)$ and $A_q(z)$ can be considered as being obtained from A(z) by adding an additional stage with reflection coefficients equal to -1 and $+1$ respectively [24]. The roots of these polynomials determine the LSPs. These polynomials have roots at $z = 1$ and/or $z = -1$, which can be removed by polynomial division to give the polynomials $G_1(z)$ and $G_2(z)$ which are symmetric polynomials of even order. These polynomials have roots that are complex-conjugate pairs. The LSFs are the angular positions of these roots.

$$\begin{aligned} G_1(z) &= \frac{A_p(z)}{1+z^{-1}} \quad G_2(z) = \frac{A_q(z)}{1-z^{-1}} \quad \text{even } m \\ G_1(z) &= A_p(z) \quad G_2(z) = \frac{A_q(z)}{1-z^{-2}} \quad \text{odd } m \end{aligned} \tag{2.59}$$

The lowest frequency LSF corresponds to a root of $G_1(z)$. An interesting property of the LSPs is the interlacing of the roots. The successive roots of $G_1(z)$ and $G_2(z)$ are interlaced between each other. This is known as the *alternating roots property* and allows for simple stability control.

LSPs have good quantization properties as well as a strong link to formant frequencies of the accoustic tube model in speech production. They also lend themselves well to smoother interpolation as a result of the interlacing of the roots.

2.4 Quality Measures

A critical aspect of speech coding systems is the need for a fidelity criterion that adequately reflects the subjective speech quality of the reconstructed speech in comparison to the original speech signal. A useful reference on objective quality assessment is [60].

Distortion or "distance" parameters directly measure the amount and type of distortion between the input and output speech signals. The distortion is calculated as a difference or ratio of input and output speech parameters. Traditionally, distortion measures have been the focus of most research regarding objective parameters for evaluating speech quality.

2.4.1 Signal-to-Noise Ratio

Signal-to-noise ratio (SNR) was one of the first measures examined for use as an objective parameter for measuring voice quality. Denoting $x(n)$ as the input speech signal and $y(n)$ as the reconstructed speech signal, the SNR is given by

$$SNR = 10 \log_{10} \frac{\sum_n x(n)^2}{\sum_n [y(n) - x(n)]^2} \tag{2.60}$$

A better assessment of subjective speech quality can be obtained by using the segmental SNR (SegSNR). This criterion compensates for the low weight given to the low-level signal performance in the SNR evaluation. SegSNR is calculated by computing the SNR over a block of, say, 256 samples, eliminating silence intervals by excluding frames whose average signal power is below a certain threshold with respect to the average power level of the entire speech file. The SegSNR is obtained as an arithmetic average of block SNR values over the entire speech file.

Though the SegSNR is a very common measurement of system performance, it has a few drawbacks such as sensitivity to delay estimation error and phase distortion. If the delay error is of the order of one half of the dominant period, lower than expected SNR estimates could result even though the actual SNR is quite high. This is because the output and input signals are nearly 180 degrees out of phase [60].

2.4.2 Mean Opinion Score

An alternative approach to evaluating quality is to use a subjective criterion such as the mean opinion score (MOS), which is the most used subjective measure today.

The MOS is obtained by averaging the scores given by a panel of untrained listeners. Untrained listeners are used to avoid any inherent bias introduced in scoring by "expert listeners". given Each listener characterizes the speech signal by a score on a scale of 1 (for poor quality) to 5 (for excellent quality). Typically, the average is done over 30-60 listeners. Though MOS scores may vary from test to test for the same system, they are reproducible when compared to a common reference. The speech quality required in commercial telephony is called toll quality. Toll quality is characterized as a MOS score better than 4.0. Pulse code modulation (PCM) standard (G.711) achieves a MOS of about 4.2 and the adaptive differential pulse code modulation (ADPCM) standard at 32 kb/s (G.721) achieves a MOS of 4.0. The LD-CELP G.728 standard also achieves

a MOS score of 4.0. A MOS score above 3.5 corresponds to "communications speech quality" [14].

The MOS tests performed in this research use 4 to 8 sentences for scoring purposes and typically 20 listeners, which results in 80 to 160 samples points in the MOS averaging computation. The mean and variance of the samples points are computed. The variance is used to compute 95% confidence intervals around the mean value of the MOS. The average MOS scored by each listener is checked to see if it falls within the 95% confidence interval around the mean. The MOS averaging process is satisfactory if at least 95% of the individual listener averaged MOS scores are within the 95% confidence interval.

2.5 Speech Coding Systems

Speech coding systems have been traditionally classified into waveform coders and vocoders. In waveform coders, the reconstructed signal attempts to match the waveform of the original signal. Vocoders by contrast, do not attempt to approximate the waveform shape but rather extract parameter information that characterize the speech signal and transmit these parameters to be used at the decoder in reconstructing a similarly sounding waveform. Vocoders operate at much lower rates than waveform coders but suffer from a perceptual drop in performance. However, in the last 10-15 years this classification is no longer appropriate as there has been significant research effort in hybrid types of coders. These coders are more popularly known as *analysis-by-synthesis* (A-by-S) coders.

A detailed history of waveform coders are presented in various references [49, 14]. Gersho provides an excellent review of the state of the art in speech coding, citing extensive research efforts in all aspects of speech coding [27].

2.5.1 Waveform Coders

The earliest developed coding system was based on PCM and still remains widely used. It has the advantage of low complexity, zero buffering delay and toll quality speech in the strictest telecommunication sense. The concept of toll quality was discussed in Section 2.4. PCM was the basis of the earliest standard for speech coding at 64 kb/s

adopted by the ITU, this standard is more commonly referred to as G.711, and uses 8-bit log-PCM.

In PCM coders, the speech signal is sampled and then quantized directly. In order to reduce the rate while preserving the quality, differential coders were introduced whereby the energy of the signal before quantization is reduced by making use of a predictive model to remove redundancy. This prediction residual is then quantized. This idea resulted in the 32 kb/s ADPCM standard, known as G.721. ADPCM still proves to be very useful due to its low computational complexity. Differential coders which utilize both short- and long-term predictors, i.e ones which attempt to remove both the short- and long-term redundancies that exist in speech are referred to as adaptive predictive coders (APC).

Interest in waveform coders has given way to focus in the area of hybrid coders and vocoders as speech researchers attempt to attain high quality speech under the constraints of much lower bit rates.

2.5.2 Vocoders

Speech quality obtained by waveform coding methods is found to degrade rapidly as the bit-rate drops below 4 kb/s. This is as a result of very few bits available to adequately approximate the original waveform in both classical waveform coders or A-by-S coders. Vocoders have a greater potential to reproduce a signal at lower rates in a perceptual sense, as they do not attempt to reproduce the waveform shape.

A commonly used vocoder model was the linear prediction coding (LPC) vocoder as shown in Figure 2.7. A similar scheme resulted in a U.S government standard more widely known as LPC-10. The encoder computes and quantizes the optimal linear prediction coefficients, a gain factor and the pitch value for each speech frame. The decoder synthesizes speech by passing an excitation signal through the LPC synthesis filter. The decoder generates a random noise excitation for unvoiced frames and a train of impulses, pitch period apart, for voiced frames [69]. The reproduced speech quality has a "buzzy" sound to it and is of poor quality.

Recently though, several effective techniques have been developed which enable vocoder based systems to achieve performance close to A-by-S coders at rates of about 4 kb/s and improved performance over existing A-by-S coders at 2.4 kb/s

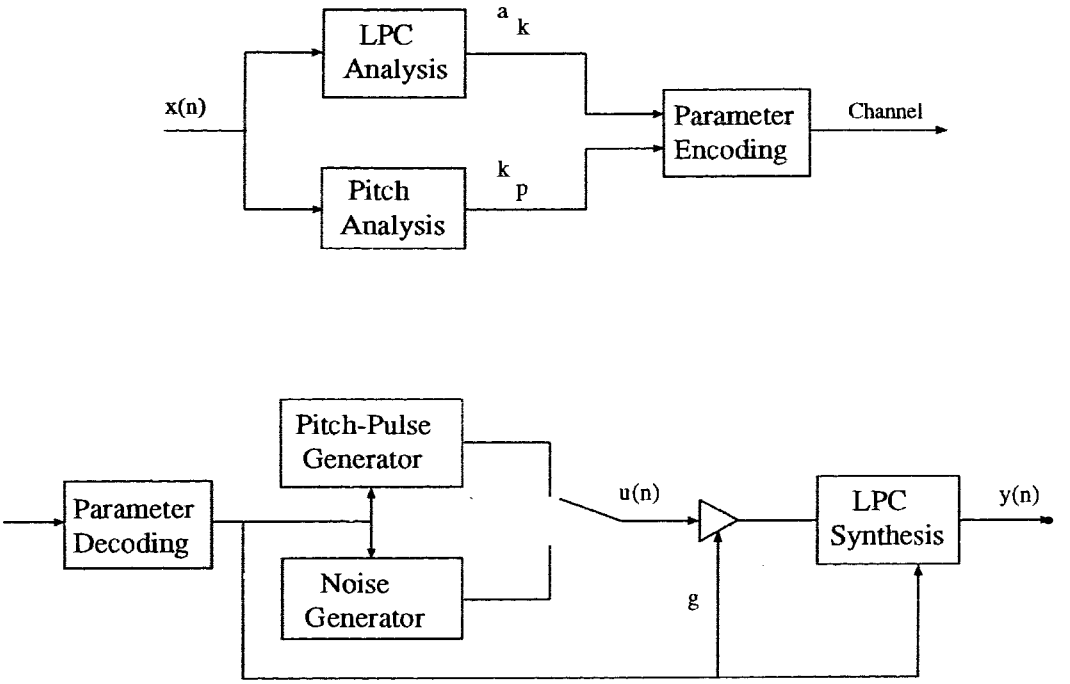


Figure 2.7: Linear prediction coding vocoder

and below. An important class of vocoders is the sinusoidal coder. These coders use a sum of sinusoids model for voiced speech generation. Some of the main types of sinusoidal coders are: the sinusoidal transform coder (STC) [72], the multi-band excitation (MBE) coder [35], and the time-frequency interpolation (TFI) coder [91]. Also, prototype waveform interpolation (PWI) [56] provides a merging of the ideas of CELP (Section 2.6) and sinusoidal coding.

2.5.3 Analysis-by-Synthesis

The analysis-by-synthesis (A-by-S) approach to speech coding has seen the most interest recently. The idea of A-by-S comes about from the merging of some features of vocoders into a coder based on the waveform coding model. In A-by-S coders, a set of parameters that characterize the signal are obtained and adapted in such a way as to obtain the best match by comparing the original signal with a reconstructed approximation at the encoder. The A-by-S approach is a closed-loop procedure, as the choice of optimal parameter set is governed by a comparison of the original and reconstructed signal at the encoder. In the vocoder approach, on the other hand, parameters are estimated in open-loop.

Figure 2.8 shows a configuration of a general A-by-S speech coder. The excitation signal $u(n)$, is obtained by a table lookup from an excitation codebook. The spectral codebook contains set of possible parameters for the synthesis filter. The excitation signal $u(n)$, is gain scaled by g and passed through the synthesis filter to generate the synthesized speech. All possible indices of both codebooks are tried and the resulting signal is compared with the original signal. The error signal $e(n)$ is weighted perceptually by a weighting filter W, and the indices which achieve the best match are selected for transmission.

An exhaustive search of the excitation and spectral codebooks proves to be prohibitive due to the excessive computational complexity associated with it. To reduce the complexity, typical A-by-S systems use forward or backward adaptation to find the best set of synthesis filter parameters using the approaches discussed in Section 2.3.3 and 2.3.4. This is an open-loop procedure designed to avoid searching a spectral codebook, with the spectral codebook used for open-loop quantization of the synthesis parameters. The synthesis filter normally consists of both a long-term and a short-term filter, which attempt to model the short- and long-term periodicity in speech. Figure 2.9 gives the block diagram of the transformed A-by-S coder.

The first effective form of A-by-S coder was the multi-pulse LPC (MP-LPC) coder, where each frame of excitation is computed as a combination of pulses whose position and amplitudes are optimized in close-loop. CELP, however, was based on using vector quantization in the A-by-S excitation model. The rest of the thesis will now solely be based on the CELP speech coding model (Section 2.6).

Perceptual Weighting

The use of perceptual weighting of the error signal in the excitation codebook search is a key aspect of A-by-S coders. The spectral envelope of the speech signal is characterized by several peaks called formants. The human ear is more sensitive to noise frequency components that lie in the valleys between these peaks, than to components in close proximity to the peaks. The goal of perceptual weighting is to attenuate the frequency components of the quantization noise in these valleys, thus giving better perceptual quality. The error is passed through a time-varying filter, which has a shape that is normally a scaled approximation of the synthesis filter, thus it tends to

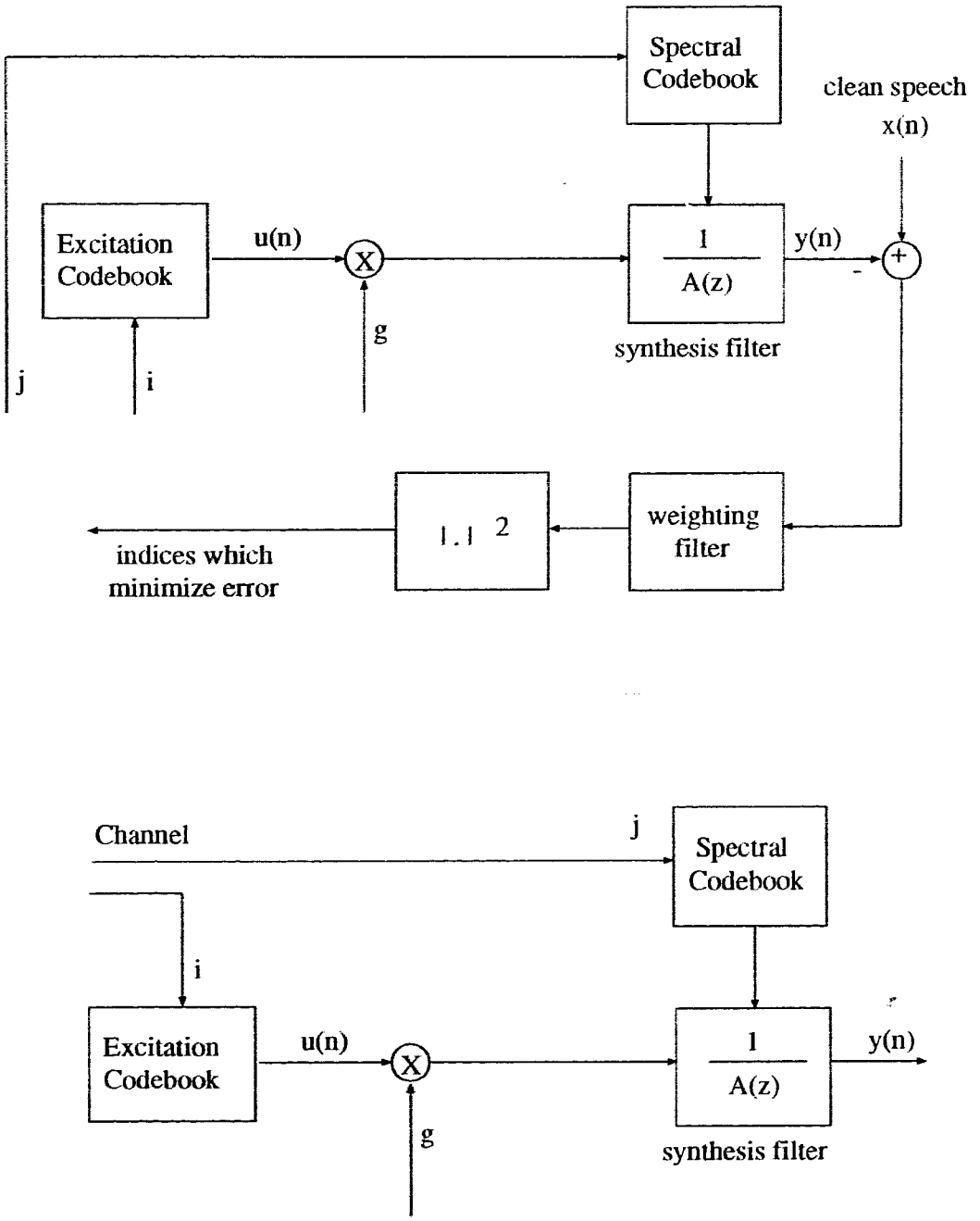


Figure 2.8: Analysis-by-synthesis coder

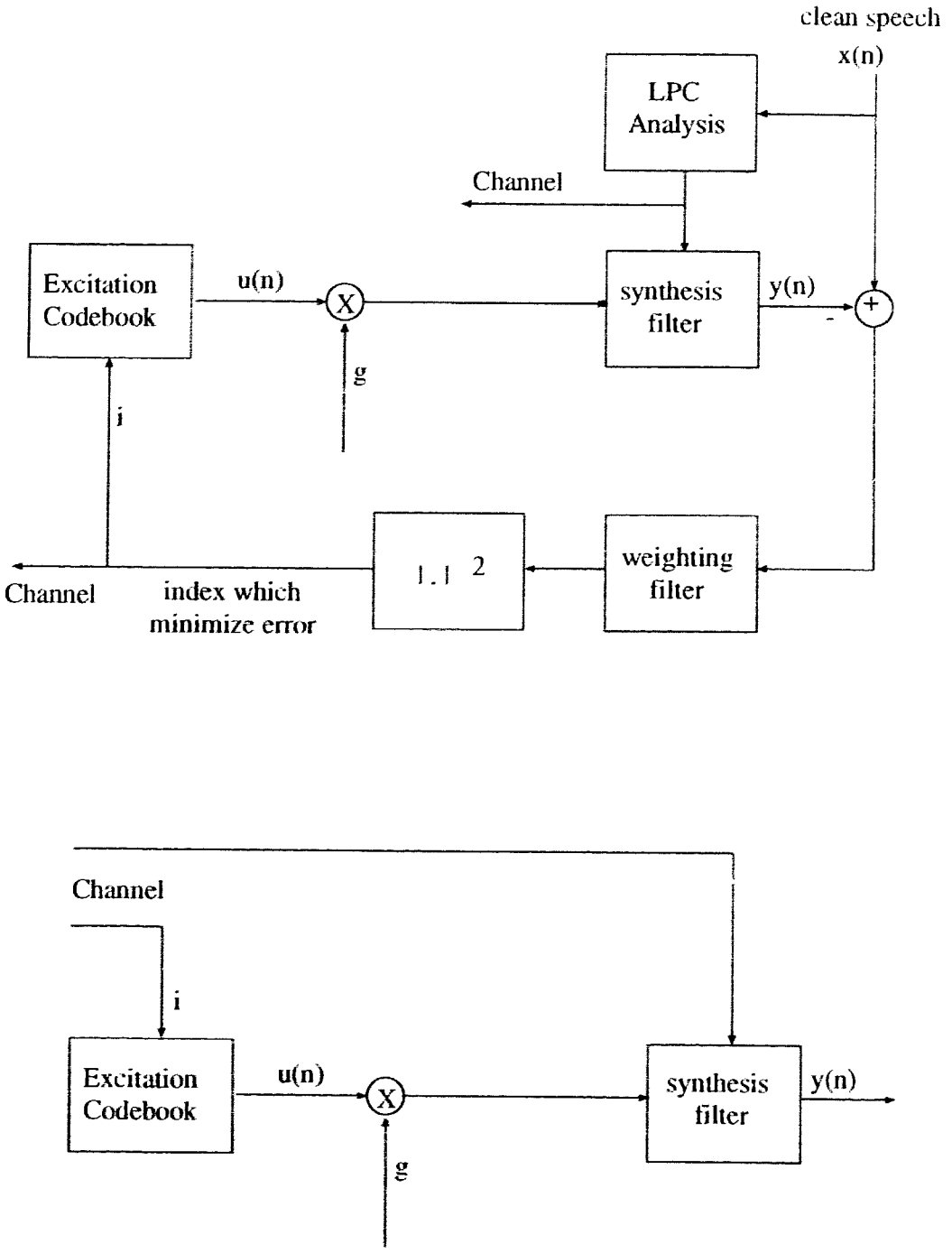


Figure 2.9: Reduced complexity analysis-by-synthesis coder

shape the noise by an approximation of the speech spectral shape. As the weighting is performed only at the encoder, the coefficients of the weighting filter are derived from the original signal with no requirement to transmit its parameters to the decoder.

Adaptive Postfiltering

The perceived speech quality may be improved in A-by-S coders by using an adaptive postfilter at the output of the speech decoder. A postfilter is essentially shaping the quantization noise that exists in the reconstructed signal to perceptually improve the reconstructed speech. This is done by passing the signal through a filter which has a shape that is derived from the transfer function of the short-term predictor, enabling the quantization noise to be attenuated in the valleys and amplified around the spectral peaks. A transfer function of a postfilter $P(z)$ can thus be given by

$$P(z) = \frac{1}{A(z/\gamma_2)} \tag{2.61}$$

where $0 < \gamma_2 < 1$. The effect of γ_2 is to dampen the spectral peaks by moving the poles towards the origin (smoothing the spectra). This in turn has the effect of increasing the spectral peaks bandwidth and so is known as bandwidth expansion.

The postfilter described above, though resulting in an improvement based on perceptual criteria, suffers from an effect known as muffling of the speech signal. This is due to the fact that the frequency response of the all-pole postfilter generally has a lowpass spectral tilt. This can mostly be removed by using a postfilter of the form [11]

$$P(z) = (1 - \Gamma z^{-1})\frac{A(z/\gamma_1)}{A(z/\gamma_2)} \tag{2.62}$$

The frequency response of the pole-zero filter can be considered as the difference of two all-pole filters with lesser amount of noise shaping for the numerator in the pole-zero filter. The first order filter section provides highpass spectral tilt to further reduce the muffling effect.

2.6 Code Excited Linear Prediction

Code excited linear prediction (CELP) has probably had the greatest impact on speech coding in the last two decades. Significant national and international standards are

based on the CELP coding algorithm. CELP was initially introduced by Atal and Schroeder [2]. Most of the techniques developed under the framework of A-by-S coders were applied to CELP. The amount of research done under CELP based coding schemes is too numerous to cite. Some of the critical aspects have been complexity reduction, robustness to channel errors, efficient representation of excitation and spectral information. A good source of reference is [27].

Figure 2.10 gives the block diagram of a general CELP coder with some essential features in place such as perceptual weighting and complexity reduction which will be discussed in the next subsection. The synthesis filter is considered as consisting of a short- and long-term predictor.

2.6.1 Excitation Codebook Search

Let the input signal $\mathbf{x}$ consist of blocks of N samples each. Let the vectors in the following analysis be column vectors. The synthesis filter operation on the excitation $\mathbf{u}$ is given by

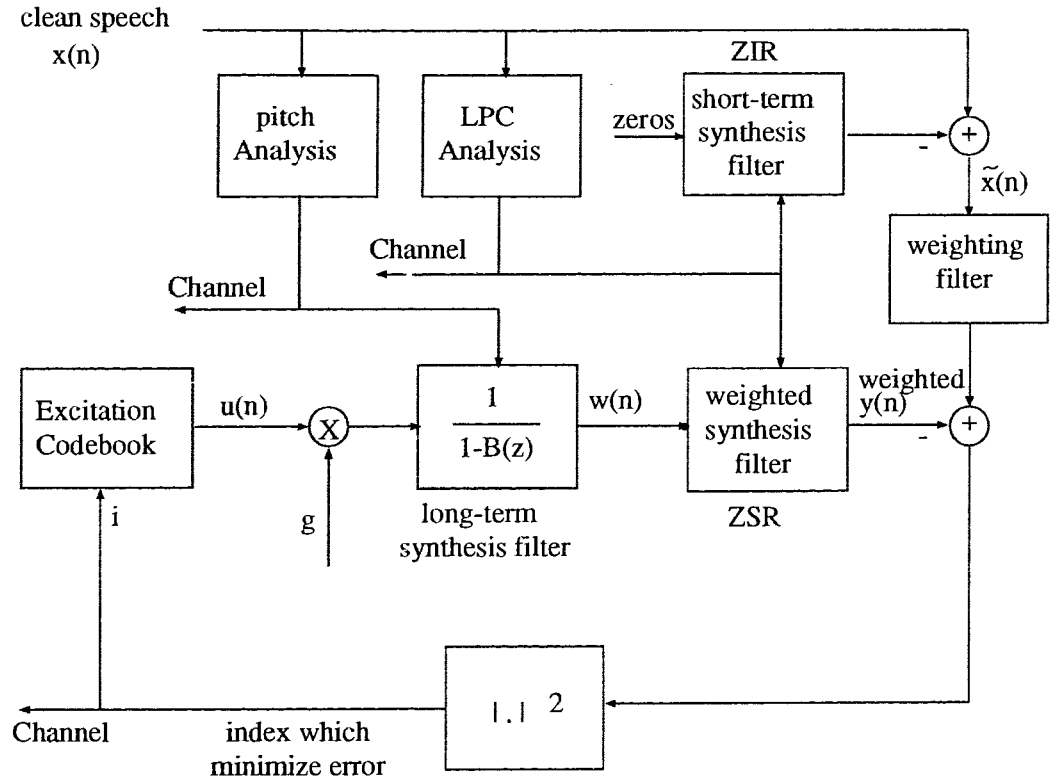
$$\mathbf{y}_i = H(g\mathbf{u}_i) \tag{2.63}$$

where g is the excitation gain and i is the excitation codebook index. H is an operation that gives the resulting synthesis filter output $\mathbf{y}_i$. The excitation codebook index is then selected by minimizing

$$\|\mathbf{e}_w\|^2 = \|W(\mathbf{x}) - W(H(g\mathbf{u}_i))\|^2 \tag{2.64}$$

where W is the weighting filter operation. This search procedures involves significant computational complexity. One very useful technique known as ZIR-ZSR decomposition attributed to Davidson and Gersho [17] can be used to reduce complexity. The complexity can also be reduced by dividing each frame of speech into a number of subframes to be then individually encoded.

The output of the synthesis filter can be written as the sum of the *zero state response* (ZSR) and the *zero input response* (ZIR) by using the linearity property of the synthesis filter. The ZIR is the output of the filter with zero input, hence does not depend on the choice of the vector in the excitation codebook. The ZSR is the output of the filter with the memory set to zero, so therefore does not depend on



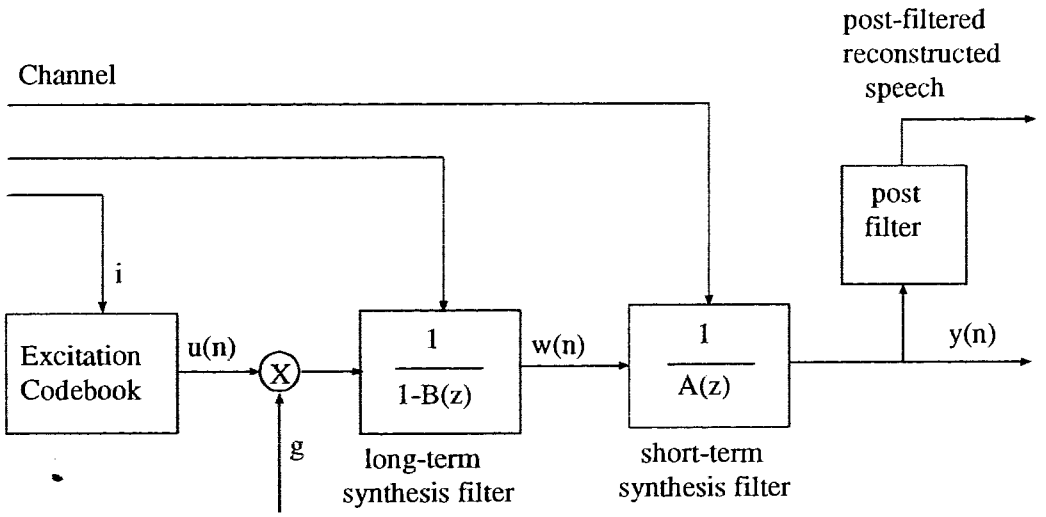


Figure 2.10: Reduced complexity CELP coder with a long-term predictor

past vectors. For the long-term predictor, if the pitch period is greater than the block size, the ZSR is unaffected. However if the pitch period is smaller, the effect of the long-term predictor has to be taken into account in the computation of the ZSR.

The weighting filter can be moved to be placed before the summation. The combined filter is now called the weighted synthesis filter and using the ZIR-ZSR decomposition, the weighted reconstructed signal $\mathbf{y}_{w_i}$ can be written as

$$\mathbf{y}_{w_i} = W(H(g\mathbf{u}_i)) = \mathbf{y}_{wzir} + g\mathbf{y}_{wzsr_i} = \mathbf{y}_{wzir} + g\mathbf{H}_w\mathbf{u}_i \tag{2.65}$$

where $\mathbf{H}_w$ is the impulse response matrix of the weighted synthesis filter $H(z) = W(z)/A(z)$ and is given as

$$\mathbf{H}_w = \begin{bmatrix} h_w(0) & 0 & 0 & \dots & 0 \\ h_w(1) & h_w(0) & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ h_w(N-1) & h_w(N-2) & \dots & \dots & h_w(0) \end{bmatrix} \tag{2.66}$$

Defining a target vector $\mathbf{t}$ in the codebook search by

$$\mathbf{t} = W(\mathbf{x}) - \mathbf{y}_{wzir} \tag{2.67}$$

where $\mathbf{y}_{wzir}$ is the ZIR of the weighted combined synthesis filter. The minimization problem is now reduced to finding the index i which minimizes the norm of the weighted error

$$\|\mathbf{e}_w\|^2 = \|\mathbf{t} - g\mathbf{y}_{wzsr_i}\|^2 \tag{2.68}$$

Using variational techniques, the optimal excitation gain $\hat{g}$ can be found as

$$\hat{g} = \frac{\mathbf{t}^T\mathbf{y}_{wzsr_i}}{\|\mathbf{y}_{wzsr_i}\|^2} \tag{2.69}$$

By replacing the gain in equation 2.68 with the optimal gain, the minimization reduces to

$$\|\mathbf{e}_w\|^2 = \|\mathbf{t}\|^2 - \frac{(\mathbf{t}^T\mathbf{y}_{wzsr_i})^2}{\|\mathbf{y}_{wzsr_i}\|^2} \tag{2.70}$$

The first term in equation 2.70 does not depend on the index i, so the optimization problem reduces to maximizing the second term. This can be considered as maximizing the normalized crosscorrelation between the target vector $\mathbf{t}$ and the ZSR codebook

entry $\mathbf{y}_{wzsr_i}$. The crosscorrelation $\mathbf{t}^T\mathbf{y}_{wzsr_i}$ can be more efficiently computed by filtering the target vector $\mathbf{t}$ by the filter H_w and then obtaining the cross product with the vector $\mathbf{u}_i$, instead of obtaining each filtered ZSR entry and then performing the crosscorrelation. The equivalence of these computational procedures is based on the equality

$$\mathbf{t}^T\mathbf{y}_{wzsr_i} = \mathbf{t}^T\mathbf{H}_w\mathbf{u}_i = \mathbf{u}_i^T\mathbf{H}_w\mathbf{t} \tag{2.71}$$

The code vector autocorrelation term can be precomputed and stored for fixed codebooks, resulting in computational savings in the energy computation $\|\mathbf{y}_{wzsr_i}\|^2$.

2.6.2 Closed Loop Pitch Prediction: Adaptive Codebook

Significant improvements in CELP coder performances were achieved by using an adaptive codebook [57] to replace the pitch filter in modeling the periodicity of voiced speech. The adaptive codebook is a closed-loop pitch predictor. The codebook consists of past excitation vectors with a pitch related delay k_p as an index in the codebook that determines the past excitation to use as a component of the current excitation. $w(n)$ is the combined excitation signal with $p(n)$ representing the pitch periodicity, while $u(n)$ is the stochastic part of the excitation signal $w(n)$. Figure 2.11 is a block diagram of a CELP encoder and decoder with an adaptive codebook replacing the long-term predictor.

Replacing the long-term filter by an adaptive codebook, the excitation equation is given by

$$w(n) = u(n) + p(n) = u(n) + \sum_{k=-(m-1)/2}^{(m-1)/2} g_{p_k} w(n - k_p - k) \tag{2.72}$$

where, $\mathbf{g}_p$ is the vector of tap-gains of the adaptive codebook and $p(n)$ is the adaptive codebook contribution of the excitation. It must be stressed that k_p here is not the pitch period but rather an index that minimizes the norm of the error

$$\|\mathbf{e}_{pw}\|^2 = \|\mathbf{t} - \sum_{k=-(m-1)/2}^{(m-1)/2} g_{p_k}\mathbf{H}_w\mathbf{w}_{k_p+k}\|^2 \tag{2.73}$$

where, $\mathbf{t}$ is the target vector in the adaptive codebook search and is given by equation 2.67 and $\mathbf{w}_{k_p+k}$ is a vector obtained from the excitation signal with a delay of $k_p + k$ samples.

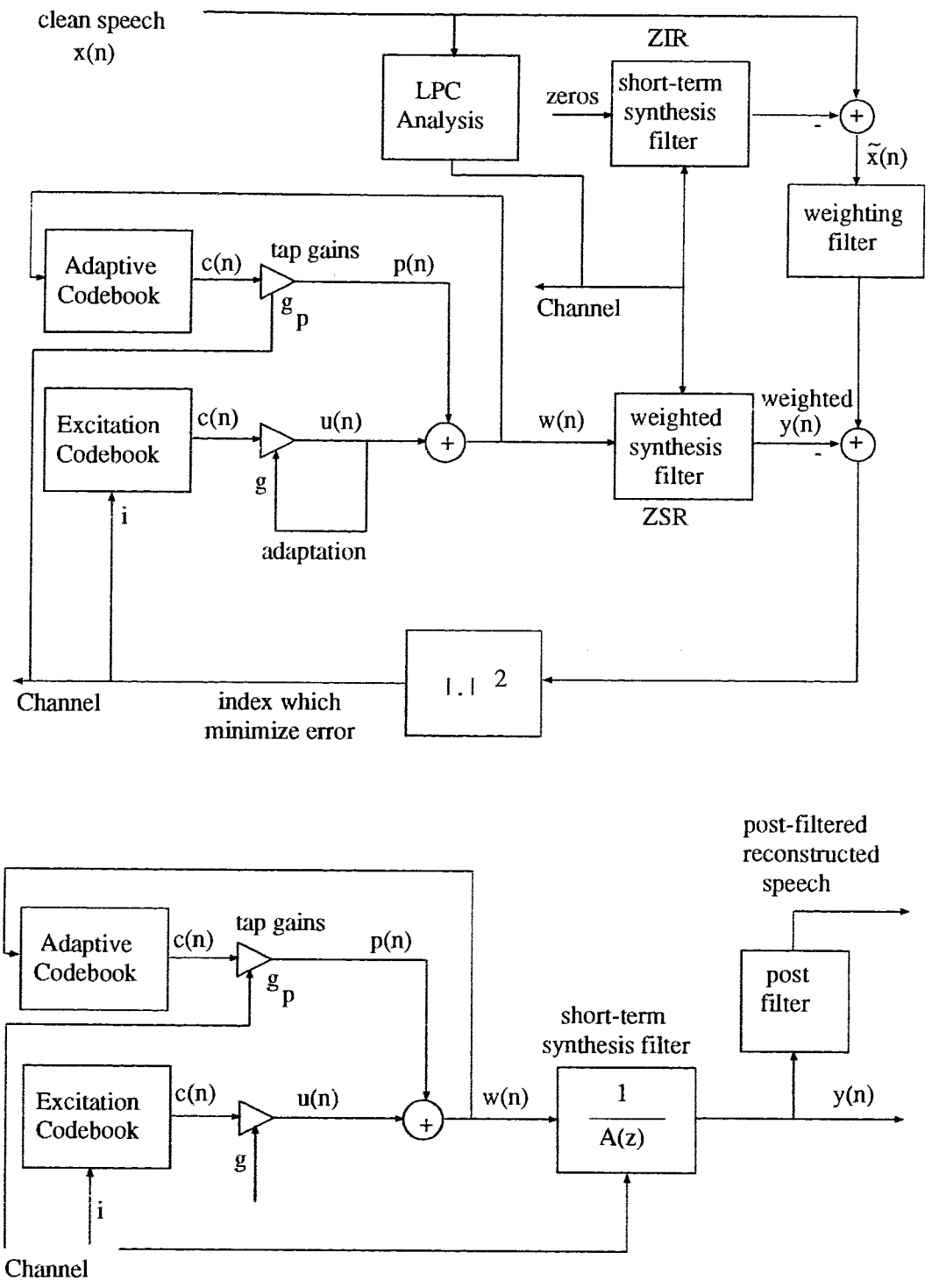


Figure 2.11: Reduced complexity CELP coder with an adaptive codebook

For the case of a 3-tap ($m = 3$) adaptive codebook, equation 2.73 can be written as

$$\|\mathbf{e}_{pw}\|^2 = \|\mathbf{t} - g_{p_{-1}}\mathbf{H}_w\mathbf{w}_{k_p-1} - g_{p_0}\mathbf{H}_w\mathbf{w}_{k_p} - g_{p_1}\mathbf{H}_w\mathbf{w}_{k_p+1}\|^2 \tag{2.74}$$

replacing $\mathbf{H}_w\mathbf{w}_{k_p}$ by $\mathbf{z}_{k_p}$, to denote the vector $\mathbf{w}_{k_p}$ filtered by the weighted short-term synthesis filter, equation 2.74 can be rewritten as

$$\|\mathbf{e}_{pw}\|^2 = \|\mathbf{t} - g_{p_{-1}}\mathbf{z}_{k_p-1} - g_{p_0}\mathbf{z}_{k_p} - g_{p_1}\mathbf{z}_{k_p+1}\|^2 \tag{2.75}$$

Using variational techniques, it can be shown that the optimal set of tap-gain coefficients $\mathbf{g}_p$ can be obtained by solving the system of equations given by

$$\begin{bmatrix} \mathbf{z}_{k_p-1}{}^T\mathbf{z}_{k_p-1} & \mathbf{z}_{k_p-1}{}^T\mathbf{z}_{k_p} & \mathbf{z}_{k_p-1}{}^T\mathbf{z}_{k_p+1} \\ \mathbf{z}_{k_p}{}^T\mathbf{z}_{k_p-1} & \mathbf{z}_{k_p}{}^T\mathbf{z}_{k_p} & \mathbf{z}_{k_p}{}^T\mathbf{z}_{k_p+1} \\ \mathbf{z}_{k_p+1}{}^T\mathbf{z}_{k_p-1} & \mathbf{z}_{k_p+1}{}^T\mathbf{z}_{k_p} & \mathbf{z}_{k_p+1}{}^T\mathbf{z}_{k_p+1} \end{bmatrix} \begin{bmatrix} g_{p_{-1}} \\ g_{p_0} \\ g_{p_{+1}} \end{bmatrix} = \begin{bmatrix} \mathbf{t}^T\mathbf{z}_{k_p-1} \\ \mathbf{t}^T\mathbf{z}_{k_p} \\ \mathbf{t}^T\mathbf{z}_{k_p+1} \end{bmatrix} \tag{2.76}$$

Once the optimal gain vector $\mathbf{g}_p$ is obtained, the index k_p is selected which minimizes equation 2.74. The target vector for the excitation codebook search $\tilde{\mathbf{t}}$ can then be obtained by subtracting from the original target vector $\mathbf{t}$, the adaptive codebook contribution of the weighted reconstructed speech signal as follows

$$\tilde{\mathbf{t}} = \mathbf{t} - \sum_{k=-1}^{1} g_{p_k}\mathbf{H}_w\mathbf{w}_{k_p+k} \tag{2.77}$$

This target vector is then used for the stochastic excitation codebook search previously discussed. This procedure is obviously suboptimal over a joint optimization of the stochastic excitation and adaptive codebook. However, joint optimization requires very high computational complexity. The joint search would attempt to minimize

$$\|\mathbf{e}_{jpw}\|^2 = \|\mathbf{t} - g\mathbf{y}_{wzsr_i} - g_{p_{-1}}\mathbf{z}_{k_p-1} - g_{p_0}\mathbf{z}_{k_p} - g_{p_1}\mathbf{z}_{k_p+1}\|^2 \tag{2.78}$$

Rather than performing just a sequential search as described above, a simple improvement can be obtained by reoptimizing the gains after the codevectors have been selected from the adaptive and excitation codebooks. Joint optimization can be performed efficiently using constrained orthogonal codebooks such as those used in VSELP [30]. The adaptive codebook and stochastic codebook idea can be considered as a special case of a multi-stage VQ. Multi-stage VQs were first used in CELP based systems by Davidson and Gersho [18]. Some further recent work on multi-stage codebook design is the work of LeBlanc [62].

2.6.3 State of the Art in Speech Coding

The state of the art in speech coding has been signifcantly motivated by past and present standardization activity. In this section, a brief history of current standards will be presented together with a look into the evolving standards of the future. Currently, CELP based systems have provided a basis for most speech coding standardization activity, although recently, vocoder type models are playing an increasing role in future evolving standards. Standardization activity can be classified into two categories: wireline and wireless.

Wireline activity is further broken into low, medium and high delay standards. Low delay coding standardization activity has been the prime motivation of the ITU. ADPCM at 32 kb/s (G.721) was adopted as an ITU standard in 1984. At 16 kb/s, LD-CELP was chosen as the ITU standard (G.728) [8, 12]. The ITU 8 kb/s standard (G.729) based on conjugate structure algebraic CELP (CS-ACELP) was in its final approval stage by the ITU in November 1995 [89]. G.729 is actually a medium delay coder as the low-delay requirements were relaxed somewhat during the coder selection competition.

Higher delay coders have also been recently selected for various applications. The U.S Department of Defence (DoD) selected a 4.8 kb/s CELP coder as federal standard (FS-1016) for secure voice communications [50]. At 2.4 kb/s, LPC-10 is used as a U.S government federal standard (FS-1015) for low-rate secure voice communications.

Wireless standardization activity has revolved around the time division multiple access (TDMA) and code division multiple access (CDMA) digital cellular telephony schemes. The global system for mobile telecommunications (GSM) subcommittee of the European Telecommunications Standards Institute (ETSI) chose 13 kb/s residual pulse excited LPC (RPE-LPC) as the European TDMA digital cellular standard [36]. ETSI has recently adopted a half-rate TDMA digital cellular standard at 6.5 kb/s based on CELP [104]. Currently ETSI is looking towards standardizing an enhanced full-rate codec. The likely candidate is based on multi-pulse and transformed binary pulse CELP (MT-CELP) at 13 kb/s, which has already been selected in North America as the basis for a GSM based PCS1900 system [74]. Vector sum excited linear prediction (VSELP) at 8 kb/s was selected by the Telecommunications Industry Association (TIA) for use as the North American TDMA digital cellular standard (IS-54) [30]. A modified

form of VSELP at 6.7 kb/s was chosen as the Japanese TDMA digital cellular standard. Recently, pitch synchronous innovation CELP (PSI-CELP) at 3.6 kb/s has been chosen as the half-rate Japanese TDMA digital cellular standard [73]. A variable-rate coding algorithm known as Qualcomm CELP (QCELP) was chosen as the CDMA digital cellular telephony standard IS-95 by the TIA [25]. Improved MBE (IMBE) at 6.4 kb/s was adopted by Inmarsat as a standard for land mobile satellite voice communications to provide one of few non CELP based standards [45].

In Figure 2.12, MOS values are plotted for various current coding standards in clean conditions, to give a qualitative idea of current technology. Currently, standardization is being actively pursued at rates of 2.4 to 4 kb/s. A good reference source for the state of the art in speech coding can be obtained from proceedings of the IEEE international workshops on speech coding [44].

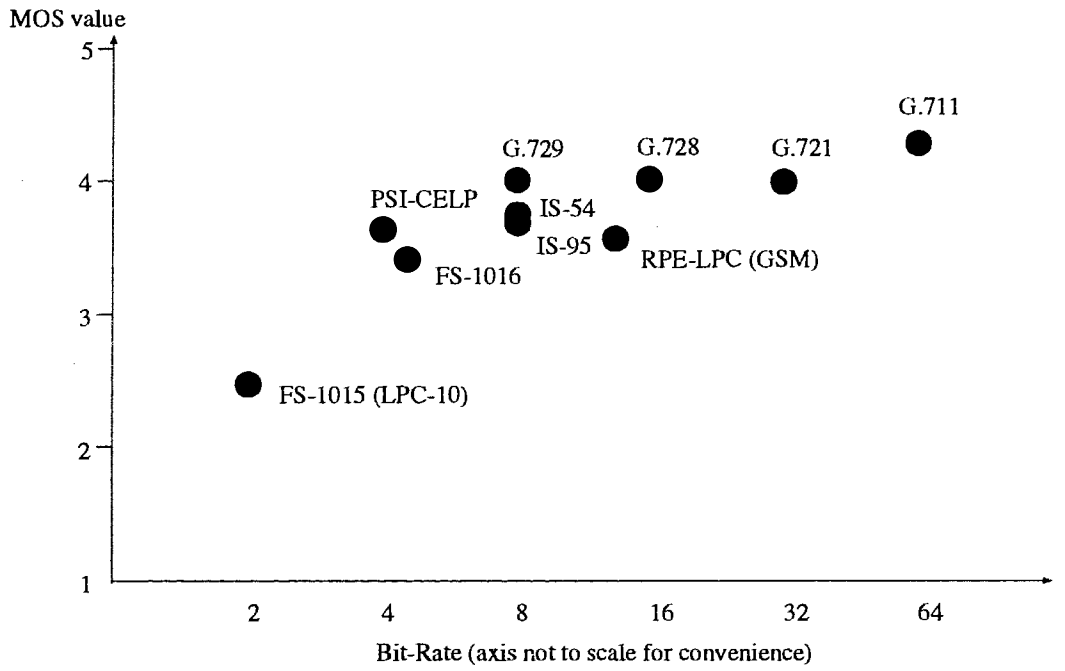


Figure 2.12: MOS for current speech coding standards

Wireline standardization activity currently include the ITU-T competition for a future 4 kb/s speech coding standard with a quality requirement not worse than G.721 [21]. Also the U.S DoD is currently looking to replace the LPC-10 standard with a new coding standard at 2.4 kb/s that will achieve the quality of the existing FS-1016 standard [96].

Wireless activity is primarily centered on replacing existing cellular standards.

The TIA is currently evaluating candidate algorithms for a North-American half-rate TDMA digital cellular standard at 4 kb/s [44]. The IS-54 TDMA digital cellular standard is possibly being replaced by the ITU-T standard G.729, while the IS-95 CDMA digital cellular standard is also possibly being replaced by a coder under development.

Chapter 3

Low-Delay Speech Coding

Recently, communication delay has become an important performance criterion for speech encoders used in the public switched telephone network (PSTN), as efforts are being intensified to achieve toll quality at rates as low as 8 kb/s, to replace existing higher rate systems. In a complex network, the delays of many encoders add together, transforming the delay into a significant impairment of the system. Delay may necessitate the use of echo cancellation and in some applications it remains an impairment even after echo cancellation has been performed. For these reasons, the proposed 8 kb/s ITU-T standard specifies low delay as a major requirement. The ITU-T was formerly known as the Consultative Committee International Telephone and Telegraph (CCITT).

The delay performance of a speech coder is characterized by the algorithmic delay and processing delay. Algorithmic delay (also known as buffering delay) is the one-way delay of the encoder and the decoder assuming infinite processing power for the coder implementation. Processing delay is the additional delay due to implementation with a finite processing power. The total codec delay is the sum of algorithmic delay and processing delay. The transmission channel delay (time required for a bit presented to the transmitter to appear at the output of the receiver) caused by transmitting over a finite bandwidth serial channel adds to the total codec delay to give the delay encountered in practical applications. The requirements of the new 8 kb/s ITU-T standard specified an objective buffering delay lower than 5 ms and a total codec delay of 10 ms. The ITU-T later revised the buffering delay requirement to 16 ms, with the selected coder based on CS-ACELP having a delay of 10 ms [89]. The 16 kb/s

standard has a total delay lower than 2 ms.

Conventional CELP [2], or vector excitation coding (VXC) [19], both achieve good speech quality; however these coders introduce a substantial delay due to the forward adaptation of the short-term predictor. The input buffering delay, typically 20 ms at 8 kHz, and other processing delays, typically result in a total codec delay of 50 to 60 ms. Although a total delay of 32 ms may be obtained by redesigning the standard CELP configuration for a different delay/quality tradeoff, lower delays (which can meet or exceed the objective of 10 ms total codec delay) and good quality can be obtained by using a backward adaptive configuration.

A possible approach to low-delay speech coding is based on tree/trellis search techniques applied to a backward adaptive configuration [47, 68, 31]. An alternative approach is based on a backward adaptive A-by-S model. In a backward adaptive A-by-S configuration, the parameters of the synthesis filter are not derived from the original speech signal, but instead computed by backward adaptation, extracting information only from the reconstructed signal based on the transmitted excitation information. Since both the encoder and decoder have access to the past reconstructed signal, side information is no longer needed for the synthesis filter, and the low-delay requirement can be met with a suitable choice of frame size.

Backward adaptive A-by-S configurations are used for speech coding at 16 kb/s in [101, 15, 80, 8]. The G.728 16 kb/s ITU-T speech coding standard is based on LD-CELP and achieves toll quality with a delay lower than 2 ms using a block backward adaptive configuration without pitch prediction [8]. The lattice low-delay VXC (LLD-VXC) achieves similar performance using backward adaptive lattice adaptation and backward adaptive pitch prediction [80].

Recently, several low-delay 8 kb/s coders have been proposed which achieve close to toll quality. These include versions of LD-CELP and LD-VXC [109, 13, 54, 38, 39], and a tree encoder based on backward adaptive prediction [103]. All the coders achieve similar qualitative performance but these coders do not attain the performance of the G.728 LD-CELP coder at 16 kb/s. Progress in improving backward adaptive configuration robustness on noisy channels was recently achieved by making use of shaping filters [103, 39]. Toll quality has recently been achieved at 8 kb/s by making use of medium delay coders [89]. The delay requirements were relaxed in order to achieve the stringent quality requirements.

Section 3.1, presents a review of the G.728 standard (LD-CELP). In Section 3.2, the LD-VXC and LLD-VXC coding systems at 16 kb/s are reviewed. Finally in Section 3.3, two versions of the 8 kb/s LLD-VXC codec are presented: a backward 8 kb/s coder, which makes use of a 3-tap hybrid backward adaptive open-loop pitch predictor [82, 14] and a partially-forward scheme, which uses a 3-tap forward adapted long-term adaptive codebook. These two codecs are compared in clean and noisy channel conditions.

3.1 LD-CELP

The LD-CELP G.728 standard is briefly presented here. The LD-CELP coding algorithm is based on an alternative backward adaptive coding configuration when compared to the LD-VXC approach (to be discussed in Section 3.2). Some of the critical differences are the removal of a pitch predictor and the use of a block adaptive short-term predictor with the adaptation done using the autocorrelation method with a recursively updated hybrid windowing technique. For a detailed description of the standard, the reader can refer to the G.728 draft recommendation [46] or papers by Chen on LD-CELP [8, 9, 12].

3.1.1 System Overview

The block diagram of the LD-CELP system is shown in Figure 3.1. At the encoder side, an A-by-S technique is used for selecting the optimum excitation codevector from a codebook. The excitation codebook structure consists of a gain-shape codebook. A gain-shape codebook generates the excitation vectors by multiplying vectors in a shape sub-codebook by gain values stored in a gain sub-codebook. In LD-CELP the speech signal is partitioned into blocks of 5 consecutive input speech samples. The excitation codebook has 10 bits available for coding each block (referred to as a vector). The codebook excitation consists of 7-bit shape and 2-bit signed gain codebooks. Four consecutive vectors form a frame, hence the frame size is 20 samples.

In the block diagram of the LD-CELP system shown in Figure 3.1, each candidate excitation codevector $\mathbf{c}_i$ is multiplied by a gain, and the resulting vector, $\mathbf{u}$, is fed into a 50-th order short-term synthesis filter. The gain is the product of the predicted

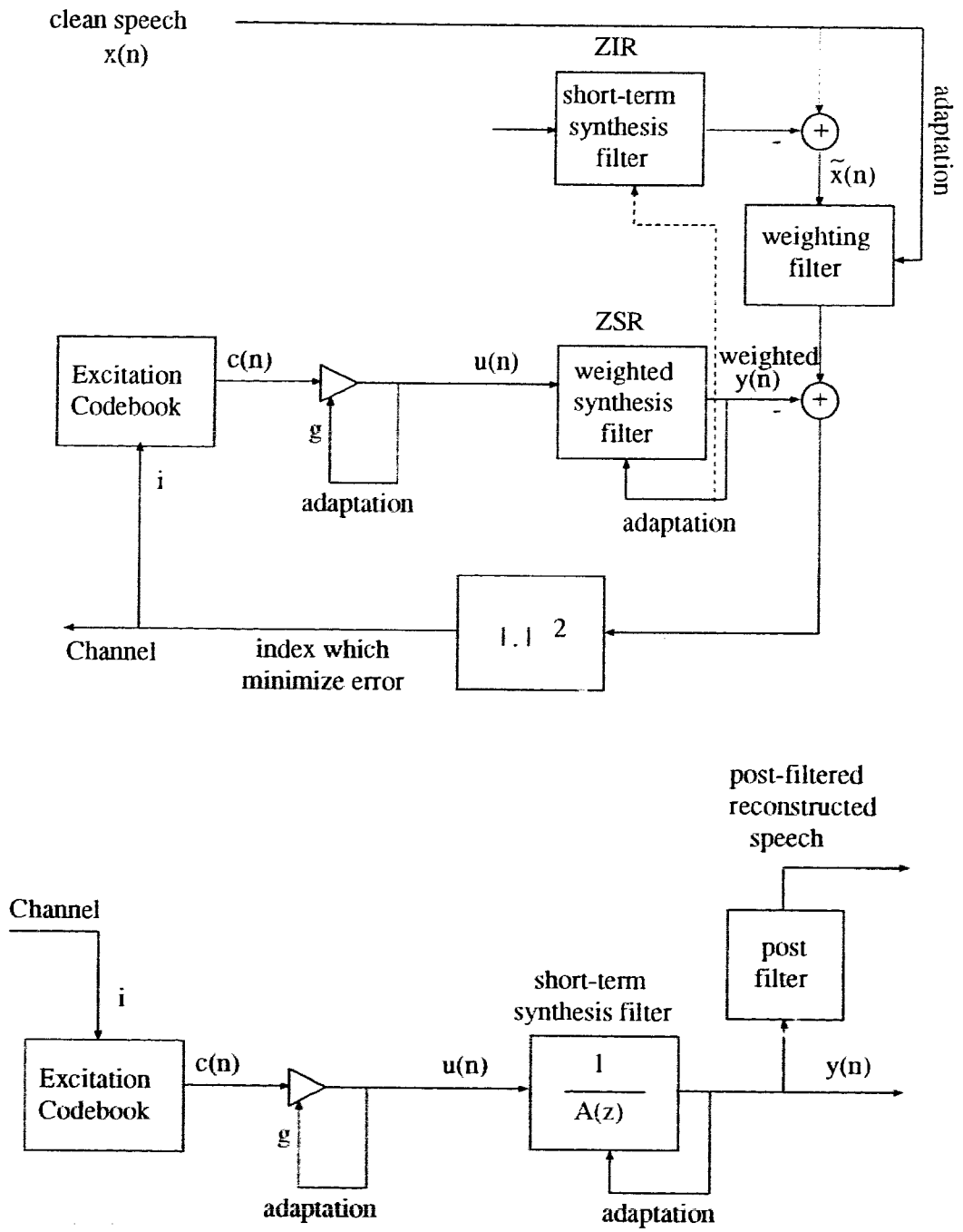


Figure 3.1: LD-CELP encoder and decoder configuration

gain obtained from the backward adaptive gain predictor and the gain value obtained from the gain codebook. The output of the synthesis filter, $\mathbf{y}$, is then compared to the actual speech signal, $\mathbf{x}$, and the best candidate codevector is selected using a perceptually weighted minimum MSE criterion.

The pitch predictor was eliminated to obtain greater robustness to bit errors at rates as high as 10^{-2}. The justification being that it was unlikely that backward adaptive pitch predictors could be robust at high error rates. However, eliminating the pitch predictor significantly affected female speech quality, although the effect on male speech was much less noticeable. To compensate for the loss in quality, the LPC predictor order was set at 50, as the prediction gain was found to saturate at predictor orders around 50 [8, 46]. The short-term predictor is attempting to model some of the pitch periodicity in the speech signal. The excitation codebook then models any remaining periodicity in the residual signal $\mathbf{u}$. It was found that for certain speech signals with a very large pitch period, the lack of pitch prediction in the LD-CELP coder results in an impulsive shape of the residual error signal.

3.1.2 Hybrid Windowing

The synthesis and perceptual weighting filters are adapted every frame (4 vectors). The weighting filter adaptation is based on 10-th order linear prediction analysis of the unquantized speech, while the synthesis filter adaptation makes use of the reconstructed speech.

The filter adaptation can be considered to be broken up into three modules: hybrid windowing; Levinson-Durbin recursion; and filter coefficient calculation. The hybrid windowing module places a recursive window on previous speech vectors and calculates the appropriate order autocorrelation coefficients of the windowed signal. This is a modified version of the recursive windowing method developed by Barnwell [3]. The idea behind using a recursive window is to reduce computational complexity in performing the windowing operation in the calculation of the estimates of the autocorrelation functions. This is achieved by computing the autocorrelation function estimates recursively. The hybrid window consists of a recursive and non-recursive part, which results in a recursive and non-recursive part of the autocorrelation function. The recursive part of the autocorrelation estimate can be used in the next

adaptation block calculation. Levinson-Durbin recursion is then performed, followed by bandwidth expansion on the computed linear prediction coefficients to ensure a more robust predictor in the event of channel errors.

3.1.3 Backward Vector Gain Adapter

Low delay speech coders use backward adaptive linear prediction for excitation gain evaluation. The gain is given by the backward adaptive gain predictor, which is based on the technique developed by Chen and Gersho [10] obtained by generalizing the Jayant robust quantizer citejaya84. A further generalization may be obtained by making use of an adaptive predictor in the logarithmic domain. The advantage of the log-gain predictor is that better performance can be achieved in clean channel conditions.

The goal of the log-gain predictor is to predict the log of the root-mean-square value of $u(n)$, $\log \sigma_u(n)$, based on the previous 10 values $\log \sigma_u(n-1), ..., \log \sigma_u(n-10)$. Denoting by $\sigma(n)$ the gain at the vector time index n, the corresponding adaptation equation can be written in logarithmic scale as

$$\log \sigma(n) = \sum_{i=1}^{10} a_{gp_i} \log |u(n-i)| \tag{3.1}$$

The coefficients of the log-gain predictor, a_{gp_i}, are obtained by using a hybrid windowing technique as is done for the perceptual weighting and short-term synthesis filters. The coefficients are computed every 4 vectors as is done for the other two filters. To improve robustness, bandwidth expansion is performed on the poles.

3.2 LD-VXC at 16 kb/s

In this section, LD-VXC at 16 kb/s and its subsequent successor, LLD-VXC at 16 kb/s, will be briefly discussed. Both of these techniques are based on a backward adaptive A-by-S configuration. For more detailed information on the LD-VXC and LLD-VXC coders at 16 kb/s the reader can refer to [15, 81, 14] and [80] respectively. The LD-VXC coder was itself derived from the vector ADPCM technique developed by Watt and Cuperman [101] which was the first application of backward adaptation to excitation coding.

3.2.1 System Overview

The block diagram of an LD-VXC system is shown in Figure 3.2. At the encoder side, an A-by-S technique is used for selecting the optimum excitation codevector from a codebook. The excitation codebook structure consists of a shape-only codebook.

In the diagram shown in Figure 3.2, each candidate excitation codevector $\mathbf{c}_i$ is multiplied by a gain, and the resulting gain scaled vector, $\mathbf{u}$, is fed into the synthesis filter. The gain is given by the backward adaptive gain predictor which is similar to the gain predictor used in LD-CELP, with the primary difference being that the coefficients are fixed in LD-VXC. The components of the vector $\mathbf{u}$ will be denoted by $u(n)$. The output of the synthesis filter, $\mathbf{y}$, is then compared to the actual speech signal, $\mathbf{x}$, and the best candidate codevector is selected using a perceptually weighted minimum MSE criterion based on the weighting filter, $W(z)$.

The index of the optimal excitation sequence is then transmitted to the decoder. At the decoder side, the received indices are used to generate the proper excitation sequence. The excitation codevector is then gain scaled using the gain computed in the same way as it is done in the encoder, and fed into the cascade of the pitch and formant synthesis filters. The output of the cascade of synthesis filters is the reconstructed speech which may be postfiltered to reduce the perceived quantization noise.

LD-VXC at 16 kb/s has available 2 bits per sample for representing the excitation shape. To achieve the low-delay requirement with low implementation complexity, LD-VXC uses a codebook of dimension 4 samples, which translates to a codebook of size 256 (8 bits per vector).

3.2.2 Synthesis Filters

The output of the short-term predictor, $y(n)$, is computed using the relationship:

$$y(n) = w(n) + \sum_{i=1}^{m_p} a_{p_i} y(n-i) + \sum_{i=1}^{m_z} a_{z_i} w(n-i) \tag{3.2}$$

where, a_{p_i} are the coefficients of the all-pole section and a_{z_i} are the coefficients of the all-zero section. The output of the long-term predictor is likewise obtained by the

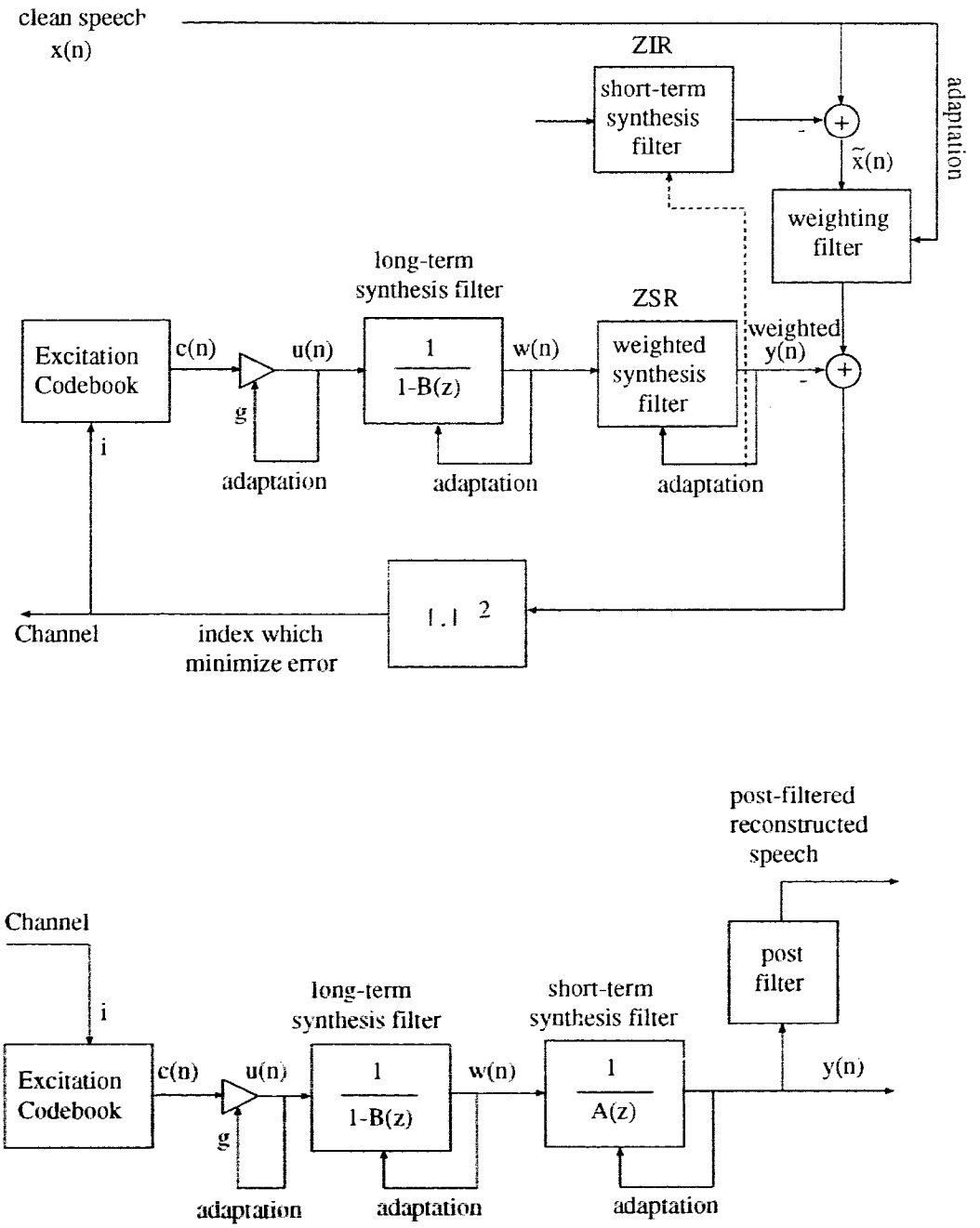


Figure 3.2: Backward LD-VXC configuration

relationship:

$$w(n) = \sum_{k=-1}^{1} b_k u(n - k_p - k) \tag{3.3}$$

A comparison between block and recursive adaptation in the backward adaptive A-by-S configuration has shown that recursive adaptation results in a performance that is comparable to that of block adaptation, but with lower complexity [15]. In block adaptation of the short- and long-term predictors in backward adaptive systems, the prediction parameters are computed from a previous block of reconstructed speech, under the assumption that the parameters are varying slowly enough that the parameters obtained from the previous reconstructed signal are close to those for the current block, thus suggesting smaller block sizes for better performance. However, the analysis frame has to be large enough to be able to compute the coefficients accurately. This results in conflicting requirements in turn resulting in a complex system.

In block adaptive systems, which traditionally use an all-pole short-term predictor, the autocorrelation method can be applied to adapt the short-term predictor coefficients by solving the Weiner-Hopf equations, and the resulting filter is always stable. In backward block adaptation of the long-term predictor, equation 2.38 is modified to include a $1 + \nu$ term in the leading diagonal and is written as

$$\begin{bmatrix} (1+\nu)r_0 & r_1 & r_2 \\ r_1 & (1+\nu)r_0 & r_1 \\ r_2 & r_1 & (1+\nu)r_0 \end{bmatrix} \begin{bmatrix} b_{-1} \\ b_0 \\ b_{+1} \end{bmatrix} = \begin{bmatrix} r_{k_p-1} \\ r_{k_p} \\ r_{k_p+1} \end{bmatrix} \tag{3.4}$$

The term ν is known as a softening factor and has a typical value of 0.03. This is analogous to the addition of white noise to the pitch predictor output signal. Such a factor would degrade performance in a forward adaptive system, where the predictor is optimized for the current block of data. However, in a backward adaptive system, the softening of the predictor is known to improve prediction gain.

Recursive adaptation on the other hand, makes use of the gradient algorithm for solving the prediction error minimization problem. From the theory discussed in Section 2.3.4, the adaptation equations can be written as [15]

$$a_{p_k}^{j+1} = a_{p_k}^{j} + \mu_{a_p} u(n) y(n-k) \tag{3.5}$$

$$a_{z_k}^{j+1} = a_{z_k}^{j} + \mu_{a_z} u(n) w(n-k) \quad (3.6)$$

$$b_k^{j+1} = b_k^{j} + \mu_b u(n) w(n-k_p-k) \quad (3.7)$$

The adaptation of the filter based on the equations above is not robust in the presence of transmission errors. Three modifications were found to improve significantly the robustness under bit errors at the expense of a slight degradation in the performance in clean channel conditions [15]. First, the use of leakage factors for all adapted coefficients improves the robustness. Second, the robustness was improved by using $u(n-k)$ in place of $w(n-k)$ in the equation for adapting the a_{z_k}. This can be considered as parallel adaptation, since adaptation no longer needs to be done in cascade as a result of generating the intermediate signal $w(n)$. Third, by making use of an all-zero reconstructed signal $\hat{y}(n-k)$ for adapting a_{p_k}, robustness was improved by eliminating the long error response of the all-pole component. The adaptation equations are now given by

$$a_{p_k}^{j+1} = \lambda_{a_p} a_{p_k}^{j} + \mu_{a_p} u(n) \hat{y}(n-k) \quad (3.8)$$

$$a_{z_k}^{j+1} = \lambda_{a_z} a_{z_k}^{j} + \mu_{a_z} u(n) u(n-k) \quad (3.9)$$

$$b_k^{j+1} = \lambda_b b_k^{j} + \mu_b u(n) w(n-k_p-k) \quad (3.10)$$

where λ is the appropriate leakage factor. Further robustness can be achieved by using the *sign algorithm* [15].

3.2.3 Hybrid Pitch Predictor Adaptation

The LD-VXC system makes use of hybrid backward adaptation of the pitch predictor [82, 14, 81]. As the name implies, this is a hybrid combination of backward block adaptation and backward recursive adaptation. Backward block or recursive adaptation both suffer in performance due to limiting tradeoffs. The use of hybrid adaptation allows a much larger update interval for the block adaptation algorithm, resulting in lower complexity. Hybrid backward adaptation also results in improved performance over backward block adaptation, due to the ability to track changing signal statistics over the duration of a block. This hybrid scheme is used in the LLD-VXC system at 16 kb/s and the subsequent LLD-VXC schemes at 8 kb/s developed in this thesis.

The hybrid adaptation operates in the following manner: the block adaptation method is used to compute the pitch period and the filter coefficients at the start of

each frame using data from the previous frame. These values are used to initialize the pitch tracker and the filter coefficients, that are then adapted recursively for the duration of the frame [82, 81]. When the pitch period is computed at the beginning of each frame, a voiced/unvoiced decision is made. If the previous frame of data contains unvoiced speech then the pitch period and filter coefficients are not initialized. This is done to prevent the pitch tracker from being initialized with an inaccurate pitch value. The frame size for block updating used in the 16 kb/s LD-VXC scheme was typically 256 samples.

The filter coefficient adaptation has already been discussed in Section 3.2.2. The LD-VXC scheme uses the adaptive step-size gradient algorithm, based on estimates of the variance of the predictor input and output. The adaptation equation for the long-term predictor coefficients now becomes

$$b_k^{n+1} = \lambda_b b_k^n + \frac{\mu_b}{\sigma_u \sigma_w} u(n) w(n - k_p - k) \tag{3.11}$$

where μ_b is the step-size constant and σ's are the signal variances, which are estimated using the running-average algorithm:

$$\sigma_x^2(n) = \delta \sigma_x^2(n-1) + (1-\delta) x^2(n) \tag{3.12}$$

where $\delta = 0.1$ determines the running-average memory.

The pitch period is estimated by making use of the autocorrelation method [84], with the output of the pitch prediction filter as the input signal. Several alternative methods are presented in [84]. The predictor output $w(n)$ is centre clipped to obtain a clipped signal. The autocorrelation function for the clipped signal is computed for all lags in the range $20 - 125$. The pitch period k_p is determined by finding the lag that maximizes the autocorrelation function. A decision is made on whether the speech is voiced or unvoiced by applying a threshold to the peak of the normalized autocorrelation function.

The pitch period is tracked recursively using the autocorrelation pitch tracker [82, 81]. The autocorrelation pitch tracker uses a running average computation of the autocorrelation function, evaluated at the three lags $k_p - 1$, k_p and $k_p + 1$ to track the pitch period. The autocorrelation tracker utilizes the fact that the pitch period should correspond to the lag at which the autocorrelation function is a maximum.

The estimate of the normalized autocorrelation function ρ_{ww}^k can be obtained from the following recursion:

$$\rho_{ww}^k(n) = \delta\rho_{ww}^k(n-1) + (1-\delta)\frac{w(n)w(n-k)}{\sigma_w^2(n)} \tag{3.13}$$

where $\delta = 0.9$ is the parameter that decides the running average memory. After each update, a decision is made to increment or decrement the pitch period by one, if ρ_{ww}^k, where $k = k_p + 1$ or $k = k_p - 1$ respectively, is the maximum value of the three autocorrelation values and is greater than a minimum threshold ρ_{min}. The constant ρ_{min} is to avoid tracking in unvoiced regions and a value of $\rho_{min} = 0.2$ is used.

If the pitch period is modified, then the values of the estimate of the autocorrelation function and the predictor coefficients are shifted by one in the appropriate direction, and the new autocorrelation function and filter coefficient are computed to be a constant fraction of $\rho_{ww}^{k_p}$ (constant typically 0.3) and b_0 (constant typically 0.67) respectively.

3.2.4 Lattice LD-VXC at 16 kb/s

The LLD-VXC system was introduced as a possible tradeoff between the requirements for accurate spectral modeling and the resulting computational complexity. The LD-VXC system proposed in [15] used a two-pole six-zero short-term predictor, which led to low computational complexity; however, it had the disadvantage of poor spectral modeling of the speech signal. On the other hand, the LD-CELP algorithm [46] used a 50-th order short-term predictor, which achieved good spectral modeling but at the expense of added computational complexity.

The lattice structure is a possible tradeoff of these conflicting requirements. For recursive adaptation configurations, the use of a lattice filter as the short-term predictor has significant advantages such as, faster tracking of coefficients, simple stability checks, and uniform distribution of computational load.

LLD-VXC at 16 kb/s with a pitch predictor combined with a 20-th order short-term predictor, achieved practically the same subjective performance as the LD-CELP algorithm, but at a lower complexity [80]. It was found that the LLD-VXC coder performance at 16 kb/s saturates for short-term predictor orders in the range 20-30 [80].

3.3 Lattice Low-Delay Vector Excitation Coding at 8 kb/s

In this thesis, two versions of the 8 kb/s LLD-VXC codec are presented: a backward 8 kb/s coder, which makes use of a 3-tap hybrid backward adaptive open-loop pitch predictor [82, 14] and a partially-forward scheme, which uses a 3-tap forward adapted long-term adaptive codebook. Also, the effect of coefficient adaptation on speech quality under clean and noisy channel conditions is investigated. The aim was to develop and present two alternative approaches towards achieving a high quality, low-delay coder which was robust to channel errors. The key contributions of this research were to present a comparison of the performance of backward adaptation with the performance of a partially-forward adaptation system for low-delay speech coding under clean and noisy conditions. A bit-allocation strategy was developed towards designing these coders to achieve greater performance. A delta pitch encoder was developed to encode pitch with fewer bits in voiced frames for the partially-forward system. Also, as part of this effort, robustness to channel errors was examined with a view to designing a speech coder that was more robust to channel errors with the adaptation system being used. This was achieved by making use of a "new" adaptation signal in filter adaptation.

3.3.1 System Overview

The block diagram of the fully backward 8 kb/s LLD-VXC system is shown in Figure 3.3. Two different excitation codebook structures are used in the 8 kb/s LLD-VXC: gain-shape or shape-only codebooks. A gain-shape codebook generates the excitation vectors by multiplying vectors in a shape sub-codebook by gain values stored in a gain sub-codebook. In a shape-only codebook, the gain sub-codebook has only one entry. This approach differs from the approach used in the 16 kb/s LD-VXC and LLD-VXC systems, which use a shape-only codebook. The system overview of the encoder and decoder in the fully backward 8 kb/s system is the same as that for the 16 kb/s LD-VXC system.

The block diagram of the partially-forward LLD-VXC system is shown in Figure 3.4. In the partially-forward system an additional gain codebook is used, which consists

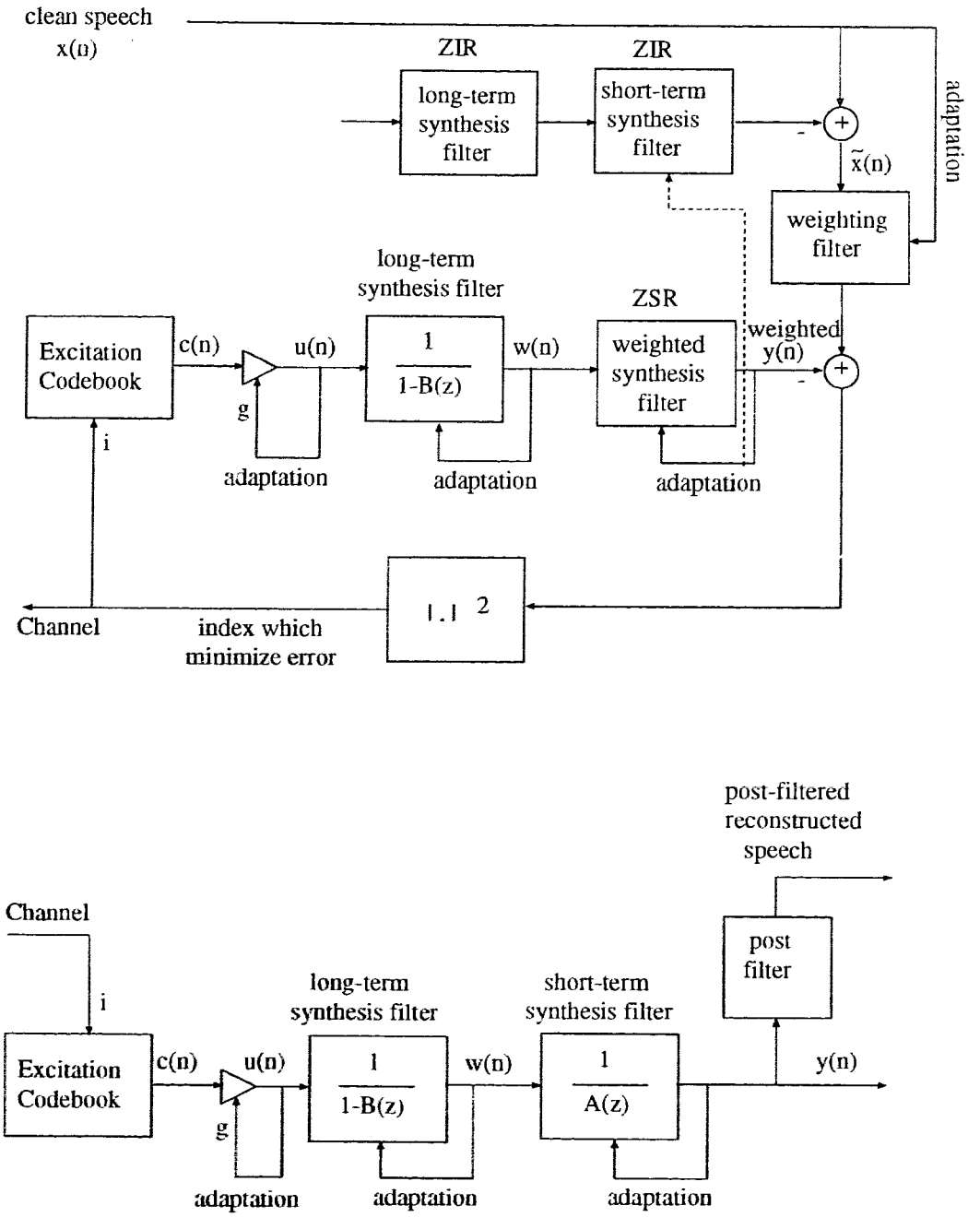


Figure 3.3: Backward LLD-VXC configuration at 8 kb/s

of the tap gains of the 3-tap adaptive codebook.

In the diagram shown in Figure 3.4, each candidate excitation codevector $\mathbf{c}_i$ is multiplied by a gain. The resulting vector, $\mathbf{u}$, is then added to the output vector of the adaptive codebook, $\mathbf{p}$, to give the excitation vector, $\mathbf{w}$, which is fed into both the short-term synthesis filter and the adaptive codebook memory. The adaptive codebook output, $\mathbf{p}$, is obtained by a closed-loop codebook search procedure which selects the best delay and tap gains. The adaptive codebook search is discussed later in this section. The output of the synthesis filter, $\mathbf{y}$, is then compared to the actual speech signal, $\mathbf{x}$, and the best candidate codevector is selected using a perceptually weighted minimum MSE criterion based on the weighting filter, $W(z)$.

The optimal excitation shape and gain indices as well as the adaptive codebook delay and tap gains indices are then transmitted to the decoder. At the decoder side, the received indices are used to generate the proper excitation sequence. The gain scaled excitation codevector is then added to the output of the adaptive codebook, the resultant being fed into both the short-term synthesis filter and the adaptive codebook memory.

The 8 kb/s LLD-VXC system makes use of a 10-th order perceptual weighting filter identical to that used in the 16 kb/s LLD-VXC system [80]. The excitation gain adaptation makes use of a 10-th order backward adaptive linear predictor.

3.3.2 Short-Term Predictor

The output of the short-term predictor, $y(n)$, is computed using the relationship:

$$y(n) = w(n) + \sum_{i=1}^{m_p} a_i y(n-i) \tag{3.14}$$

where a_i are the coefficients of the all-pole section.

A comparison between block and recursive adaptation in the backward adaptive A-by-S configuration has shown that recursive adaptation results in a performance that is comparable to that of block adaptation, but with lower complexity [15]. In block adaptive systems, which traditionally use an all-pole short-term predictor, the autocorrelation method can be applied to adapt the short-term predictor coefficients by solving the Weiner-Hopf equations, and the resulting filter is always stable. Recursive adaptation on the other hand makes use of the gradient algorithm for solving

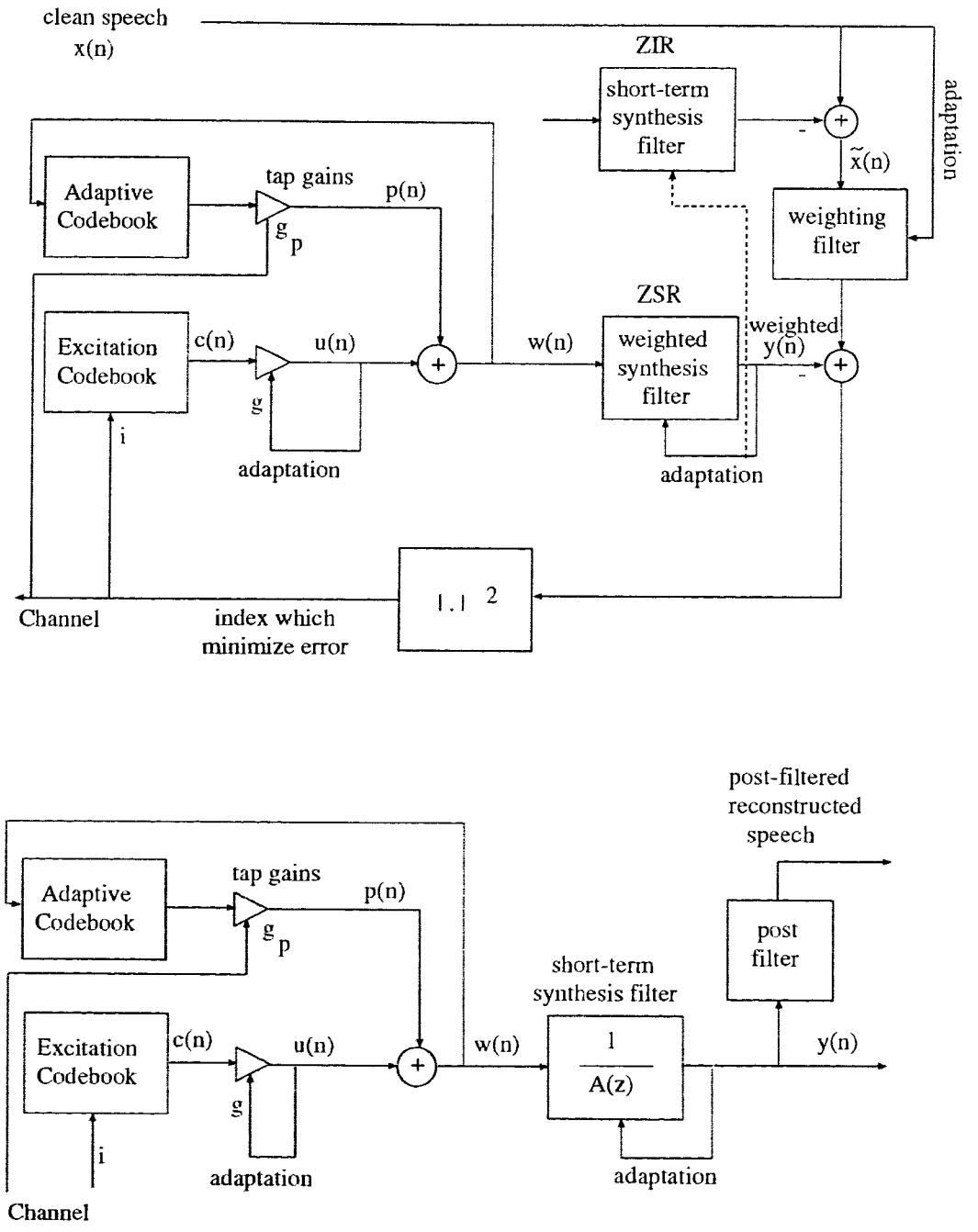


Figure 3.4: Partially-forward LLD-VXC configuration at 8 kb/s

the prediction error minimization problem. It now becomes necessary to check the stability of the all-pole filter at every stage in the adaptation. Simple analytical stability checks do not exist for filter orders greater than 3. However, by making use of a lattice implementation of the short-term predictor, it is easy to ensure the stability of the predictor by confining the magnitude of the reflection coefficients to be strictly less than unity.

In the LLD-VXC system, an all-zero lattice filter is used to obtain the coefficients of the perceptual weighting filter, and an all-pole lattice is used as the short-term synthesis filter. The lattice structures are shown in Figures 2.5 and 2.6 respectively.

In an all-zero lattice filter, the prediction error is updated by the recursions

$$e_j(n) = e_{j-1}(n) - k_j(n)r_{j-1}(n-1) \tag{3.15}$$

$$r_j(n) = r_{j-1}(n-1) - k_j(n)e_{j-1}(n) \tag{3.16}$$

where $k_j(n)$ are the jth-order reflection coefficients, e_j and r_j are the forward and the backward prediction errors. The input to the all-zero filter is the actual speech signal. The all-pole filter is just the inverse of the all-zero filter so the corresponding prediction error recursion equations are given by [87]

$$e_{j-1}(n) = e_j(n) + k_j(n)r_{j-1}(n-1) \tag{3.17}$$

$$r_j(n) = r_{j-1}(n-1) - k_j(n)e_{j-1}(n) \tag{3.18}$$

The input to the all-pole lattice filter is just the output of the pitch predictor.

The lattice reflection coefficients $k_j(n)$ are updated by making use of the least mean squares (LMS) algorithm with a leakage factor λ introduced to improve performance in noisy channel conditions. The adaptation equations are given by

$$k_j(n+1) = \frac{\alpha_j(n+1)}{\beta_j(n+1)} \tag{3.19}$$

$$\alpha_j(n+1) = \upsilon\alpha_j(n) + \lambda r_j(n)e_j(n) \tag{3.20}$$

$$\beta_j(n+1) = \upsilon\beta_j(n) + e_j^2(n) \tag{3.21}$$

The automatic gain control term, $\beta_j(n)$ is nothing more than an exponentially weighted sum of the squared prediction errors of the input process.

A significant reduction in the computational complexity can be obtained by using the so-called ***sign algorithm*** which avoids the division operations required by the

standard LMS algorithm [108]:

$$k_j(n+1) = v k_j(n) + \lambda \, \mathrm{sgn}(r_j(n)) \; \mathrm{sgn}(e_j(n)) \tag{3.22}$$

A good choice for the constants is $\lambda = 1 - v = 2^{-8}$.

3.3.3 Gain Predictor

The excitation gain adaptation scheme is essentially the same as the 16 kb/s LLD-VXC algorithm [80]. The fixed prediction coefficients a_{gp_i} have been optimized on a large training set for each vector dimension investigated [38]. The fixed predictors are more robust in the presence of transmission errors, but have slightly lower performance than the adaptive predictors for clean channel conditions.

3.3.4 Weighting Filter

The choice of the best excitation codeword is based on a perceptually weighted minimum MSE criterion defined by the weighting filter, $W(z)$. Perceptual weighting of the error signal is used to shape the noise spectrum in order to reduce the level of perceived noise.

To derive the weighting filter parameters, the input speech signal is fed into a 10-th order all-zero lattice filter and the coefficients are adapted by the algorithm described in Section 3.3.2. The reflection coefficients of the lattice predictor are then converted to their direct form transversal representation, $W_1(z)$. The transfer function of the weighting filter is given in terms of the optimal linear predictor for clean speech as

$$W(z) = \frac{W_1(z/\gamma_1)}{W_1(z/\gamma_2)} \tag{3.23}$$

Suitable values for γ_1 and γ_2 are 0.9 and 0.4 respectively [80]. The use of the original speech in deriving the weighting filter coefficients is acceptable because the decoder does not need any information about the weighting filter used in the encoder.

3.3.5 Long-Term Prediction

The pitch predictor equation for a three-tap predictor is given by

$$w(n) = u(n) + \sum_{i=-1}^{1} b_i w(n - k_p - i) \tag{3.24}$$

where b_i are the filter coefficients and k_p is the current pitch period estimate. Block or recursive adaptation may be used to obtain the predictor coefficients b_i.

In backward block adaptation, the pitch prediction coefficients, b_i, are calculated from the previous frame of reconstructed speech. However, in forward block adaptive systems, the coefficients, b_i, are obtained in closed-loop by comparing the synthesized speech with the original.

Backward recursive adaptation, which adapts the filter coefficients on a sample by sample basis, can be used in place of backward block adaptation and may result in better tracking of the changing signal statistics [15, 82, 14].

Backward Pitch Predictor

The fully backward LLD-VXC 8 kb/s system makes use of the hybrid backward adaptive pitch predictor [82, 14] which has already been presented in Section 3.2.3. The block adaptation is done every 160 samples. *i.e.* 20 ms for a sampling rate of 8 kHz.

Adaptive Codebook Search and Parameter Encoding

The previous pitch predictor adaptation algorithm fall into the category of open-loop algorithms. The partially-forward LLD-VXC system makes use of a forward closed-loop pitch predictor to improve the codec prediction gain. Closed loop implies here that instead of using a pre-determined pitch filter as in an open-loop scheme, the pitch parameters are included in the waveform matching process. The drawback of forward adaptation, is the necessity to increase the vector dimension of the coder to accomodate the extra bits required for pitch information in order to ensure that the bit rate of the resulting system remains unchanged.

A 4-bit delta pitch encoder can be used thereby saving 3 bits per vector when compared to a conventional 7-bit pitch quantizer [109, 13, 38]. This saving could translate to a reduction of the vector dimension by 3 samples since at a bit-rate of 8 kb/s there is available one coding bit per sample. However, the delta pitch encoder needs to have highly correlated pitch values in adjacent frames to be effective. This is difficult to achieve in a closed-loop search environment, as pitch doubling and tripling are very common. To solve this problem, an open-loop estimate of the pitch is obtained. This together with the previous frame's pitch is used in making the

decision whether to use a delta pitch encoder (locked mode) or the conventional pitch quantizer (unlocked mode) for the current frame [109, 13, 38]. The savings of 3 bits per vector for the locked mode translates to having 3 extra bits to be used in quantizing the excitation signal.

The 3 taps of the adaptive codebook were close-loop vector quantized using a tap-gain VQ (bit allocation is discussed in Section 3.3.7). The first step in the VQ search procedure is selecting a set of possible candidate pitch values. A search is then performed for obtaining the best possible entry in the tap-gain VQ for each pitch candidate previously selected. Finally, the best pitch candidate and tap-gain VQ index that minimize the distortion measure are selected. For the locked mode, the delta pitch encoder is used, and the pitch candidates are selected from 16 possible pitch values by performing a closed-loop search using the optimal gains. However, for the unlocked mode, the conventional 7-bit pitch quantizer is used and the pitch candidates are selected from 128 possible pitch values. The delay of the adaptive codebook is referred to as the pitch for simplicity though it may not necessarily be an estimate of the pitch. The design of the pitch predictor gain codebook is discussed in Section 3.3.6.

If a vector was classified as unvoiced, the mode flag was set to unlocked and a closed-loop search was performed over all possible pitch values. When a vector was first classified as voiced in a unvoiced to voiced transition, a locked/unlocked mode decision is made. Figure 3.5 illustrates the decision flowchart for the locked/unlocked mode. If the mode flag stays set at unlocked mode, the check is repeated for subsequent voiced vectors until the mode flag is set to locked mode.

3.3.6 Codebook Design

The optimization of the gain-shape codebook is an important consideration in coder design. The design problem may be defined as follows [80]: given the training sequence of input speech vectors $\{\mathbf{x}_n; n = 1, \ldots, N\}$ and the q-th shape and gain codebooks of size K and L respectively find the $(q+1)$-th ("new") shape and gain codebooks which will minimize the q-th average distortion

$$D(q) = \sum_{n=1}^{N} d(\tilde{\mathbf{t}}_n, \mathbf{y}_{\mathbf{wzsr}_n}(q)) \tag{3.25}$$

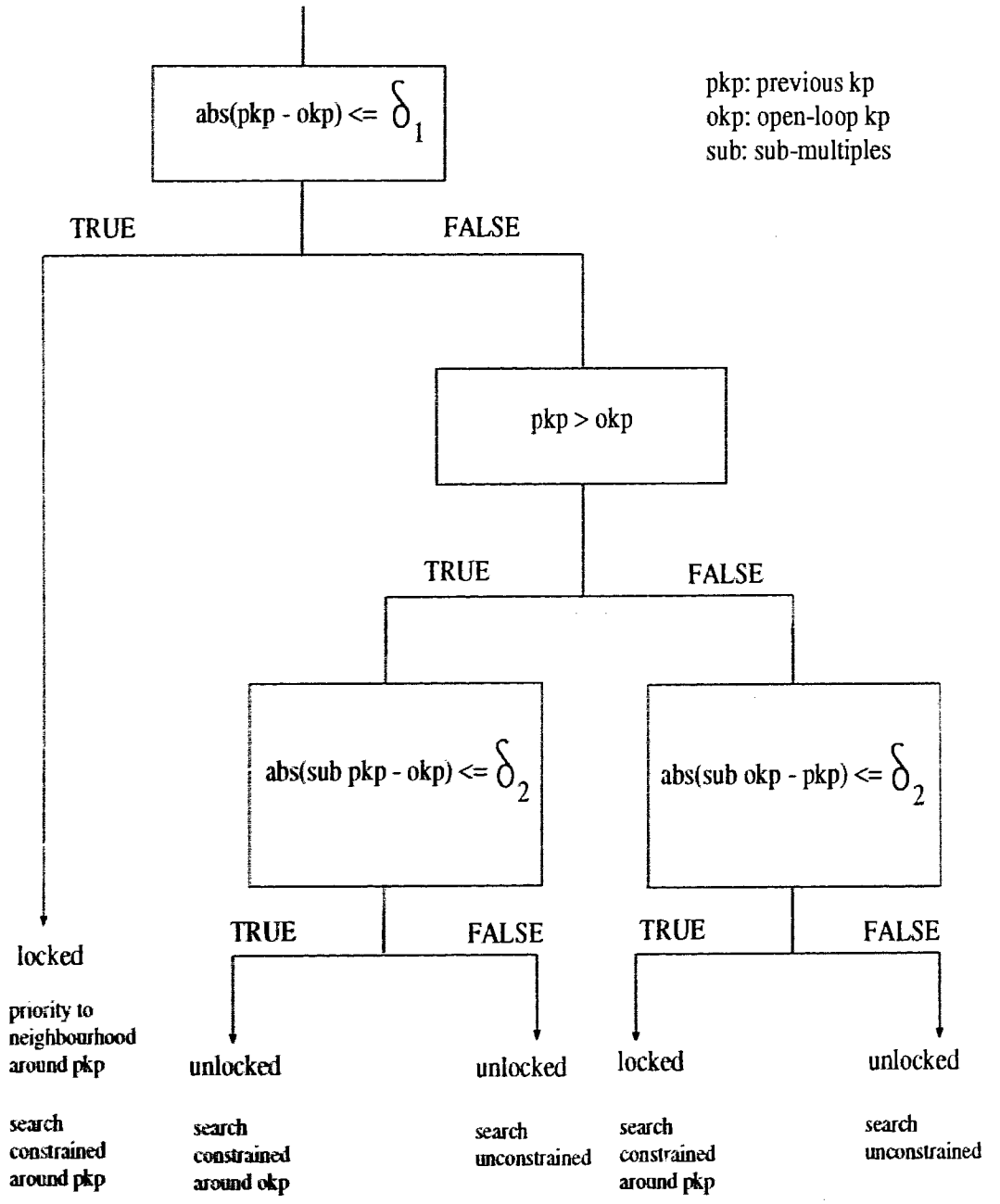


Figure 3.5: Locked/unlocked mode decision flowchart

where $\{\mathbf{y}_{\mathbf{wzsr}_n}(q); n = 1, ..., N\}$ is the q-th reconstruction vector and $\tilde{\mathbf{t}}_n$ is the target vector sequence.

The LLD-VXC system makes use of a ZSR-ZIR approach to reduce the complexity of the codebook search. In the fully-backward approach, the target vector $\tilde{\mathbf{t}}_n$ is defined as the difference between the weighted input speech vector and the ZIR response of the weighted short- and long-term predictors. For the partially-forward LLD-VXC system, the target vector is defined as the weighted input speech minus the ZIR of the weighted short-term filter minus the weighted ZSR response of the adaptive codebook excitation. The block diagram of the complexity reduced encoder for the fully-backward system is as in Figure 3.3. The block diagram of the complexity reduced encoder for the partially-forward system can be obtained from Figure 3.4 by considering seperately the adaptive codebook and stochastic codebook excitation contributions through the weighted short-term filter. For simplification, iteration index, q, will be suppressed in the following derivation.

The reconstruction vector, $\mathbf{y}_{\mathbf{wzsr}_n}$, is defined as

$$\mathbf{y}_{\mathbf{wzsr}_n} = G_n g_l \mathbf{H}_{w_n} \mathbf{u}_k \tag{3.26}$$

G_n is the predicted gain obtained from the backward adaptive gain predictor, $\mathbf{H}_{w_n}$ represents the zero state impulse response matrix, g_l is the l-th entry of the gain codebook, and $\mathbf{u}_k$ is the k-th entry of the shape codebook, where l and k are the indices found in the codebook search procedure. For simplification, index n has been suppressed for g_l and $\mathbf{u}_k$.

By making use of variational techiques the centroids of the shape and gain codebooks can be obtained by solving the following equations

$$(\sum_{n=S^i} G_n^2 g_l^2 \mathbf{H}_{w_n}^T \mathbf{H}_{w_n}) \mathbf{u}_i^{new} = \sum_{n=S^i} G_n g_l \mathbf{H}_{w_n}^T \tilde{\mathbf{t}}_n \tag{3.27}$$

$$(\sum_{n=G^j} G_n^2 \mathbf{u}_k^T \mathbf{H}_{w_n}^T \mathbf{H}_{w_n} \mathbf{u}_k) g_j^{new} = \sum_{n=G^j} G_n \tilde{\mathbf{t}}_n^T \mathbf{H}_{w_n}^T \mathbf{u}_k \tag{3.28}$$

S^i is the i-th cluster of the target vector, defined by making use of the nearest neighbour condition with the corresponding centroids given by $\mathbf{u}_i^{new}$. Similarly G^j is the j-th cluster of the target vector, with g_j^{new} as the corresponding gain centroid.

The design problem of the long-term predictor gain codebook is similar to that of the excitation shape-gain codebook discussed above [38], and can be defined as

follows: given the training sequence of input speech vectors $\{\mathbf{x}_n; n = 1, ..., N\}$ and the q-th long-term predictor gain codebook of size P find the $(q+1)$-th "new" gain codebook which will minimize the q-th average distortion

$$D(q) = \sum_{n=1}^{N} d(\mathbf{t}_n, \mathbf{Z}_n \mathbf{g}_p(q)) = \sum_{n=1}^{N} \|\mathbf{t}_n - \mathbf{Z}_n \mathbf{g}_p(q)\|^2 \tag{3.29}$$

where $\mathbf{g}_p$ is the p-th entry of the long-term predictor gain codebook, $\mathbf{t}_n$ is the target vector for the long-term predictor gain codebook search and matrix $\mathbf{Z}_n$ is the unscaled adaptive excitation passed through the ZSR of the weighted short-term filter. For simplification, index n has been suppressed for $\mathbf{g}_p$. The matrix $\mathbf{Z}_n$ is given by

$$\mathbf{Z}_n = \left[\begin{array}{ccc} \mathbf{z}_{k_p-1} & \mathbf{z}_{k_p} & \mathbf{z}_{k_p+1} \end{array} \right] \tag{3.30}$$

For simplification, index n has been suppressed for the $\{\mathbf{z}_{k_p+k}; k = -1, 0, 1\}$ vector terms, which are the adaptive codebook contributions at lag $k_p + k$, passed through the ZSR of the weighted short-term filter, and are given by

$$\mathbf{z}_{k_p} = \mathbf{H}_{wn} \mathbf{w}_{k_p} \tag{3.31}$$

By making use of variational techniques, the centroid of the long-term predictor (adaptive codebook) gain codebook can be obtained by solving

$$(\sum_{n=G^p} \mathbf{Z}_n^T \mathbf{Z}_n) \mathbf{g}_p^{new} = \sum_{n=G^p} \mathbf{Z}_n^T \mathbf{t}_n \tag{3.32}$$

where G^p is the p-th cluster of the target vector, with $\mathbf{g}_p^{new}$ as the corresponding gain centroid of the long-term predictor.

3.3.7 Initial Comparison of Systems

At 16 kb/s, in the presence of a pitch predictor, the short-term prediction gain saturates for a 20-th order predictor for male and female speakers [15, 80]. At 8 kb/s, there is no performance improvement for predictors of order larger than 10 [80]. The poor performance of high-order predictors at 8 kb/s may be caused by the quantization noise present in the adaptation loop.

In the backward adaptation coder, the short and long-term predictor parameters are obtained from the reconstructed signal. Hence, the only bits transmitted are those

representing the stochastic excitation. At a sampling rate of 8 kHz and a desired coding rate of 8 kb/s, we have available 1 bit per sample for quantization. It is desirable to have smaller vector dimensions as this will result in better quality if infinite bits are available. However, the actual number of bits available is equivalent to the vector dimension. Hence, the more bits available, the better will be the quantization of the stochastic excitation. This leads to a tradeoff between smaller vector dimensions and the number of bits required for effective quantization. It was found that vector dimensions of length 8 to 10 samples gave us the best qualitative tradeoff.

In the partially-forward coder, an adaptive codebook is used in place of the long-term predictor to obtain a larger prediction gain and subsequently better speech quality. In order for the adaptive codebook to give an improved performance, the adaptive codebook information is forward transmitted. This results in a larger vector dimension requirement to account for the extra bits needed to represent the adaptive excitation. One bit is needed to differentiate between the locked and unlocked mode. In locked mode, since a delta pitch encoder is used, 4 bits are sufficient to represent the pitch information. For the tap-gain quantization, it was found that 7 bits were required to obtain a satisfactory perceptual performance. For the unlocked mode, 7 bits are required to represent the pitch index information (pitch period is 20 to 147 samples). The tap-gains could be effectively quantized with less than 7 bits. The choice of the number of bits allocated to the adaptive codebook tap-gains and the stochastic excitation gain/shape codebook is based on achieving the best quantitative performance.

In order to compare the fully backward system with the partially-forward system, the following `LLD-VXC` systems were tested:

- System 1 - open-loop pitch predictor, vector dimension 8, 8-bit shape codebook
- System 2 - open-loop pitch predictor, vector dimension 10, 8-bit shape and 2-bit gain codebooks
- System 3 - forward adaptive codebook, vector dimension 22, bit allocation as shown in Table 3.1
- System 4 - forward adaptive codebook, vector dimension 24, bit allocation as shown in Table 3.2

bit allocation	flag	shape cbk	gain cdbk	pitch	tap gains
locked	1	8	2	4	7
unlocked	1	8	1	7	5

Table 3.1: Bit allocation for partially-forward system 3

bit allocation	flag	shape cbk	gain cdbk	pitch	tap gains
locked	1	8	4	4	7
unlocked	1	8	2	7	6

Table 3.2: Bit allocation for partially-forward system 4

All the above systems operate at 8 kb/s and use a 10-th order lattice short-term predictor and a fixed 10-th order gain predictor optimized for each vector dimension. It was found that, even though the coefficient optimization of the gain predictor did not provide any objective performance improvement, it did offer marginal perceptual improvement. The results of simulation tests are shown in Table 3.3 below.

System	SNR	Seg SNR	Mos Score
1	12.05	12.48	3.55
2	13.37	13.94	3.58
3	12.09	12.38	3.65
4	12.01	12.60	3.75

Table 3.3: Initial System Comparison Performance Results

The system employing open-loop pitch prediction has an informal MOS score around 3.6, while the closed-loop pitch prediction system has an informal MOS score of about 3.75. The results show that the partially-forward system has a performance similar to that of VSELP, while the backward system is slightly inferior. In comparison, LLD-VXC at 16 kb/s achieves a SegSNR of about 19 dB and a informal MOS close to 4.0. Systems 2 and 4 were chosen for some further studies on robustness under channel errors because they gave better objective and subjective quality evaluations in clean channel conditions when compared to systems 1 and 3 respectively.

3.3.8 Short-Term Predictor Adaptation

The 8 kb/s system makes use of a lattice structure for short-term prediction hence it is referred to as Lattice LD-VXC (LLD-VXC). For the 8 kb/s LLD-VXC system, the

short-term predictor adaptation is essentially the same as for the 16 kb/s LLD-VXC algorithm [80, 38]. The adaptation is based on the LMS algorithm, with a leakage factor introduced to improve performance in noisy channel conditions [87]. The leakage and exponential weighting factors are optimized so as to achieve robustness in noisy channel conditions without a noticeable degradation in quality for clean channel conditions.

In the 16 kb/s LLD-VXC system, the driving signal for coefficient adaptation is the reconstructed speech signal. In the 8 kb/s system, various driving signals were investigated to examine the effect of coefficient adaptation on system performance in clean and noisy channel conditions. The signals used are defined below (Figure 3.6).

- $y(n) \leftrightarrow$ reconstructed speech signal.
- $u(n) \leftrightarrow$ reconstructed excitation signal.
- $u_s(n) \leftrightarrow u(n)$ passed through a short-term shaping filter.
- $u_l(n) \leftrightarrow u(n)$ passed through a long-term shaping filter.
- $u_{ls}(n) \leftrightarrow u(n)$ passed through long and short-term shaping filters.

The short-term shaping filter is an finite impulse response (FIR) approximation of the short-term synthesis filter and is similar to the one described by Woo and Gibson in [103]. The long-term shaping filter, $B_{fir}(z)$, is obtained as a truncated FIR approximation of the long-term predictor. We denote by $B(z)$ the equivalent transfer function for the long-term predictor. Then, $B_{fir}(z)$ is given by

$$\frac{1}{1 - B(z)} \approx 1 + B_{fir}(z) \tag{3.33}$$

where

$$B_{fir}(z) = \sum_{i=1}^{m_{fir}} (B(z))^i \tag{3.34}$$

where m denotes the number of taps of the predictor $B(z)$ and m_{fir} is the order of the long-term shaping filter $B_{fir}(z)$. $m.m_{fir}$ is the number of non-zero terms in $B_{fir}(z)$

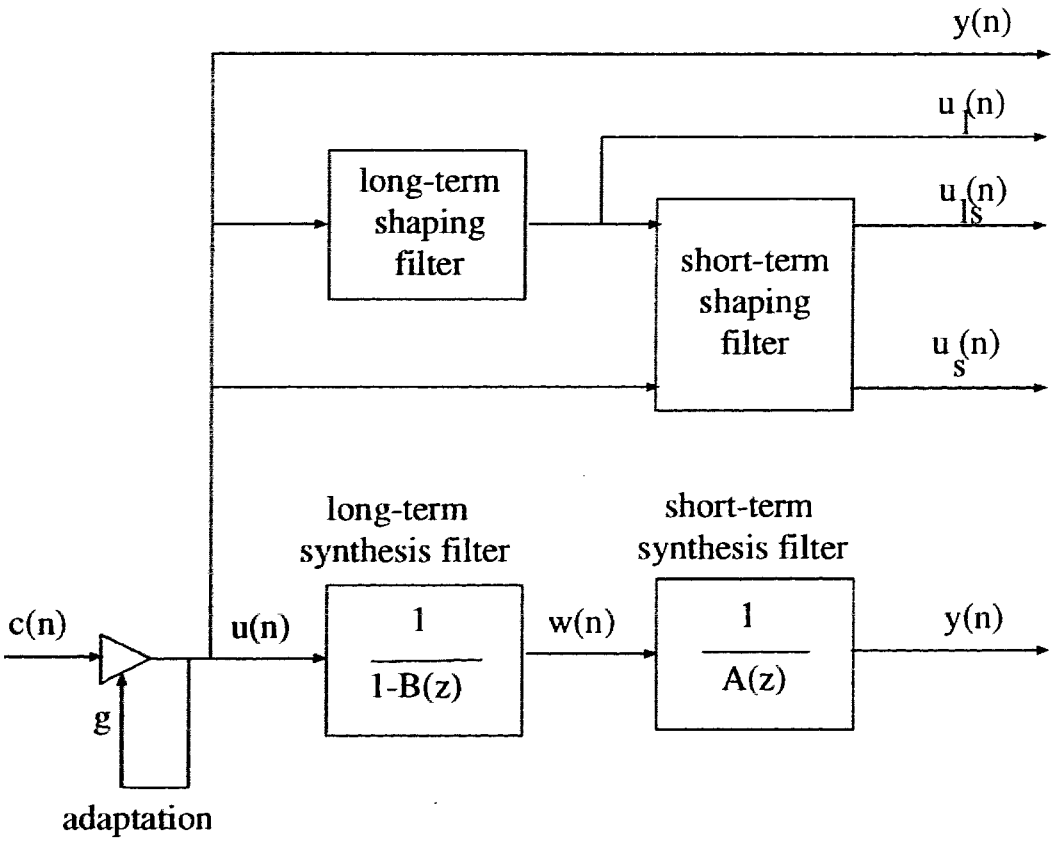


Figure 3.6: Lattice filter adaptation signals

for an m-tap predictor. Equation 3.34 can be obtained easily for a three-tap case ($m = 3$) as

$$B_{fir}(z) = \sum_{i=1}^{m_{fir}} (b_{-1}z^{-k_p+1} + b_0 z^{-k_p} + b_1 z^{-k_p-1})^i \tag{3.35}$$

Simulation results are shown in Table 3.4 for the backward system using various adaptation driving signals shown in Figure 3.6.

Signal	$y(n)$	$u(n)$	$u_l(n)$	$u_s(n)$	$u_{ls}(n)$
BER=0	12.95	12.12	12.53	12.72	12.94
BER=10-3	9.47	10.06	10.44	10.55	10.66

Table 3.4: SegSNR results for the backward system for various adaptation signals

The results in Table 3.4 show that using FIR approximations of the short- and long-term synthesis filters in the adaptation loop improves performance on noisy channels without any significant degradation in clean channel conditions. Particularly, the adaptation signal, $u_{ls}(n)$, achieves comparable performance to $y(n)$ in clean conditions and out-performs all other adaptation signals in noisy conditions. The backward system uses $u_{ls}(n)$ for adaptation of the short-term synthesis filter.

In the partially-forward system, $u_{ls}(n)$ provides a small performance improvement in noisy channel conditions. However, this improvement comes at the expense of a degradation in clean conditions which is roughly equivalent to the improvement in noisy conditions. As a result, the partially-forward system employs $y(n)$ for the adaptation of the short-term synthesis filter.

3.3.9 Robustness Results

A comparison of the fully backward open-loop system (now referred to as system 1) with the partially-forward closed-loop system (now referred to as system 2) leads to the results shown in Table 3.5 below.

Table 3.5 shows that the partially-forward system has better performance in clean channel conditions, while the backward system has better performance on a noisy channel. However, informal subjective listening tests show that the forward system performance at a bit error rate (BER) of 10^{-3} can be significantly improved by

System	BER=0	BER=10-3
1	12.94	10.66
2	14.09	6.10

Table 3.5: SegSNR results for the backward and partially-forward systems under bit errors

post-filtering. With post-filtering, the subjective performance of the partially-forward system is roughly equivalent to the performance of the backward system.

Informal subjective tests indicate that the partially-forward 8 kb/s system has quality comparable to the 8 kb/s VSELP standard in clean conditions, while the backward system is just slightly inferior. For noisy channels, at bit error rates of 10^{-3}, both systems achieve MOS scores which are within 0.2 on the MOS scale from the scores obtained in clean conditions. In the backward system, the use of the short-term adaptation signal $u_{ls}(n)$ results in a robust codec, which achieves good subjective quality even at BER as high as 10^{-2}. The partially-forward system degrades significantly at 10^{-2} mainly due to the errors on the forward transmitted tap gains.

Chapter 4

Packetized Speech Coding

This chapter is organized into five sections. An introduction into packetized speech coding is presented in Section 4.1. Some current and future system approaches in wireless/wired networks are presented while considering the need for a unifying network for the future. Section 4.2 provides an overview of packet networks with issues such as fast packet switching and a comparison of circuit and packet switching being presented briefly. In Section 4.3, some critical issues such as packet losses in networks and appropriate choice of packet duration are discussed with regards to possible applications and systems under which a packet recovery model may operate. Section 4.4 presents various system aspects of variable-rate coding with particular emphasis on flexibility of variable-rate coding in wired packet networks and existing and future wireless networks systems. The variable-rate CELP coder used in the chapter on packet recovery is briefly reviewed. Section 4.5 presents a literature review of earlier (and current) investigations into possible solutions to packet losses in speech coding systems.

4.1 Introduction

Currently there is great interest towards achieving a flexible, efficient and service independent network. The asynchronous transfer mode (ATM) approach provides a flexible means for dynamic bandwidth allocation and multi-media communications to support a wide range of broadband integrated services digital network (BISDN) services. Fast packet switching systems for T1 transmission and ATM for broadband fiber systems

have motivated interest in packetized speech transmission in wired networks. ATM based wireless networking systems are now also being actively considered to provide an untethered expansion to ATM based BISDN systems under development.

Third generation wireless networking will provide a wide range of wireless access applications. These systems in harmony with ATM based BISDN will use shared resources to convey many information types. Future public land mobile telecommunications systems (FPLMTS) is envisioned as such a wireless system. Service quality of wireless personal communication services (PCS) is expected to be the same as the wireline PSTN. However the applicability of current wired speech coding standards such as G.728 for wireless access remains questionable. Among the potential problems are system capacity, robustness to bit errors, and frame erasures due to bursty errors and packet losses.

An ATM approach for wireless networks has a few advantages such as: flexible bandwidth allocation; end-to-end provision of broadband services over wireless and wired networks; improved service reliability with packet switching techniques to name but a few. The wireless segment of the network requires additional medium access control (MAC) layers for channel sharing on radio links. Within the wireless segment, wireless specific protocol layers can be added to the ATM payload as required. To achieve a sufficient degree of transparency, two possible approaches have been investigated recently. Packet CDMA was developed for spread spectrum modulation and a packetized speech scheme called packet reservation multiple access (PRMA) was proposed for TDMA systems.

In the emerging communication networks, packet loss may result from:

- Discarding/loss of packets in a packetized network (*i.e.* ATM/BISDN or packetized FPLMTS)
- Loss of information as a result of bursty errors in non-packetized systems (*i.e.* second generation wireless and non-packetized third generation systems)

Although these two information loss sources are basically different, loss of information in bursty error channels has the same effect as loss of packets, and recovery techniques are very similar. With that in mind, a packet here could refer to an actual packet as in packet based systems or a block of samples in non-packet based systems.

4.2 Packet Networks

In this section, circuit and packet switching are briefly presented followed by a description into the evolution of fast packet switching with particular reference to asynchronous transfer mode (ATM).

4.2.1 Circuit vs Packet Switching

Prior to circuit switching, the first transfer mode used in the telecommunication world was some sort of "packet switching" such as in telegraphy. In telegraphy, a "packet" which was defined as a telegram, contained the address of the source and destination and the content of the message. The messages were coded more or less in a "digital" fashion using short and long pulses. Circuit switched networks were first introduced for use in telephone networks. In these networks which generally transmit voice, a private transmission path is established between any pair or group of users attempting to communicate and is held operative as long as transmission is required. The existence of a telephone set did not require any longer the coding of the signal in a "digital way", but the signal could be directly transmitted in an analog way.

Frequency division multiplexing (FDM) was developed to improve the efficiency of the telephone network by carrying multiple voice channels over the available bandwidth by dividing the band into a number of narrower bands. The introduction of digital technology into the telephone network was motivated by the need to improve the quality, add new features, and reduce the cost of conventional voice services. Time division multiplexing (TDM) was introduced to provide for greater efficiency in digital networks. In TDM one can assign time slots which are then time multiplexed by ensuring that a circuit will use the same time slot for the complete duration of a voice transmission.

In some applications which involve long distance connections, circuit switching may not be efficient enough due to periods of silence in voice transmission. Time assignment speech interpolation (TASI) was introduced to increase the capacity in multiplexed carrier systems by making use of voice activity detection. Degradations occurred due to clipping in moments of heavy talk spurt activity. TASI was subsequently replaced with digital transmission of voice using the same general idea of increasing capacity by exploiting the silent intervals in speech. Such schemes are

commonly known as digital speech interpolation (DSI) systems. In order to provide graceful degradation in moments of heavy talk spurts activity, embedded speech coding (see Section 4.4) is also employed to reduce the bit-rate, eliminating clipping inherent in analog TASI systems. Two multiplexing formats are used worldwide. The North American standard combines 24 voice channels into one time stream operating at 1.544 Mb/s. This system is also known as T1. The international standard used elsewhere multiplexes 30 voice channels for an aggregate bit rate of 2.048 Mb/s [90].

In packet switched technology, blocks of data called packets are transmitted from a source to a destination. Included in each packet is a header containing address and other control information. Each packet is transmitted as soon as the appropriate link is available with no holding of the link when a source has nothing to transmit. As the traffic load increases in a packet switched network, so also does the average transmission delay. In contrast, in circuit switched networks there is no graceful degradation in service, with a network either granting or rejecting service.

There are two modes of packet switched transmission: connection oriented and connection-less transmission. In connection oriented transmission, a path is first set up end-to-end through the network. User packets then traverse the network following the path chosen and arrive at the destination in the sequence in which they were transmitted. On the other hand, in connection-less transmission no initial connection is set up, with the routing information contained within the packet. Good sources of references on packet and circuit switching are [90, 4, 20]. A good description of routing within packet networks is presented in [4].

The time division multiplexing technique as described above is also known as synchronous TDM (STDM), since for the duration of a "call" a particular time slot is dedicated to a respective connection. Circuit switched telephone networks use STDM. Another form of multiplexing known as asynchronous TDM (ATDM) periodically changes the number of time slots within a frame. This is as a result of capacity being assigned only when it is needed. A time slot is eliminated (the frame shortened) when the respective source becomes inactive. This technique is similar to a packet switching system but usually transmits smaller blocks of data and does not include a header. In summary, a basic ATDM is strictly a multiplexer. A packet switching node however not only provides multiplexing-like functions, but also provides network level control functions as well [4].

Packet switching was introduced as a result of a need to utilize more efficiently a data communications circuit when compared to the existing circuit switched transmission techniques which were largely ineffective due to the burst characteristics of data traffic.

4.2.2 Evolution towards Fast Packet Switching

Today's telecommunication networks are characterized by specialization. This means that for every individual telecommunication service at least one network exists that transports this service. A few examples are: telex network, plain old telephone service (POTS), computer data transported through X.25 or X.21 protocols in the public domain and computer data transported through local area networks (LAN) in the private domain.

The networks of today suffer from disadvantages such as:

- Service dependence: Each network is only capable of transmitting a specific service for which it was designed. Even with the introduction of narrowband integrated services digital network (NISDN) in which voice and data are transported over a single medium, there will exist a limited service for some time, as packet switched networks and circuit switched networks will be dimensioned only for data or voice service.

- Inflexibility: Advances in audio, video and speech coding algorithms and progress in very large scale integration (VLSI) technology influence the bit-rate generated by a certain service and thus change the service requirements for the network. A specialized network has great difficulties in adapting to changing or new service requirements. As an example one can consider the evolution of speech coding standards from 64 kb/s PCM to rates as low as 6 kb/s in the ITU-T G.729 standard.

- Inefficiency: The internal available resources are used inefficiently. Resources which are available in one network cannot be made available to other networks.

It is consequently very important that the network of the future be service independent, flexible and efficient. The most flexible network in terms of bandwidth

requirements, and the most efficient in terms of resource usage, appears to be a network based on the concepts of packet switching. Recently, BISDN was introduced with the objective of solving the problems summarized above. This network will have the advantages of being flexible and efficient.

The definition of a service independent network has been influenced by an evolution in two key areas: technology and system concepts [20]. In recent years, a large technological progress has occured in semiconductor and optical technologies. This progress will allow new telecommunication networks to operate at much higher speeds. System concepts are introduced as a need to avoid repeating functions in the network if the required service can still be guaranteed when these functions are only implemented once at the boundary of the network.

Progress in System Concept

There are two key issues in system concepts: semantic transparency and time transparency. Semantic transparency is the function which guarantees the correct delivery at the destination of the bits which were transmitted by the source. Time transparency is the timely delivery of the bits that were transmitted. In the initial packet switched networks the quality of the transmission media was rather poor. In order to guarantee an acceptable end-to-end quality, error control was performed on every link. This error control is supported by the high-level data link control (HDLC) protocol, which includes functions such as frame delimiting, bit transparency, error checking, and error recovery (see [90] for more information on the HDLC protocol).

Figure 4.1 shows the different packet switched networks and the basis by which error control is done in the network. The layers correspond to the open systems interconnection (OSI) model as defined by the International Standards Organization (ISO). Figure 4.1 (a) shows the error control in initial packet switched networks.

With the advent of ISDN for narrowband services, the quality of the transmission and switching systems increased, thus reducing the errors within the network. In such high-quality networks, it is proposed to implement only the core functions of the HDLC protocol on a link-by-link basis, and other functions such as error recovery on an end-to-end basis. In Figure 4.1 (b), the OSI layer 2 is subdivided into two sublayers: layer 2a supporting the core functions and layer 2b supporting additional functions. Layer

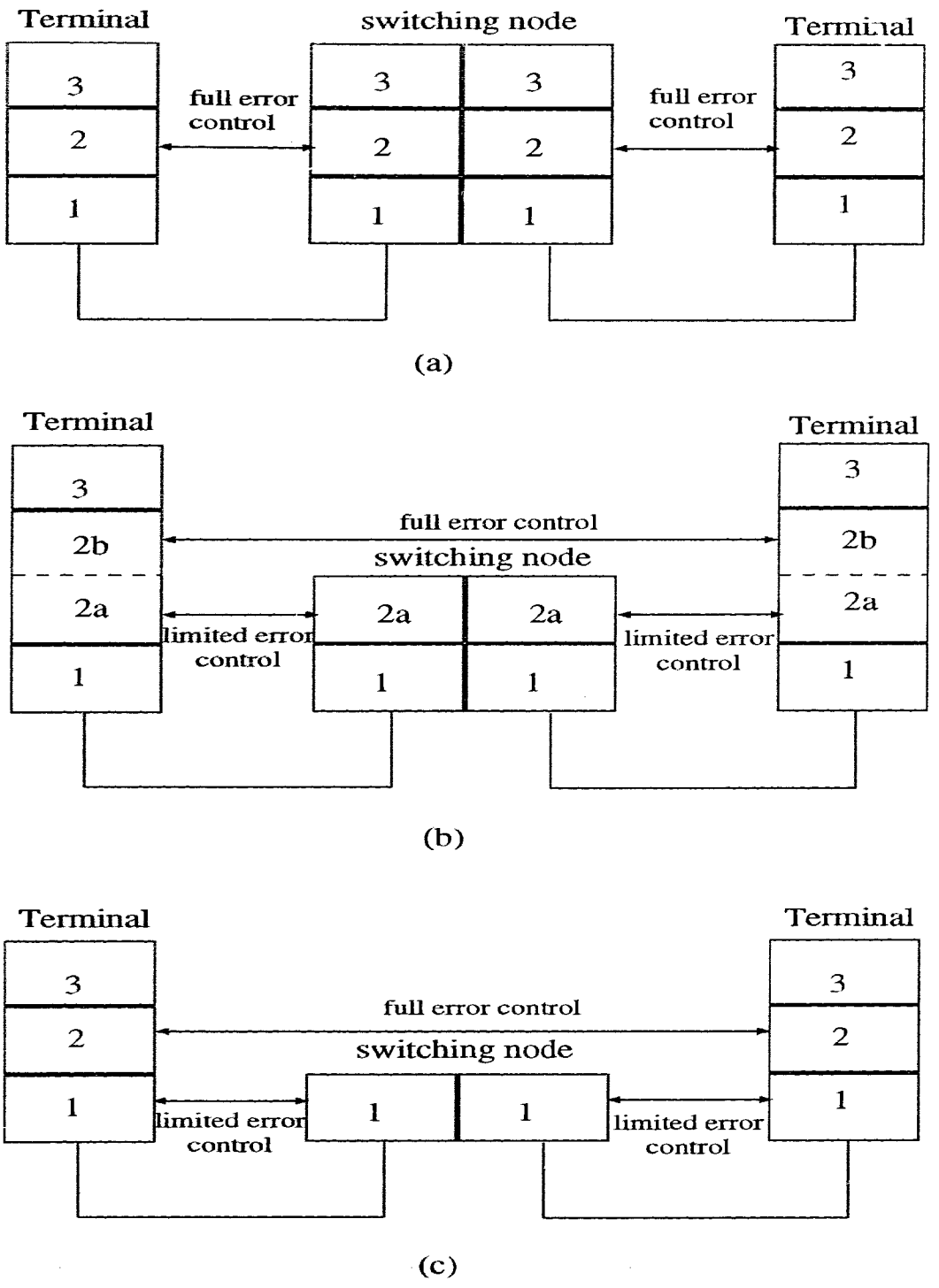


Figure 4.1: Error control in (a) earlier packet switching networks (b) frame-relay networks (c) ATM networks

2a operates on a link-by-link basis, layer 2b on an end-to-end basis. This concept is known as frame relaying.

This idea is further extended in BISDN. In this case, packets are still used, but the core functions of layer 2a have also been moved to the edge of the network (see Figure 4.1 (c)). This concept is used in ATM. In this case, error handling functions are no longer supported in the switching nodes inside the network. Due to a reduction in functions performed, there is a substantial decrease in switching node complexity which allows for higher speeds.

Some services such as voice and video require that the bit-stream arrive after a short delay at the other end. These services are called real time services. Packet switching and frame relaying to a lesser extent, have difficulties in supporting real time service. Because they require medium to large complexity at the switching nodes, they can only operate at medium-to-low speeds. This means that the delay will be quite large. ATM, on the other hand, needs minimal functionality and hence has a much smaller delay through the network, possibly ensuring time transparency.

Fast packet technology is characterized by: low end-to-end delay; high capacity switches; and wider bandwidth. The wider bandwidth gives rise to the term wideband packet technology. With the advent of fast packet technology, voice applications are now playing a more viable role in packet networks. Packetized voice has been achieved within fast packet technology (details are presented in [77]).

4.3 Issues and Applications of Packetized Speech

Before presenting packet recovery techniques, it is necessary to address some of the issues that exist in packet networks or inter-networks and bursty error channels in wireless communications. Although these two information loss sources are basically different, loss of information in bursty error channels has the same effect as loss of packets and recovery techniques are very similar. With that in mind, a packet here could refer to an actual packet as in packet based systems or a block of samples in non-packet based systems. In this section, some of the methods by which packets are lost in a network are presented. The choice of packet size needs to be addressed keeping in mind some of the applications and systems under which the packet recovery model could possibly operate.

4.3.1 Packet Losses in Coding Systems

Block erasures may appear in packetized systems due to network overload conditions. Speech blocks may be lost due to buffer overflow or they may be dropped in the network nodes by the congestion control mechanism. The congestion control operates as follows: newly arriving packets are blocked if the number already present at an input queue exceed some threshold. A distinction being made between input packets (packets newly arriving at a node) and transmit packets (those already in the network and arriving from another node). Another source of block erasures could be fixed delays due to transmission and propagation, as well as statistically varying delays such as queuing delay in nodes which result in packets not arriving at their destination in time and being considered as lost. Additional varying delays components are caused by packet retransmissions to compensate for errors in delivery. Alternatively, packets actually could be lost in traversing the network. For real-time voice communication, some reliability needs to be sacrificed by tolerating a small percent of lost packets [102].

In wireless networks, channel impairments can cause bursty errors. A portable radio channel exhibits very complex and variable behaviour. In such an environment, the direct path between any pair of transmitter and receiver is usually blocked by walls, ceilings and other objects. Different propagation paths are produced by reflections from various objects. Each path may have a different time delay and a different attenuation. Therefore, the portable radio channel experiences signal fluctuations caused by multi-path signal additions from different propagation paths. The fade and inter-fade durations depend on the velocity of the user, the carrier frequency, and the fade threshold. The effect of fading is to produce errors in bursts when the received signal envelope fades below some noise related threshold. Therefore, fading produces a channel that is either very good or very bad. For such channels, unless errors can be randomized by some means, forward error correction (FEC) is not worthwhile, since the channel is either very good and hence FEC is not required, or the channel is so bad that it cannot benefit from FEC. These error bursts directly result in packet losses if the speech information is packetized in wireless networks. Even in wireless systems which are not packetized, these error bursts directly result in loss of speech information. An excellent source of reference on the mobile communication channel is the text by Lee [63].

Codecs can be evaluated using a random errors channel model which is an acceptable model for wired networks. However, speech coders operating in a personal communications environment also need to address bursty errors in order to achieve acceptable performance in applications [21]. The random error assumption can still be applied if burst errors can be randomized by using techniques such as bit interleaving. The feasibility of bit interleaving depends on the fade duration and the tolerance of the system to the associated delay. For a given carrier frequency, fade threshold, and other system parameters, the fade and inter-fade durations, on the average, are inversely proportional to the velocity of the user. Therefore, the faster a user moves, the shorter the fade and inter-fade durations experienced by the user will be. Hence in the vehicular mobile environment, bit interleaving to randomize bursty errors may be achieved within acceptable delay. The random error assumption may also be meaningful to model spread spectrum systems (*e.g.*, CDMA), as the inherent frequency diversity helps in reducing the long-time autocorrelation properties of the signal envelope. On the other hand, a PCS channel, especially when using TDMA, does not experience random errors. Such channels, where the user terminal is moving slowly or not at all, experience slow fading. The fade duration could be as much as several hundred milliseconds long. For instance at 850 MHz and a speed of 12 mph, the signal goes into a -15 dB fade at the rate of approximately 6 times a second. The probability that the duration of this fade is 8 ms or more is about 0.2 [53]. Because of the slow fading nature of the channel, a substantial amount of bit interleaving would be required to randomize errors and this results in unacceptable delay.

4.3.2 Packet Size Considerations

Packet size is an important design criteria for the packet recovey tests. An interesting tradeoff is packet size with respect to the average interfade interval. The average interfade interval is the duration between fades. If a packet size is too large and is comparable to the average interfade interval, it is quite likely that the packet will be corrupted by a fade. On the other hand, if the packet size is too small, multiple packets may be embedded in the fade. In either case, a number of retransmissions would be needed which will cause delays. In order to minimize both the packetization delay at the transmitter and the perceptual effect of lost packet anomalies at the receiver,

packets should be as short as possible. On the other hand, in order to maintain high channel utilization, we would like to keep the number of speech bits per packet as high as possible relative to the overhead which must accompany each packet. This tradeoff becomes more difficult for lower rate speech. For higher speech rates, relative packet overhead is less of a problem. The choice of packet size is also influenced by limitations on network throughput in packets/s. For the same user data rate, processing loads on network nodes will generally increase as packet size is decreased. This warrants use of larger size packets.

Experiments performed on PCM systems by Goodman *et al.* showed that the choice of packet size had an effect on perceived quality of the reconstructed speech [33]. With small packets of the order of 1-2 ms, with 10% packets missing, there is a constant, annoying crackle. For very large packets (32 ms or more), the speech quality degrades significantly. A system with a packet size of 8-16 ms was found to be most tolerant to packet losses. Jayant *et al.* had earlier reported packet sizes of the order of 16-32 ms for greatest tolerance to packet losses [48]. As speech can be assumed stationary for atmost 30 ms, packet sizes are upper limited by this figure to enable effective performance under packet losses. Therefore, suitable packet size durations of the order of 10-20 ms were chosen for the packet recovery experiments in this thesis.

4.3.3 Packetized speech in ATM/BISDN

The ATM approach provides a flexible means for dynamic bandwidth and multi-media communications to support a wide range of BISDN services. Fast packet switching systems for T1 transmission and ATM for broadband fiber systems have motivated interest in packetized speech transmission in wired networks. ATM, which is similar to packet multiplexing, is implemented by dividing the information flow into short entities and by sending independently these entities to their destination. One drawback in packet transmission is the possibility that some packets are not received within the maximum allowed delay or are lost due to network overload. When this condition occurs, one or more packets are discarded, yielding the received speech signal to be degraded because of the missing information. Minzer presented a good review paper on BISDN/ATM as an all purpose digital network with a discussion into technical issues that impacted ATM specifications [75].

The ATM cell size is 53 bytes (5 bytes header, 48 bytes payload). Therefore, the information part of the cell is 384 bits. Since ATM can support many different traffic types, voice can be transported over the broadband network in a variety of ways. Three methods by which voice can be transported in an ATM-based broadband network are:

1. PCM voice serviced via constant bit rate at 64 kb/s.

2. Compressed voice using reduced coding rate and possibly variable-rate features and cell discarding to reduce congestion.

3. Compressed voice already packetized using an existing protocol such as packetized voice protocol (PVP) could be used. However, some inefficiency can result as PVP voice packets are transformed into ATM cells.

The facility of operating at variable bit-rates is particularly attractive considering the scenario of future telecommunication networks based on ATM [78].

Packetized Voice Protocol

Recommendation G.764 defines a packetized voice protocol (PVP) for speech packetization in permanent virtual circuit applications [7]. The principal application of the PVP is at the primary rate (1.544 Mb/s or 2.048 Mb/s) and in fractional primary rate applications.

The stream of coded speech is transformed into packets by collecting samples over a period of 16 ms and dividing into blocks of 128 bits each. For example, in ADPCM at 32 kb/s, a packet would consist of 512 bits (*i.e.* 4 blocks). A block consists of bits of the same significance collected from all speech samples that are packetized. The blocks are arranged according to the significance of the bits. A feature of the PVP is the ability to drop blocks from a packet as a congestion control mechanism. This differs from cell discarding, which may be used in ATM networks, where cells are categorized as more significant and less significant, with less significant cells being discarded during congestion.

The G.764 assumes use of PCM, ADPCM and embedded ADPCM for encoding of speech. It has a provision through reserved coding types, as other voice coding algorithms are standardized.

4.3.4 Wireless Personal Communication Service

Mobile networks provide communications between moving vehicles and PSTN customers. With these networks possibly being partially replaced by ATM networks, customers will probably accept a variety of quality grades for services such as mobile communications. Next generation wireless networks will be required to co-exist with wired broadband communication networks such as BISDN. In order to avoid mismatch between future wireline and wireless networks, timely consideration of broadband wireless systems with similar service capabilities has begun [86, 32]. Third generation wireless networking will provide a wide range of wireless access applications. These systems in harmony with ATM based BISDN will use shared resources to convey many information types. Future public land mobile telecommunications systems (FPLMTS) is envisioned as such a system.

An ATM approach for wireless networks has a few advantages such as: flexible bandwidth allocation; end-to-end provisioning for broadband services over wireless and wired networks; and improved service reliability with packet switching techniques. The wireless segment of the network requires additional medium access control (MAC) layers for channel sharing on radio links. Within the wireless segment, wireless specific protocol layers can be added to the ATM payload as required.

Variable-rate coding techniques provided a viable role in second generation mobile systems. The CDMA approach proposed by Qualcomm was chosen as the basis for a new digital cellular standard which provided for an increased system capacity over the IS-54 standard [99]. Variable-rate coding allows the CDMA system to have a "soft capacity". DSI has been applied to TDMA systems in enhanced TDMA [29]. Both these MAC approaches in the forms above, are insufficient to meet a good degree of transparency with wired ATM systems. To achieve a sufficient degree of transparency, two possible approaches have been investigated recently.

Packet CDMA was developed for spread spectrum modulation (some detail is presented in [86]). When narrow band modulation is used at the physical level, some form of TDMA based MAC is typical for PCS systems. A packetized speech scheme called packet reservation multiple access (PRMA) was proposed for TDMA systems by Goodman [32]. PRMA was originally conceived for local wireless networks, which dynamically assigns a sequence of fixed-length packets corresponding to one talk spurt to a TDMA

slot. At the start of the talk spurt, the terminal contends for an available slot. Once assigned, the slot is reserved for the duration of a talk spurt. When the talk spurt has ended, the slot is released. The contention in PRMA is similar to that of slotted ALOHA. PRMA can be viewed as a combination of TDMA and slotted ALOHA.

Service quality of wireless PCS is expected to be the same as the wireline PSTN. However the applicability of current wired standards such as G.728 for wireless access remains questionable. Among the potential problems are system capacity, robustness to bit errors and frame erasures/packet losses. A review of ITU-T liaison statements (see [105, 106, 107]) is presented now to give a clearer perspective on issues such as packet size and loss models that need to be considered in FPLMTS.

At the November 1991 meeting of ITU-T working party (WP) XV/2 and the Ad Hoc meeting on 8 kb/s speech coding, an agreement to consider the use of 8 kb/s speech coding for the FPLMTS was introduced. WP XV/2 requested the ITU-R (radio communications sector of the ITU, formerly known as CCIR) to provide a model of radio-channel characteristics to be used in examining codec performance under burst errors [105]. BellCore provided ITU-R with such a model which employed a 16 ms block length based on the frame specifications as originally required for 8 kb/s speech coding [97].

At the November 1992 meeting of ITU-T WP XV/2, it was noted that ITU-R had requested WP XV/2 to consider the use of 16 kb/s LD-CELP in place of the future 8 kb/s coder in FPLMTS as a matter of the highest importance [106]. This was brought upon by a need to consider using the 16 kb/s LD-CELP algorithm in the early phase of PCS. The burst error model would need to accommodate the 16 kb/s LD-CELP frame length of 2.5 ms as well as the modified 10 ms frame length requirement of the 8 kb/s coder [106].

The most critical modification that was considered for the LD-CELP algorithm was to enhance the performance under block erasure conditions. The objectives as set out by the ITU-T/R were, less than 0.5 MOS degradation for block erasure rates of 3% when compared to G.728 under clean channel conditions [107].

4.4 Variable Rate Coding

For digital transmission over a fixed-rate channel, a constant bit-rate data stream at the output of a speech encoder is usually needed. However, for some applications in telecommunications such as ATM based BISDN systems and wireless networks, a variable bit-rate output is particularly advantageous. Variable bit-rate speech coders can exploit the pauses and silent intervals which occur in conversational speech and may also be designed to take advantage of the fact that different speech segments may be encoded at different rates while maintaining a given reproduction quality. Consequently, the average bit-rate for a given reproduced speech quality can be substantially reduced if the rate is allowed to vary with time [27].

Recent applications that have motivated the study of variable-rate coding include packetized voice operating in an ATM based BISDN system as well as ATM based future wireless networks. Multiple access schemes for wireless communication, particularly CDMA systems have been an important application for variable bit-rate coding. Recently, PRMA and packet CDMA have become an important application for variable-rate coding within wireless networks.

4.4.1 Overview

Variable-rate coders can be divided into two main categories [29]:

1. source-controlled variable-rate coders: where the coding algorithm responds to the time-varying local character of the speech signal to determine the data rate.

2. network-controlled variable-rate coders: where the coder responds to an external control signal to switch the data rate to one of a predetermined set of alternative rates. The external control signal is generated typically in response to traffic levels in the network rather than the short-term character of the speech.

   Network controlled coders can be further subdivided into two categories:

   (a) multi-mode variable-rate coders: where a different mode of encoding or perhaps an entirely distinct coding algorithm is performed for each bit-rate option

(b) embedded coders: where a single coding algorithm generates a fixed-rate data stream from which one of several reduced rate data signals can be extracted by a simple bit-dropping procedure. Thus, each lower rate data signal is embedded in the full-rate bit-stream. Embedded coders can be considered as a special case of multi-mode coders.

Generally, it is much simpler to design a variable-rate coder whose rate is externally controlled rather than source controlled. The simplest way is to design a family of fixed-rate coders each with a different rate and simply switch to the appropriate coder to produce the currently needed rate (multi-mode coder). An important consideration is the frequency with which the rate is to be switched. For frequent switching, a smooth transition may require preserving the context or state of the old coder for use in initializing the new coder. In such cases, each coder is likely to use the same algorithm but with modified bit allocations. For infrequent switching it is possible to have entirely different coders.

4.4.2 Variable-Rate 3 kb/s CELP

Speech Classification and VAD

An important component in source controlled variable-rate speech coding is voice activity detection (VAD) which is needed to distinguish active speech segments from pauses, when the speaker is silent and only background acoustical noise is present. An effective VAD algorithm is critical for achieving low average rate without degrading speech quality in variable-rate coders [27]. When silence is detected as speech due to VAD errors, the capacity is reduced: on the other hand, when speech is detected as silence, degradations in the recovered speech quality are introduced. For the mobile environment, the design of a VAD is complicated by the high level of acoustic noise coming to the microphone. To avoid degrading the speech quality, the VAD algorithm may be designed conservatively so that a lot of background noise will be classified as active speech rather than silence, thus reducing the potential capacity gain.

Although voice activity patterns offer an important and essential component for source-controlled variable-rate coding, even during active talk spurts the speech signal has a time varying short-term entropy. In other words, variable-rate coding of active

speech segments is a natural way to achieve further reductions in average bit-rate for a given reproduction quality. Some of the variable-rate coding techniques are briefly reviewed here, however, Gersho and Paksoy provide a more complete overview of variable-rate coding [29].

A speech coder based on variable-rate phonetic segmentation which makes use of a good technique for VAD to better identify speech in low SNR environment is presented in [79]. The VAD scheme is similar to the VAD introduced in the GSM TDMA full-rate standard, with some added features such as energy level comparisons in individual frequency sub-bands, measurement of the spectral flatness of the signal at the output of the noise suppression filter, and a variable hangover time.

QCELP which was chosen as part of the CDMA system proposal to the TIA for a wideband digital cellular standard makes use of a form of variable-rate coding, where the coder selects one of four rates for each frame by comparing the energy level of the frame with a set of three adaptive threshold levels [25].

The variable-rate coder which is used in this thesis will be briefly reviewed in the next subsection and is an alternative to variable-rate techniques discussed in this subsection.

System Overview

The variable-rate coder used in the packet recovery experiments in Chapter 5 is based on a coder presented in [110]. The variable-rate coder has a peak rate of 8.8 kb/s. The system switches between three distinct codec configurations: 8.8 kb/s for voiced and transition frames, 3.9 kb/s for unvoiced frames and 750 b/s for silence frames, with an overall average bit-rate of about 4-5 kb/s. The appropriate coder configuration is selected by specifying the allowed ranges for the shape and adaptive codebook indices (indicated by control signals).

The block diagram of the variable-rate CELP (VR-CELP) coder is shown in Figure 4.2. The bit allocation of the variable-rate system is shown in Table 4.1. The details of the VR-CELP coder are presented in [110].

The frame classifier is based on thresholding. The threshold approach analyzes the speech on a fixed frame basis. One or more parameters are derived from the speech source and a class decision is then made.

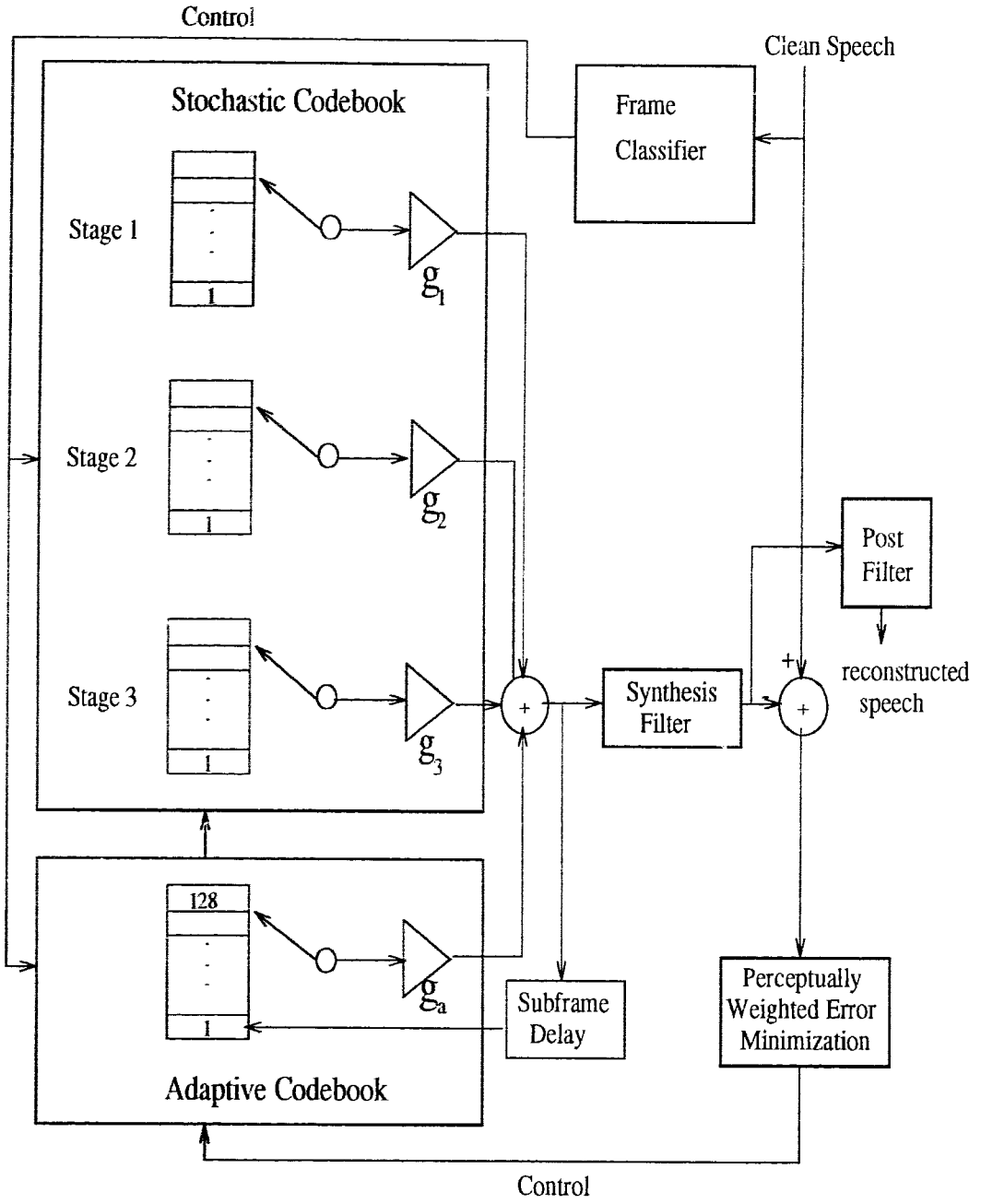


Figure 4.2: Block diagram of variable-rate CELP coder

Parameter	S	UV	V/TR
Frame Size	160	160	160
Sub-frames	1	4	4
short-term	6	24	24
RMS gain	4	4	6
ACB index	-	-	7x4
ACB gain	-	-	8x4
SCB index	-	8x4	5x3x4
SCB gain	-	4x4	6x4
classification	2	2	2
Total Bits	12	78	176
Bit-rate	750	3900	8800

Table 4.1: Bit allocation for VR-CELP coder

The synthesis filter is a tenth order LPC all-pole filter. The LPC coefficients are computed once per frame and converted to LSPs for quantization and interpolation. The LSPs are linearly interpolated every subframe and converted back to LPCs to update the synthesis filter. A vector quantization technique is used to code the LSPs.

For the voiced and transition classes, the excitation is obtained from an adaptive codebook and a stochastic codebook. The pitch index is directly coded using a 7-bit quantizer. Unvoiced speech has little periodicity, hence the adaptive codebook can be omitted, resulting in a substantial reduction in bit-rate at the expense of a slight reduction in quality. The variable-rate system makes use of multi-stage excitation codebooks. Multi-stage VQs were mentioned in Chapter 2.

Even though a frame classified as silence may not contain active speech, it is still necessary to reproduce the background noise to preserve the naturalness of the reconstructed speech. The LPC coefficients are still computed, but are quantized using only 6 bits. Both the adaptive and stochastic codebooks are omitted resulting in a substantial reduction in the bit-rate.

4.5 Literature Review

The study of packet losses in speech coding can be traced back to the work of Jayant *et al.* [48] who studied odd-even sample interpolation in 12-bit (uniform quantization) pulse coded modulation (PCM) and 4-bit differential pulse coded modulation (DPCM)

systems. Results of these studies were compared with a system in which missing samples were replaced by zero amplitude values. Jayant *et al.* reported packet sizes of the order of 16-32 ms for greatest tolerance to packet losses. With these packet lengths, acceptable quality can be obtained with loss probabilities as high as 2 to 5 percent without interpolation and 5 to 10 percent with interpolation [48].

Weinstein presented a good review of the possible use of speech communications in packet networks, including a series of experiments conducted under the sponsorship of the United States Defence Advanced Research Projects Agency (DARPA) [102]. A generic packet speech system is described as a point of reference. The experimental system configuration and key results for packet speech on the ARPANET, SATNET and wideband system are then presented. The reconstitution procedures used to decide what to play out when a packet is not received, include filling with silence, filling by repeating the last segment of speech data and filling with repeated frames of speech data which are made voiceless and have energy values which decay with time. Because of limited network bit rates, the speech coding algorithms used in these tests were mostly based on the LPC vocoder at 2.4 kb/s. However, for the wideband tests, PCM at 64 kb/s was also accomodated to allow for inter-operation with digital circuit switched systems. Such inter-operation would be essential in introducing packet speech into an environment dominated by circuit switched voice users.

Goodman *et al.* reviewed waveform substitution techniques such as pattern matching and pitch detection in PCM systems [33]. Pattern matching is based on scanning a search window to find the samples of the missing packet. This is done by selecting a template of speech samples that came just before the missing packet and then searching for a best match from the samples in the search window. It then uses the samples that follow the best match in reconstituting the missing samples. In the pitch detection approach, a missing packet can be reconstructed by repeating samples at the pitch period. Experiments performed by Goodman *et al.* showed that the choice of packet size had an effect on perceived quality of the reconstructed speech [33]. With small packets of the order of 1-2 ms, with 10% packets missing, there is a constant, annoying crackle. For very large packets (32 ms or more), the speech quality degrades significantly. A system with a packet size of 8-16 ms was found to be most tolerant to packet losses.

Erdol *et al.* [22] presented more recent work in waveform substitution based on interpolative techniques on short-time energy parameters in PCM systems, whereby the short-time energy parameters and zero crossing information is transmitted in the preceding packet. Hence, when a packet is lost, its envelope and frequency characteristics are obtained from a previous packet and used to synthesize a substitute waveform. Experiments showed that intelligible speech can be obtained with packet loss rates of up to 40%. They reported excellent speech quality at a loss rate of 10% and fair speech quality at 40%. This corresponds to an MOS of 4.5 and 3.0 respectively. The performance of this method is not significantly affected by the characteristics of the segment missing. The overhead in bits was quite significant however. The method requires the allocation of one bit per sample to the zero crossing information, in addition, significant number of bits are needed to represent the energy information. In the event of two or more consecutive lost packets, zero stuffing is used in place of the procedure described above.

Erhart and Gibson presented results on making use of a tree searching technique in interpolative recovery for DPCM coders under frame losses [23]. Marvasti reviewed various interpolation techniques to be used in voice packets to improve the SNR under packet losses [70].

Embedded coding provided a strong basis for significant work on packet recovery techniques by providing a fail-safe mechanism for packet losses at the expense of a slight degradation under clean conditions. However, embedded coders imply significant change in encoder design, resulting in inter-operability constraints for coders already deployed. Embedded coders can have supplementary and essential bits placed in different packets with supplementary packets being discarded in overload conditions. Suzuki and Taka presented a theoretical analysis of the performance expected for PCM systems for different recovery schemes which include sample interpolation techniques and least significant bits dropping. Suzuki and Taka also presented a detailed study of embedded coding with least significant bit dropping in 32 kb/s ADPCM [95]. Embedded coding applied to ADPCM ensures that there is no mistracking between the encoder and the decoder when least significant bit packets are lost. Also, the performance of the system with embedded coding and least significant bit dropping is superior to the system with just least significant bit dropping under packet losses.

Work on packet-based embedded coding of 32 kb/s ADPCM also includes the work

of Lara-Barron and Lockhart [61]. In their scheme, whole packets deemed less significant to speech quality, are simply discarded to achieve a reduction in bit-rate in the network. At the transmitter, a packet predictor attempts to predict the current packet on the basis of previously encoded speech. If the prediction is deemed successful, the packet is labelled as supplementary (*i.e.* less significant to speech quality), but otherwise it is labelled as essential. At the receiver, when packets are missing, the predicted packets are used in place of the missing packet. Predicted packets are used to update both the encoder and decoder parameters for supplementary packets to ensure no mistracking of parameters.

Extensive literature on ATM networks abound but there are relatively very few papers that discuss the specific issues of speech coding and missing packets. An attempt has been made in this thesis to review some of the papers specific to research into packet losses in the speech coding environment. Minzer presented a review paper on BISDN/ATM as an all purpose digital network with a discussion into technical issues that impacted ATM specifications [75]. Kitawaki *et al.* presented feasible applications of ATM technology in telecommunication networks such as ISDN, PSTN and mobile networks. Kitawaki *et al.* also examined the effect of packet losses on performance of 64 kb/s PCM and 32 kb/s ADPCM (both embedded and standard) [55]. A speech coding technique for ATM networks was presented by Nakada and Sato [78]. A block coding scheme is employed, which is based on a variable-rate coding algorithm that makes use of pitch prediction. A variable-rate coding mechanism exists to give a rate independent capability in ATM networks. The proposed algorithm exhibits better quality than that of 32 kb/s ADPCM at a mean bit-rate of less than 13 kb/s.

Variable-rate embedded ADPCM for ATM networks is presented in [58]. In ATM networks, as speech is packetized into fixed-length cells, the contents of a cell cannot be discarded selectively as is the case for variable-length cells. This means that the essential and supplementary bits are packed into seperate cells with discarding of the supplementary cells primarily being considered. Embedded ADPCM does not maintain sufficient quality when directly applied to ATM. The performance is enhanced by making use of a variable-rate embedded coder to achieve superior performance in high supplementary cell loss conditions.

A good review paper on voice compression and cell discarding in ATM networks is the paper by Sriram *et al.* [92]. The advantages of voice compression and cell

discarding in broadband ATM networks are demonstrated in terms of transmission bandwidth savings and resiliency of the network during congestion. Compression is achieved by a 32 kb/s embedded ADPCM coder with DSI, while congestion control is achieved by cell discarding, where cells containing low-priority voice samples are discarded during congestion.

Leung *et al.* considered vector linear prediction in voiced frame reconstruction in an embedded 16 kb/s CELP/multi-pulse coder over frame relay networks. The 16 kb/s coder is based on an embedded 8 kb/s CELP coder with the supplementary information being specified by a multi-pulse algorithm. The embedded CELP coder has minor degradation in quality at a loss rate of 2% [64].

Su and Mermelstein presented error detection and speech regeneration techniques that were compatible with VSELP and improve subjective quality. Error detection is enhanced by detecting frames that are beyond the error correction capability of the correction code used. Rejected frames are generated using the most recent correctly received speech coding parameters [94]. Gerlach combined error detection and speech extrapolation in a probabilistic framework ,by computing *a posteriori* probability of coder parameters if the previous parameters were received correctly, and then developing optimum estimators adapted to human perception [26].

Goodman presented work on cellular packet voice communications for possible use in third generation wireless networks by combining cellular packet switching with PRMA and investigating system capacity as a result of dropped packets [32]. In cellular packet switching, base stations and public network trunks are placed on a high speed metropolitan area network, enabling better management of functions currently performed by a single mobile telephone switch. Information enters and leaves the MAN through cellular interface units. The packet switching capability of a MAN works fittingly with the PRMA protocol for information transfer between base stations and wireless terminals. As PRMA is a mobile-to-base transmission protocol, there is now a need for a wireless interface unit that delivers packets to the radio transmitter.

Raychaudhuri and Wilson proposed an ATM based approach, as the basis for the next generation wireless network that would ease interfacing with proposed ATM based wired BISDN systems [86]. Issues related to the physical, MAC and data link layers of the ATM-based radio link are discussed. Several factors tend to favour the use of ATMs in wireless networks such as: end-to-end provisions for broadband services over

wireline and wireless networks; and improved service reliability with packet switching techniques. A 48-byte ATM cell (or a suitable integer submultiple) would be the basic unit for data transmission in the wireless network. In the wireless network, wireless channel specific layers would be added to the ATM payload as required and replaced by ATM headers before entering the wireline network.

Karim presented a packetized voice technique for mobile radio applications, making use of error detection to handle for error bursts [53]. If the receiver detects an error, it throws away the packet and requests the transmitter to retransmit the same packet. If the requested packet has not arrived in a given time period, the missing packet is replaced with zero amplitude samples, resulting in a noticeable SNR improvement.

Watkins and Chen developed some frame erasure concealment techniques specific to the G.728 standard as part of the standardization effort carried out by the ITU-T to meet the demands for future PCS and FPLMTS systems [100]. Their techniques were also based on excitation extrapolation and maintained compatibility with G.728. Their system achieved an MOS of 3.59 with 3% bursty errors and an MOS of 3.79 in clean conditions. This is comparable to the performance achieved by the packet recovery model developed in this thesis. At 10% bursty errors, their system obtained significantly lower MOS. Watkins and Chen also presented techniques to improve robustness by sacrificing compatability with G.728 and allowing for changes in the encoder. These systems achieved noticeable improvements in speech quality when the frame erasure rate was as high as 10%. The research presented in this thesis was performed independently of Watkins and Chen's endeavour within the same time frame.

Chapter 5

Packet Recovery

The speech coding systems considered in the research of missing packet recovery were the LD-CELP system (described in Chapter 3) and the VR-CELP system (described in Chapter 4). The need to consider the LD-CELP system under block erasures is justified by the possible application of LD-CELP for FPLMTS as well as the consideration for future possible application in broadband networks.

In future networks employing ATM with a statistical bandwidth allocation strategy, physical channel structures that are prevalent in fixed rate coding will become less common, and a virtually continuous range of bit-rates will be available within physical network limits. This will make variable-rate codecs an effective choice and justifies the selection of VR-CELP in this research.

In this chapter the concept of packet recovery is presented. The chapter is organized into six sections. In Section 5.1, the packet loss model is described. Section 5.2 provides a brief overview of the packet recovery model (PRM). The block classification and pitch estimation algorithms that are a critical feature of the PRM are presented in Section 5.3. In Section 5.4, the speech residual excitation reconstruction model is presented. Section 5.5 presents the short-term spectral extrapolation procedure. Simulation results are given in Section 5.6 which also includes a discussion of the results.

5.1 Packet Loss/Block Erasure Model

In order to evaluate a speech codec design under packet losses it is necessary to characterize the channel over which speech is transmitted. The speech coder should then be subjected to error patterns and packet losses typical of the channel in order to evaluate its performance. The speech segment corresponding to a packet will be called a block hereafter.

5.1.1 BellCore Model

The packet loss model was developed by BellCore, as requested by the ITU-R, to evaluate the ITU 8 kb/s coder under burst errors typically encountered in FPLMTS applications [105]. The BellCore channel model was defined to simulate the error patterns typical of the portable radio environment. Though the actual error sequence will depend upon the carrier frequency, user speed, detection scheme, type of diversity employed, mean signal-to-noise ratio and the mechanism for providing hand-off, for the sake of convenience, the error statistics can be generated using a few parameters.

The parameters are the block duration and burst block lengths. The block duration is obtained as a suitable multiple of the frame size used in the speech codec. The burst duration is then achieved by a suitable choice of burst block length, which can be defined as the number of consecutive blocks in error. The number of consecutive missing blocks is actually defined by a statistical model. A bursty channel model characterized by block erasures of length 1-6 was introduced in [97]. This model was developed for testing speech codecs in the FPLMTS environment.

The BellCore model is based on a Markov process for generating the error pattern. The states in the model denote the burst length duration (*i.e* number of consecutive blocks in error). The choice of state transition probabilities is governed by the block erasure rate that is desired. Figure 5.1 shows the BellCore loss model which was designed to satisfy a block erasure rate of 3%. The block erasure (or loss) rate will be referred to as BLR hereafter to prevent it being confused with the notation for bit error rate.

Given the model in Figure 5.1, the state transition matrix can be easily obtained. Having obtained the transition matrix, the steady state probabilities of being in each

individual state can be derived using probability theory.

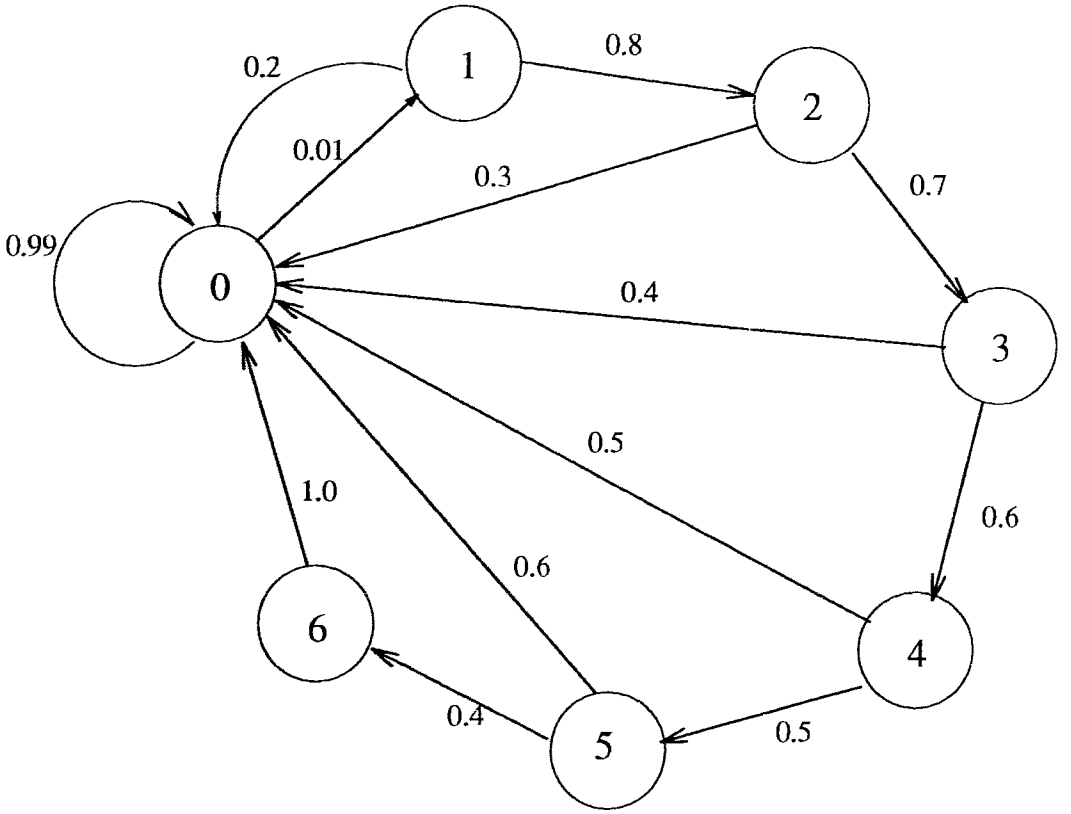


Figure 5.1: BellCore packet loss model

5.1.2 Thesis Model

The packet loss model used in this thesis was obtained by simplifying the BellCore loss model as described in Section 5.1.1. A simple realization of a bursty channel model can be obtained by introducing short, medium, and long bursts. To achieve an error pattern with a desired burst length duration, the model used in this thesis operates as described below. The packet loss model used in this thesis does not better model a typical portable radio environment. However, what it does provide is a greater degree of flexibility in testing for different burst durations.

The states in the model denote the burst length duration. The probability of going from state 0 to state 1 is p and the probability of staying in state 0 is $1 - p$. Once in state 1, the probability of going to the desired state is set at 1.0 and the probability of returning to state 0 from the desired state is also set to 1.0. Figure 5.2 shows the

loss model used in the thesis (for a desired burst length of 6 blocks), which can be designed to satisfy varying BLR by suitably changing the probability p.

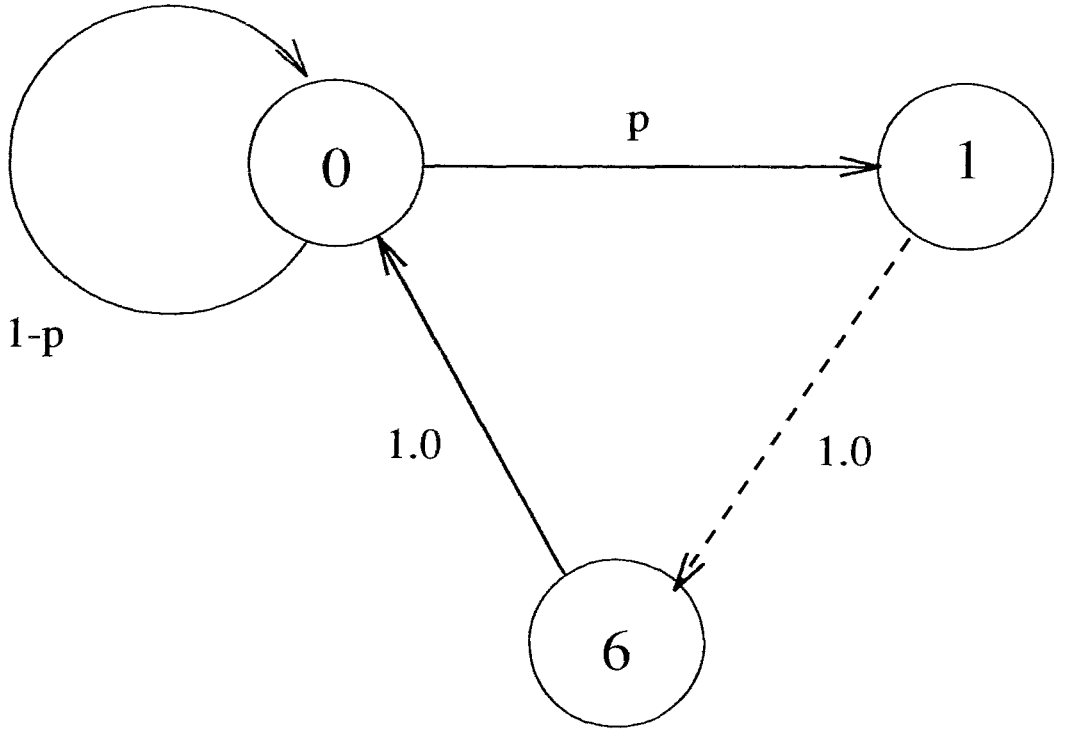


Figure 5.2: Thesis packet loss model

The thesis model as described above is better able to generate long bursts as the probability of being in a long burst state can be easily adjusted without having to recompute the state transition probabilities, unlike in the BellCore model. Also, the probability of being in a long burst state is quite small for the BellCore model. The thesis model, however, is more flexible in setting larger probabilities of being in a long burst state. As a result of which, there wasn't a need to test the system on very large databases to test the performance under long burst durations.

5.2 Packet Recovery Model Overview

In this thesis, the environment in which packet recovery techniques are tested is based on the CELP coding technique. Both forward and backward adaptation coders were considered for the packet recovery tests. The forward adaptation coder was based on a variable-rate CELP (VR-CELP) coder [110]. In backward adaptation, we

considered three systems: the LD-CELP 16 kb/s standard (G.728) [9] and two non-standard systems at 8 kb/s which were developed in Chapter 3. The packet recovery experiments were performed on the standard system only, as it was considered being of greater interest as the ITU-T was considering the use of the LD-CELP coder in FPLMTS. The packet recovery model (PRM) presented is general in nature and can be applied to other CELP coding mechanisms at rates of 4-16 kb/s.

The VR-CELP codec uses a modular approach whereby the general structure and the coding algorithm for all rates is based on the structure of the highest bit-rate system. Lower bit rates are obtained by disabling codec components. Each speech frame is analyzed by a frame classifier and classified as either voiced, unvoiced, transition, or silence in order to determine the coding rate. The system switches between three distinct codec configurations: 8.8 kbit/s for voiced and transition frames, 3.9 kbit/s for unvoiced frames, and 750 bit/s for silence frames, with an overall average bit rate of 4.5 kbit/s based on averaging of typical male/female speech files with 40% silence. An advantage of the VR-CELP codec is the use of fixed excitation sequences for silence, which allows the receiver to resynchronize after a packet loss.

The problem of frame losses is more acute in codecs using backward adaptation of the synthesis filter (e.g. LD-CELP) because the choice of filter parameters is dependent on past (possibly incorrect) synthesized speech, resulting in error propagation and inaccuracies in tracking the correct speech signal.

A number of existing recovery techniques are based on extrapolating the synthesized speech waveform. In a codec which uses backward prediction, this would imply adaptation on a signal which repeats itself in some deterministic fashion. Our experiments show that such an approach may lead to artifacts in the reconstructed speech. To avoid this situation we introduce a recovery procedure in which the excitation and the synthesis filter are extrapolated independently based on class information. The recovery of the missing speech packets is then achieved by passing the extrapolated excitation through the updated (extrapolated) short-term synthesis filter.

The LD-CELP standard is already deployed, hence any proposed changes have to address the issue of inter-operability. Best performance in packetized networks would be obtained by changes in both encoder and decoder; however, this would lead to complete lack of inter-operability with already deployed codecs. On the other hand,

changes in the decoder only, ensure inter-operability but limit the performance improvement under packet losses. We chose a compromise approach in which the only change in the encoder is the addition of class information. This information would be discarded by the decoder using the original standard. The proposed decoder has a default mode in which it can operate without class information at the expense of a performance degradation in packet loss situations.

The recovery models used for both the LD-CELP and the VR-CELP codecs are basically the same. Although the effect of packet losses is significantly different mainly due to forward versus backward adaptation and the presence of the adaptive codebook index (pitch lag) for VR-CELP, we were able to achieve good recovery results for both coders using the same recovery model.

A block diagram of the packet recovery model (PRM) is given in Figure 5.3. We assume that the codec's output bit-stream is packetized by grouping the information generated by a fixed number of frames into a packet. For example, in LD-CELP, 6 frames of 20 samples each result in a 240-bits packet, while in VR-CELP, one frame of 160 samples results in a variable-size packet. The speech segment corresponding to a packet will be called a block hereafter. A classifier is introduced at the transmitter and each block is classified as either silence, unvoiced, voiced, or transition. We assume that the current packet class information is available at the receiver even if the packet is lost.

Class information could be transmitted with the previous packet bit-stream, albeit at the expense of additional delay. Alternatively, the class information bits could be protected using a low-rate forward error correction code. In the former case, in the event of the previous packet also being considered lost, the system makes use only of class information derived at the receiver. However, for the first lost block, the system makes use of class information received from the encoder. In the later case, if the class information bits are received in error, the system can revert back to using only class information derived at the receiver since it had detected that the class information bits were erroneous.

We experimented also with a system using only class information derived at the receiver; for missing packet length equal to one and a burst erasure rate of 3%, this system has a slight quality degradation when compared with the system which transmits class information. However, for longer bursts, the degradation may be noticeable.

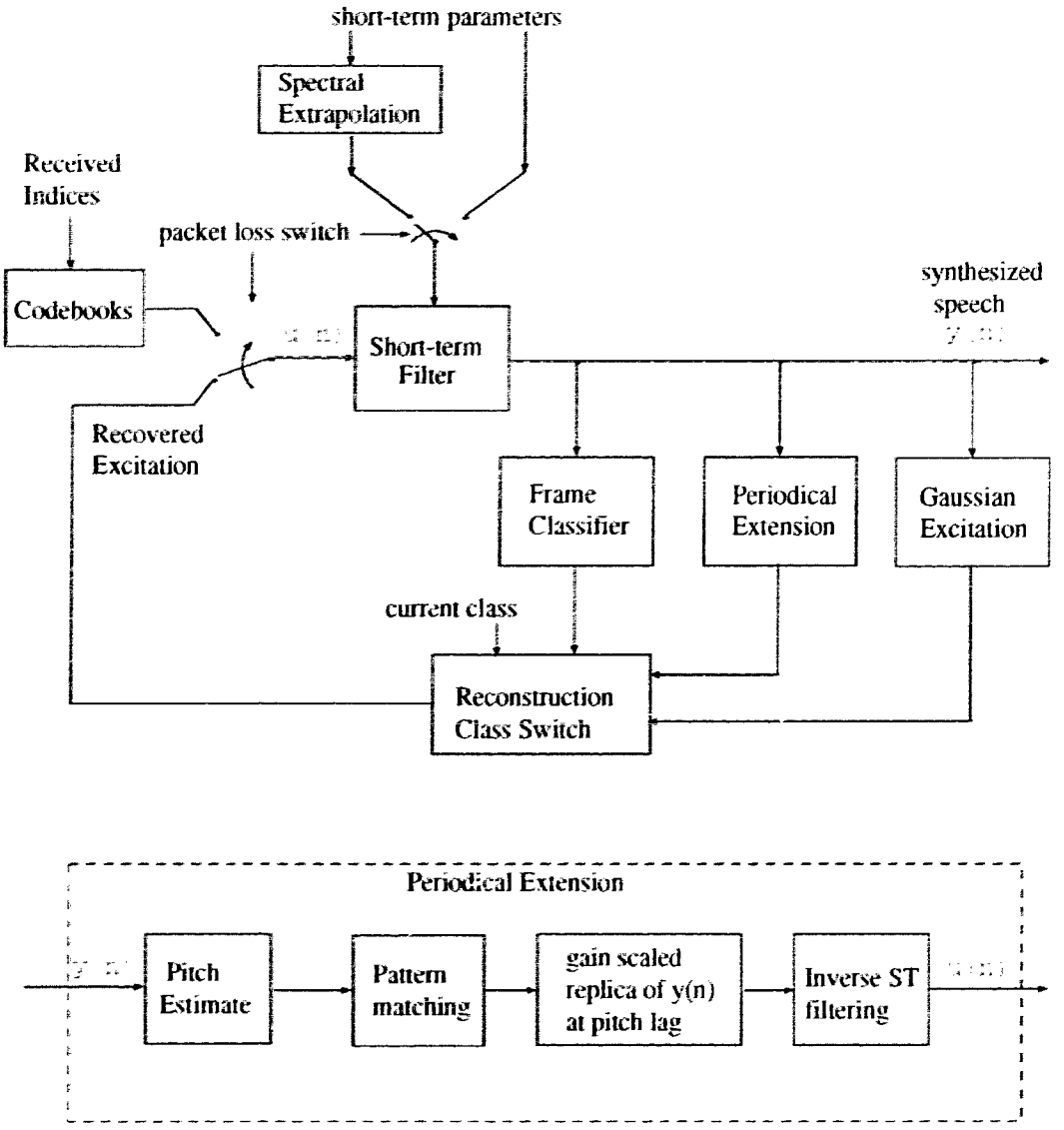


Figure 5.3: Packet Recovery Model

In the case of packet(s) loss, the decoder uses a number of different recovery techniques selected as a function of the past and current decoded class. Voiced class excitation reconstruction is based on periodical extension of the past reconstructed speech and filtering by the inverse of the synthesis filter. Unvoiced class excitation reconstruction is based on Gaussian modeling. Finally, the past residual excitation memory is recomputed for blocks following lost packets in order to improve the quality at transitions. Details of the excitation reconstruction model are given in Section 5.4.

5.3 Classification

The classification of speech is an integral part of the packet recovery model. The use of a classification based model is justified by the need to make use of different procedures to reconstruct speech signals with varying characteristics, resulting in a class dependent recovery model. A good classifier should be able to accurately determine voiced, unvoiced, silence and transition blocks. Transition blocks could represent a voiced onset or offset as well as changes from one voiced phoneme to another. Ineffective classification results in an incorrect choice in the recovery model, which results in reconstructed speech signal which is perceptually annoying.

A critical aspect of classification is pitch estimation, which provides a very effective discriminatory measure in classifying voiced blocks. For a block classifier to be effective, there should be significant interaction between the pitch estimator and block classifier. This section shall provide a brief overview of the block classifier and pitch estimation strategy chosen for the packet recovery model.

5.3.1 Pitch Estimation

In this chapter, the pitch estimation procedure is based on the simplified inverse filtering technique developed by Markel, which is compared with other techniques in the paper by Rabiner *et al.* [84]. The salient features of the pitch estimator are as described below.

The pitch estimator uses a set of candidates in selecting the best possible pitch using the autocorrelation computation on the low-pass 4:1 decimated residual excitation sequence. The best pitch index k_p is then selected by searching around these

pitch candidates for the maximum autocorrelation peak on the undecimated residual excitation signal. The number of candidates selected is obtained by making use of previous class and zero crossing information in determining voiced onsets, and thereby allocating more candidates for such onsets (for onsets use 4 candidates, otherwise use 1 candidate).

The normalized autocorrelation measures computed at the resulting pitch value, k_p, are compared with the normalized autocorrelation measures at the pitch index of the previous block, pk_p, and a choice of pitch index k_{p_1} is made which maximizes the normalized autocorrelation measures. To ensure smoother pitch variation, a heuristic approach was developed which makes use of class information in determining the level of priority given to pk_p in selecting the larger normalized autocorrelation . A provision is made in the algorithm to track momentary abrupt changes in the pitch trajectory, enabling it to return to its steady state value and thereby preventing the estimator from locking-in on an incorrect (sub-) multiple pitch.

5.3.2 Block Classification

The block classifier is based on thresholding. The threshold approach analyzes the speech on a fixed block basis. One or more parameters are derived from the speech source and a class decision is then made. For the classifier to be effective, the following parameters are used in the classification procedure in a concurrent manner:

- Zero crossing information: the number of zero crossings is larger for unvoiced sounds thereby providing a means of effective discrimination.
- Normalized autocorrelation measures at the pitch lag: voiced blocks have larger normalized autocorrelation at the appropriate pitch lag.
- Previous class information: provides for a greater degree of control in the classification procedure as a result of inherent biases due to class dependency.
- Block energy: provides for an effective silence discrimination.

5.4 Speech Residual Excitation Generation Model

The voiced excitation reconstruction model makes use of the previous block pitch estimate, kp_1, as an initial estimate for the current packet. This initial pitch estimate is computed at the receiver based on previous reconstructed speech. The estimate is then refined using a pattern matching procedure to select the best lag, kp_2, in a window around the initial estimate [41, 40]. The best lag is selected by minimizing the prediction error criterion

$$e(kp_2) = \sum_{n=-kp_1+1}^{n=0} (y(n) - g_{kp_2} y(n - kp_2))^2 \tag{5.1}$$

where $y(n)$ is a buffer containing the past reconstructed speech signal and g_{kp_2} is a gain value. The value of the gain which minimizes the prediction error criterion is given by

$$g_{kp_2} = \sum_{n=-kp_1+1}^{n=0} (y(n)y(n - kp_2)) \Big/ \sum_{n=-kp_1+1}^{n=0} (y(n - kp_2)y(n - kp_2)) \tag{5.2}$$

An extrapolation of voiced excitation is obtained by passing through the inverse short-term synthesis filter a scaled replica of previously synthesized speech with a lag of kp_2. The scaling coefficient is obtained by an empirical procedure which combines the gains obtained in the pattern matching procedure together with an estimate obtained from the trajectory of the main pitch-pulse peak. Figure 5.3 shows the voiced excitation model as part of the more general PRM.

Figure 5.4 shows a typical probability density function of an unvoiced residual signal and compares it to the Gaussian and Laplacian probability density functions. As a result of this analysis, a Gaussian generated signal was chosen as an appropriate model to approximate actual unvoiced residual signals. The parameters of the Gaussian process for unvoiced excitation extrapolation are estimated from the previous excitation signal [41, 40].

For offset blocks, a transition from voiced excitation to unvoiced excitation is positioned in the middle of the missing block. Randomization is used in the excitation extrapolation for offset transitions in order to avoid excessive periodicity in the residual during offsets.

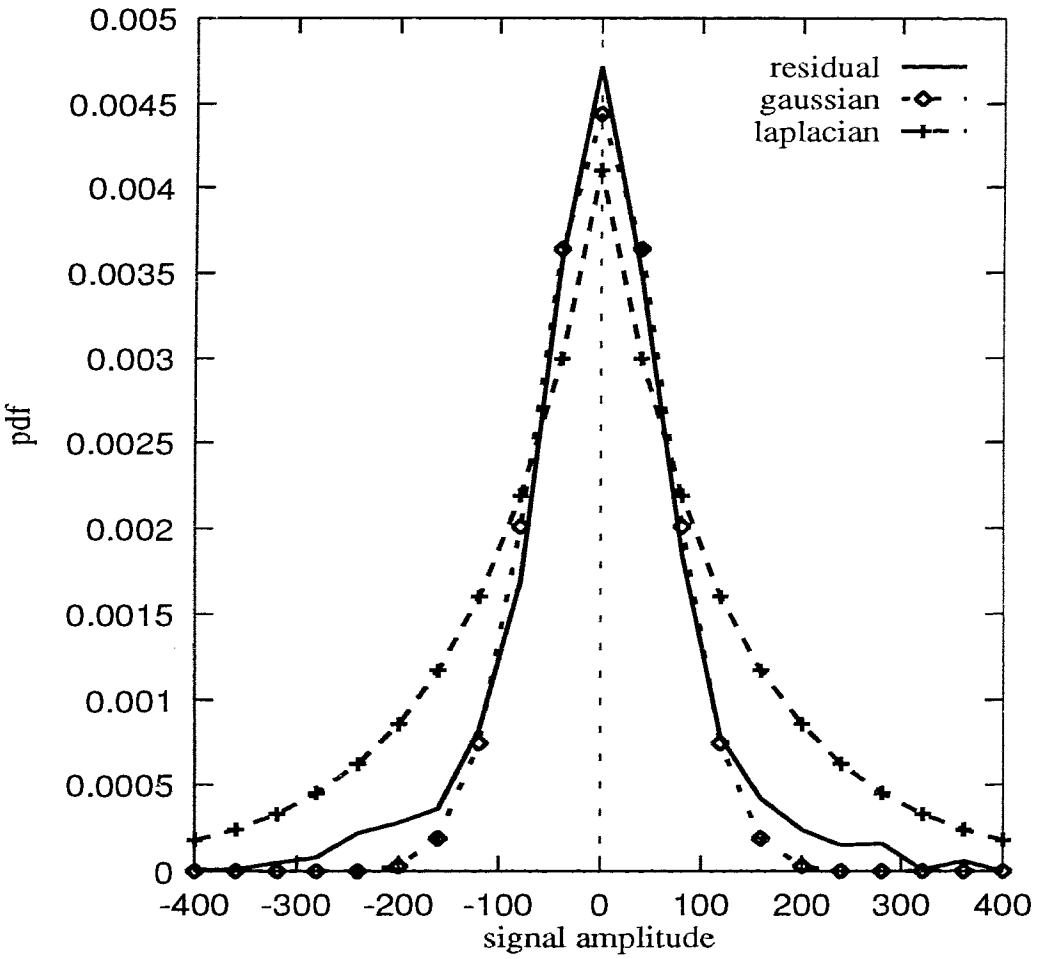


Figure 5.4: Unvoiced Residual Excitation Probability Density Function

Transition excitation generation is very critical towards achieving satisfactory performance under packet losses. For voiced transitions, which could correspond to unvoiced to voiced onset transitions as well as voiced to voiced transitions, we make use of the voiced excitation model. Just modeling the lost excitation using a voiced recovery model is not sufficient as there is not enough accurate voiced history for effective replication. This leads to inadequate voicing in the voiced blocks following as a result of absent or misplaced pitch pulses in the excitation memory, which result in the excitation memory being corrupted. This results in error propagation and hence poor reconstructed speech quality.

Solution to this problem is, before synthesizing the next correctly received block, recalculate the lost block by making use of information from the next correctly received block. The recalculation of the excitation reduces error propagation effects dramatically. Offcourse one can resynthesize the speech in the lost packet at the expense of an extra block of delay but the experiments here avoided that. The stochastic excitation and adaptive codebook information in the first correctly received block are used to improve the past residual excitation memory before synthesizing the correctly received block. This is done by computing an initial residual excitation for the correctly received block and then extrapolating it backwards to recalculate the past residual excitation memory. This procedure was only performed for the PRM applied to the VR-CELP coder.

In an unvoiced to voiced transition, mistracking can occur due to absent pitch pulses. The adaptive codebook information in the next correctly received block can be used to obtain an estimate of the pitch period at the start of the current block (following a lost block). Having obtained an estimate of the pitch period at the end of the lost block, the residual excitation of the lost block is recalculated by extrapolating backwards at the appropriate pitch lag, the stochastic contribution of the residual excitation in the next correctly received block.

In a voiced to voiced transition, mistracking can occur due to misplaced pitch pulses caused by significant changes in the pitch period across the lost block. The adaptive codebook information in the next correctly received block together with the pitch period pk_2 prior to the lost block can be used to obtain an estimate of the pitch period at the start of the current block (following a lost block). Having obtained an estimate of the pitch period at the end of the lost block, the residual excitation of the

lost block is recalculated by replicating the past excitation from a buffer containing the last correctly received block. The pitch period is selected to ensure a smooth pitch evolution from the pitch period pk_2 prior to the lost block, to one subsequent to it.

Figure 5.5 shows a plot of the residual excitation signal and the reconstructed speech signal (with (blue plot) and without (red plot) excitation recalculation) for an unvoiced to voiced transition to better illustrate the performance improvement obtained by making use of some form of excitation recalculation. The plots are compared with similar plots obtained when the system operates under no errors (green plot), *i.e.* the codec system with BLR = 0%. In the plots shown in Figures 5.5 and 5.6, the last 160 samples correspond to the first correctly received block after the packet loss.

Figure 5.6 shows a plot of the residual excitation signal and the reconstructed speech signal (with (blue plot) and without (red plot) excitation recalculation) for a voiced to voiced transition with a fast evolving pitch period. The system without excitation recalculation is unable to adapt to the varying pitch, which in turn leads to misalignment in the residual signal and hence reconstructed speech, which is perceptually annoying. The excitation recalculation case, on the other hand, is able to ensure a smoother pitch variation which results in the pitch peaks being aligned closer to the reference signal (green plot), *i.e.* codec with BLR = 0%. Some portions of the residual excitation and reconstructed signal plots for the system with excitation recalculation overlap with the corresponding plots of the reference signals.

5.5 Short-term Spectral Extrapolation

A number of existing recovery techniques are based on extrapolating the synthesized speech waveform. In a codec which uses backward adaptation of the short-term synthesis filter, this would imply adaptation on a signal which repeats itself in some deterministic fashion. Our experiments show that such an approach may lead to artifacts in the reconstructed speech due to the short-term filter following a forced trajectory path, which differs from the actual trajectory under clean conditions. To avoid this situation we introduce a recovery procedu which the excitation and the synthesis filter are extrapolated independently based on class information. The recovery of the missing speech packets is then achieved by passing the extrapolated

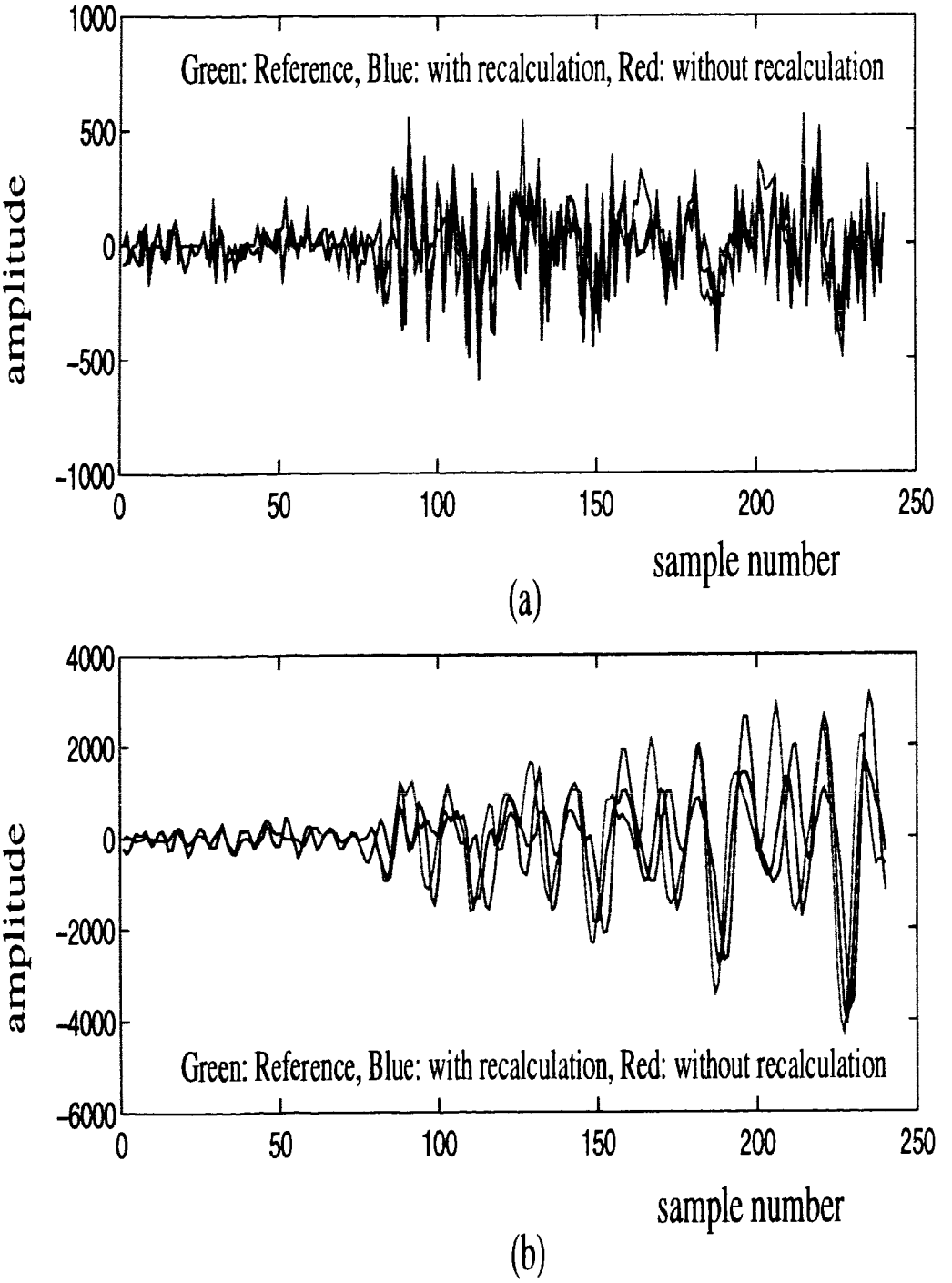


Figure 5.5: Unvoiced to voiced transition with and without excitation recalculation a) residual excitation and b) speech signal

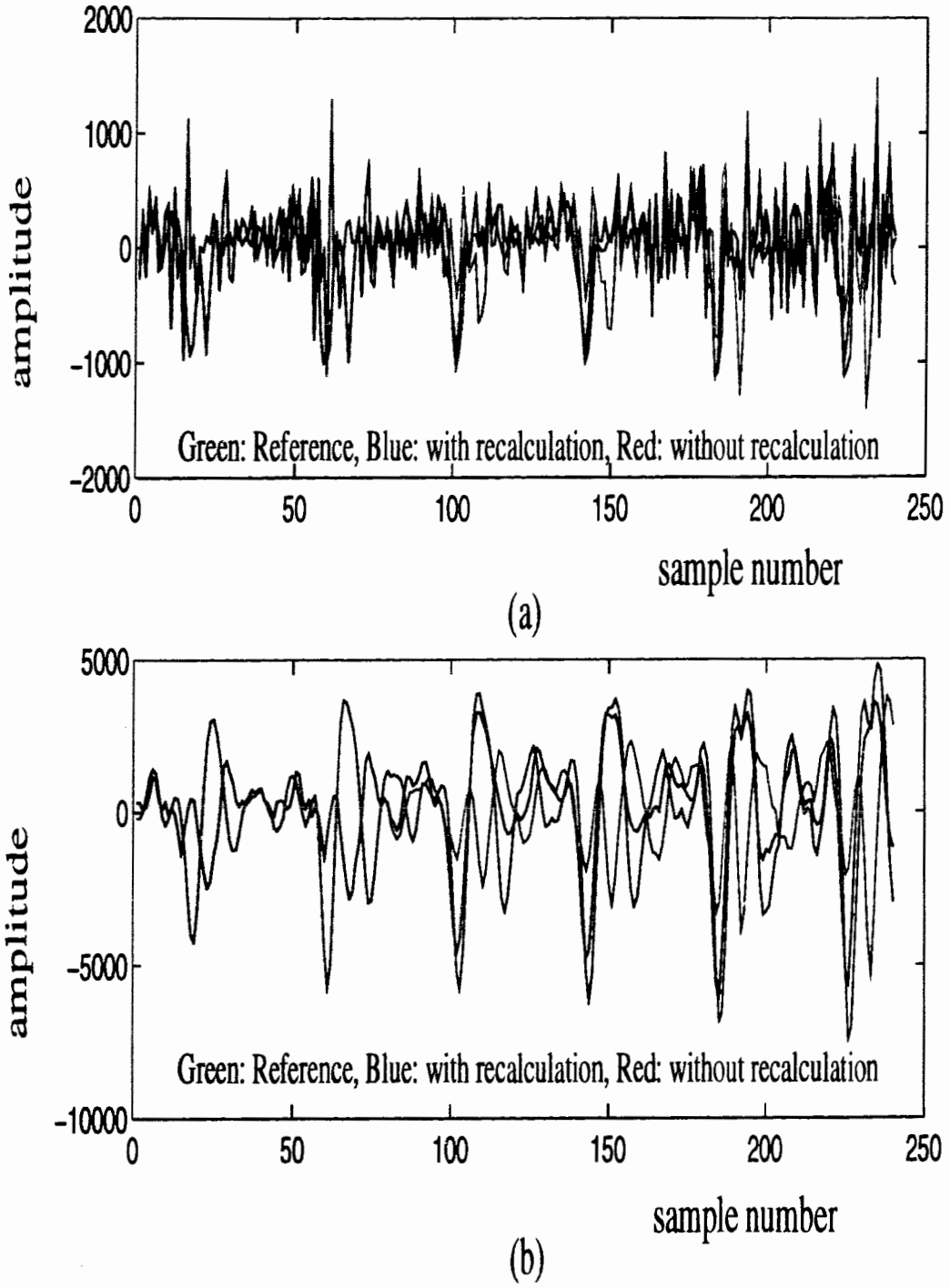


Figure 5.6: Voiced to voiced transition with and without excitation recalculation a) residual excitation and b) speech signal

excitation through the updated (extrapolated) short-term synthesis filter.

The spectral extrapolation of the short-term filter is based on the least-squares fading memory polynomial filter, which makes use of the discrete Laguerre polynomials [76]. The extrapolation is applied to the reflection coefficient trajectories rc_n,

$$rc_n = \left[\begin{array}{cccc} rc_n^{(0)} & rc_n^{(1)} & \ldots & rc_n^{(m)} \end{array} \right] \tag{5.3}$$

where rc_n is a vector of the current reflection coefficients at time n, with dimension m less than or equal to the order of the short-term filter.

The spectral information rc_n is extrapolated by a polynomial in k of degree d, $[p^*(k)]_n$, where k is an index pointing back on the time scale and n is the current time instant and is included to show that the polynomial is found on data up to rc_n. The objective is to minimize

$$e_n = \sum_{k=0}^{\infty} \{rc_{n-k} - [p^*(k)]_n\}^2 \theta^k \tag{5.4}$$

where θ^k is a discount factor, whose value decreases as k increases, provided $\theta < 1$. The value of θ used is close to 1.0 and is based on the following tradeoff: if θ is small the transient errors are small but the smoothing is bad, on the other hand making θ close to unity results in good smoothing but bad transients. $[p^*(k)]_n$ may be expressed as a linear combination of the discrete Laguerre polynomials $\varphi_j(k)$,

$$[p^*(k)]_n = \sum_{j=0}^{d} (\beta_j)_n \varphi_j(k) \tag{5.5}$$

Replacing $[p^*(k)]_n$ in equation 5.4 by equation 5.5 and solving for minimum e_n with respect to $(\beta_j)_n$, results in the best choice of the coefficients $(\beta_j)_n$,

$$(\beta_j)_n = \sum_{l=0}^{\infty} rc_{n-l} \varphi_j(l) \theta^l \tag{5.6}$$

which minimize the error e_n. Substituting, $(\beta_j)_n$ back into equation 5.5 results in

$$[p^*(k)]_n = \sum_{j=0}^{d} \left[\sum_{l=0}^{\infty} rc_{n-l} \varphi_j(l) \theta^l \right] \varphi_j(k) \tag{5.7}$$

$[p^*(k)]_n$ can now be considered as an estimate of rc_n based on observations till rc_n. As rc_n can be estimated by $[p^*(k)]_n$, so also can derivatives of rc_n be estimated by

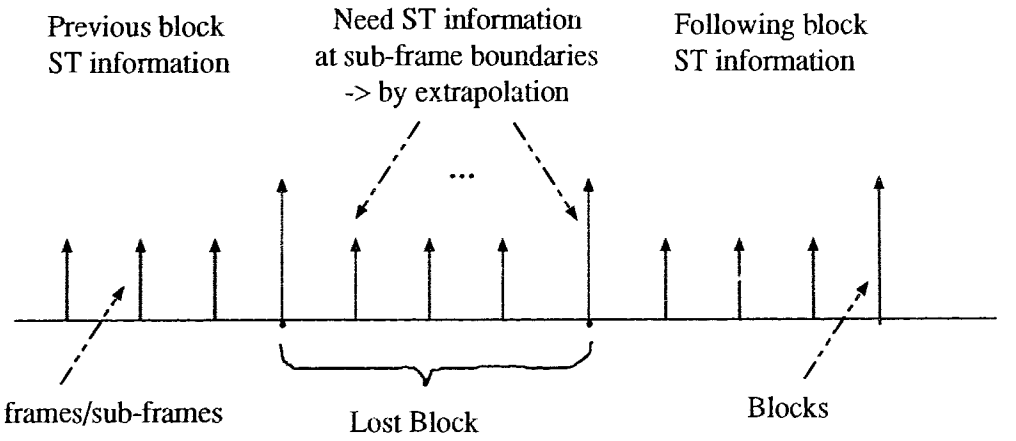
time-derivatives of $[p^*(k)]_n$. The 1-step prediction state-vectors (to be defined shortly) can be obtained by first solving for derivatives of the process rc_n and setting $k = -1$. The above analysis is unsatisfactory as it is computationally inefficient. A convenient recursive form can however be derived. A compact recursive form for the state-vectors using a degree 1 ($d = 1$) polynomial is then obtained as follows (the proof is lengthy and is presented in [76]):

$$\begin{aligned} (rc_1^*)_{n+1,n} &= (rc_1^*)_{n,n-1} + (1-\theta)^2\epsilon_n \\ (rc_0^*)_{n+1,n} &= (rc_0^*)_{n,n-1} + (rc_1^*)_{n+1,n} + (1-\theta^2)\epsilon_n \end{aligned} \tag{5.8}$$

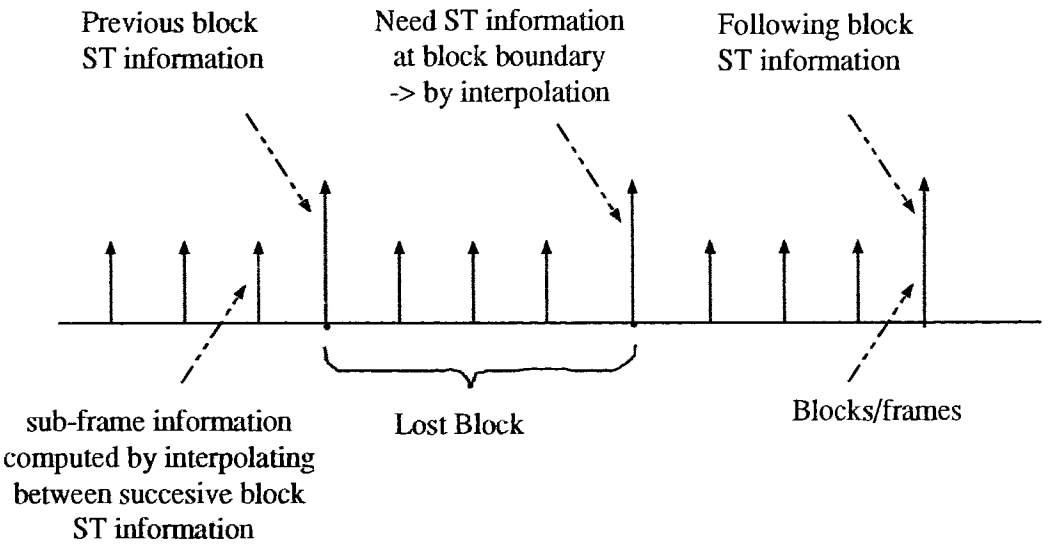
where $(rc_0^*)_{n,n-1}$ and $(rc_1^*)_{n,n-1}$ are the 1-step prediction state vectors with $(rc_0^*)_{n,n-1}$ denoting the position state vector and $(rc_1^*)_{n,n-1}$ denoting the velocity state vector (*i.e.* rate of change of position). In equation 5.8, $(rc_0^*)_{n+1,n}$ and $(rc_1^*)_{n+1,n}$ are the 1-step position and velocity predictions respectively, which minimize equation 5.4, and $\epsilon_n = rc_n - (rc_0^*)_{n,n-1}$ is the prediction error. We now use $(rc_0^*)_{n+1,n}$ as the choice of reflection coefficients at time $n+1$.

Figure 5.7 shows a sketch of the short-term information timing diagram that better illustrates what is involved in short-term information extrapolation and/or interpolation in both the LD-CELP and VR-CELP systems. In VR-CELP a block consists of 1 frame which in turn consists of 4 subframes and in LD-CELP a block contains a number of frames (depending on the choice of block size) which in turn consist of only one subframe each. The short-term information is computed every frame for both LD-CELP and VR-CELP. In VR-CELP the short-term information is also required at each subframe. The subframe values are computed by linearly interpolating between the short-term information obtained for the current and preceding frames.

In LD-CELP when a block is lost the short-term information for each subframe within the lost block is obtained by extrapolating the short-term information from subframes of preceding blocks. In VR-CELP as the short-term information is transmitted to the receiver, once a block is correctly received, short-term information mistracking is corrected for, resulting in a lack of urgency to perform short-term extrapolation. However, since the last block was lost, the previous frame short-term information in unavailable for the interpolation necessary to obtain the short-term information at each subframe. This is corrected by interpolating between the current correctly received block and the last correctly received block to obtain the necessary



(a)



(b)

Figure 5.7: Short term information extrapolation/interpolation timing sketch for (a) LD-CELP (b) VR-CELP

frame short-term information.

Figure 5.8 shows a plot of the 1st and 2nd order reflection coefficient trajectories with (blue plot) and without (red plot) extrapolation. These plots are compared with reference plots obtained by considering the system under BLR=0% (green plot). The system with extrapolation is better able to adapt to the varying coefficient trajectories, which in turn leads to a perceptually improved reconstructed signal.

5.6 Simulation Results and Discussion

Our objective was to develop a recovery algorithm which when used in a CELP codec over a noisy channel with packet losses, would result in a performance as close as possible to that of the system under error free conditions. Ideally, a difference of the order of less than 0.5 MOS (mean opinion score) would be desirable. The speech decoder should attempt to estimate speech during short and medium length packet losses. During longer bursts of missing packets, the reconstructed speech output should decay progressively and recover as quickly as possible after a burst to its error free state. A burst length of one packet corresponds to a loss of 120 samples of speech information in LD-CELP and 160 samples in VR-CELP. Burst lengths of 1, 3 and 6 packets approximately correspond to short, medium and long bursts respectively.

Table 5.1 shows the SegSNR Performance of the system under various block erasure rates (BLR) as well as different burst lengths. It must be stressed that SegSNR is not a very effective measure of quality for bursts, especially long ones, since there can be significant quantitative differences in the signals which are not reflected in results on perceptual quality. The results of a MOS test are presented in Table 5.2. The performance of the proposed PRM system is compared with a system in which recovery is based on simple repetition of the indices from the last correctly received packet. This simple recovery system is referred to as the reference system (REF) in Tables 5.1- 5.2. It should be noted that the reference system based on the simple recovery technique described above, performs much better than a system in which the lost packet information is replaced by a random choice of indices. Note that random index substitution may occur in applications if packets affected by fading are transferred to the decoder when block erasures are not detected.

At block erasure rates of 3% and a burst length of one packet, MOS evaluation tests

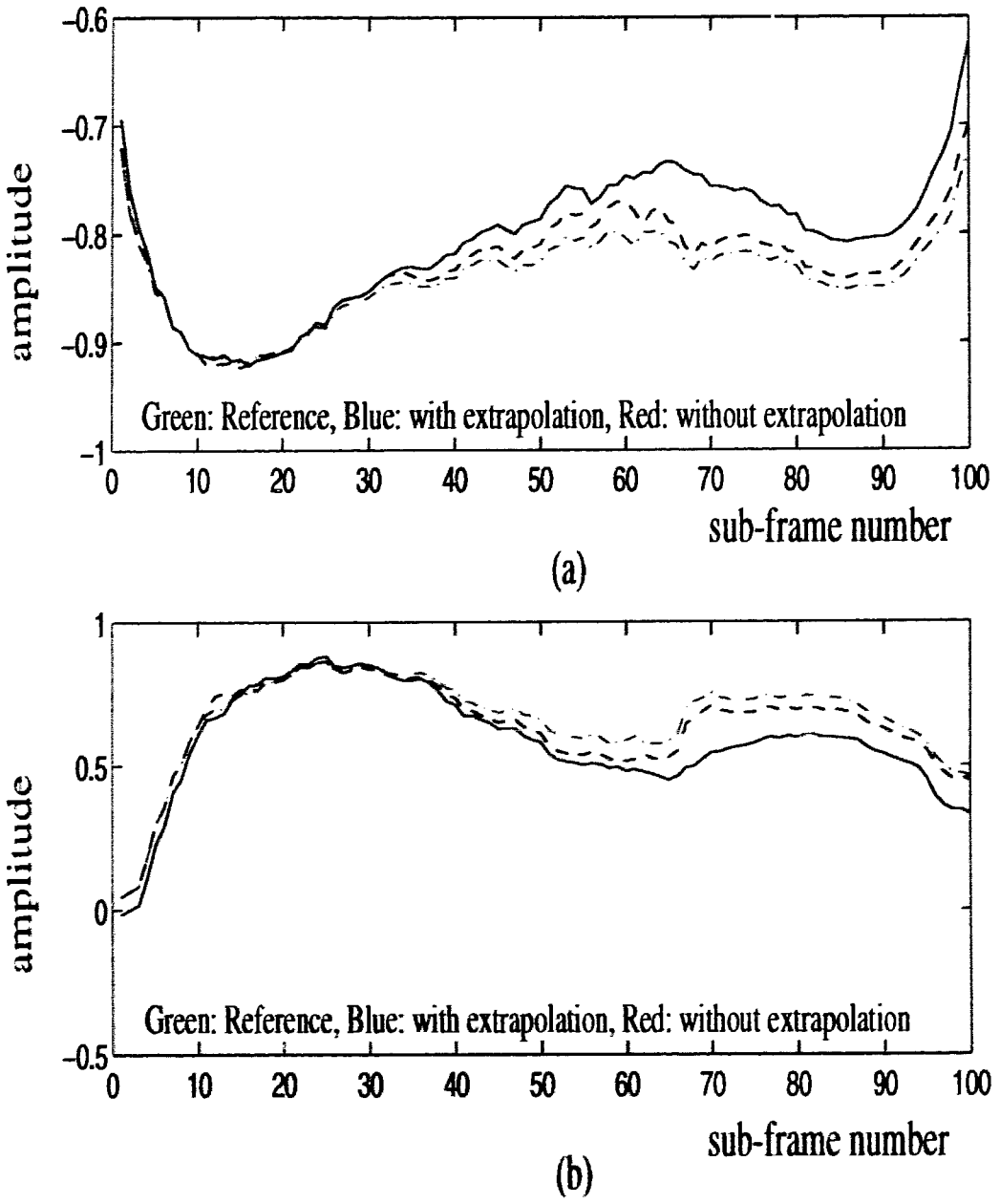


Figure 5.8: Short term parameter Trajectory a) 1st order reflection coefficient b) 2nd order reflection coefficient

System	clean	BLR=3%		BLR=10%	
		PRM	REF	PRM	REF
A, bl=1	17.19	13.94	12.75	9.25	6.49
A, bl=3	17.19	15.04	14.51	11.46	9.34
A, bl=6	17.19	16.68	16.02	9.91	8.59
B, bl=1	11.26	9.08	8.64	6.01	5.34
B, bl=3	11.26	7.82	7.42	6.07	5.65
B, bl=6	11.26	9.63	9.40	6.16	5.96

Table 5.1: Segmental SNR values for BLR=3%, 10%; burst length=1, 3, 6; for A) LD-CELP and B) VR-CELP

	clean	BLR=3%				BLR=10%	
		bl=1		bl=3		bl=1	
		PRM	REF	PRM	REF	PRM	REF
A	3.85	3.74	3.29	3.45	2.87	3.17	2.66
B	3.70	3.55	3.40	3.25	3.10	3.24	3.03

Table 5.2: MOS for BLR=3%, 10%; burst length=1, 3; for systems A) LD-CELP and B) VR-CELP

have shown only a relatively minor perceptual degradation of the PRM systems over the error free systems. For longer bursts some performance degradation occurs mainly due to mistracking of speech transitions, though the MOS score for the PRM systems are still much better than that of the reference systems. Note that at BLR = 3% and burst length (bl = 1) for LD-CELP, the PRM system achieves an improvement of 0.45 on the MOS scale and 1.19 dB in SegSNR. On the other hand for the VR-CELP, the PRM system achieves an improvement of 0.15 on the MOS scale and 0.44 dB in SegSNR. With a BLR = 10% and burst length (bl = 1) for LD-CELP, the PRM system achieves an improvement of 0.51 on the MOS scale and 2.76 dB in SegSNR. On the other hand for the VR-CELP, the PRM system achieves an improvement of 0.21 on the MOS scale and 0.67 dB in SegSNR.

MOS test results show that the LD-CELP-PRM system achieves good recovery performance for short burst lengths at block erasure rates as high as 10% (MOS is approximately 3.2 at 10% BLR). The proposed PRM system achieves results better by 0.45-0.6 on the MOS scale when compared to the reference system.

MOS test results show that the VR-CELP-PRM system achieves good recovery performance for short burst lengths at block erasure rates as high as 10%. The difference between the recovered and reference system is not as much as in the case of the recovery model tested on the LD-CELP system. A possible reason is the better pitch tracking possible in VR-CELP due to adaptive codebook information still preserved from the preceding frame in the reference system.

It should be noted that the reference system based on the simple recovery technique described above, performs much better than a system in which the lost packet information is replaced by a random choice of indices. The LD-CELP system with a burst length (bl = 1) and BLR = 3% gives a SegSNR of 8.17 dB. The LD-CELP with random choices for missing indices has a subjective quality which was found to be too low for formal MOS testing. Similar MOS quality evaluation findings were made for the VR-CELP system with random choices for missing indices.

The packet recovery techniques developed in this thesis can be extended to work with coding techniques other than CELP, such as MBE and STC. These coders characterize the evolving short-term spectra of the speech by extracting and quantizing certain parameters which specify the spectra, giving particular attention to the pitch harmonics in voiced speech. The key feature of sinusoidal coders is that voiced speech is synthesized in the decoder by generating a sum of sinusoids whose frequencies and phases are carefully modified in successive frames to represent and track the evolving short-term spectral character of the original signal. The recovery model developed in this thesis is based on speech classification with excitation and spectral extrapolation. Based on this idea, the spectral information in sinusoidal coders can be extrapolated. The missing speech segment can also be obtained by extrapolating the past speech segments making use of pattern matching techniques together with speech classification. It is expected that the performance of such a system under packet losses would be somewhat equivalent to the performance of recovery models based on CELP coders. Offcourse, one could also adapt the recovery model to fit the specifics of the sinusoidal coding technique. Sinusoidal coders can be made more robust by making use of an embedded coding design formulation. The recovery model then makes use of this embedded principle in packet recovery. However, these techniques are very specific to the coding model used and can not be easily generalized to work with other coding techniques.

Chapter 6

Conclusions and Future Work

The objective of this thesis was the study of speech coding for packet networks. In network environment, we have many possible sources of delay, hence low-delay coders are required. In this research, two approaches towards achieving a high quality low-delay speech coder at 8 kb/s were developed: a backward 8 kb/s coder, which made use of a 3-tap hybrid backward adaptive open-loop pitch predictor and a partially-forward scheme, which used a 3-tap forward adapted long-term adaptive codebook. Also, this research investigated the effect of coefficient adaptation on speech quality under clean and noisy channel conditions using various synthesis driving signals.

Informal subjective tests indicated that the partially-forward 8 kb/s system had quality comparable to the 8 kb/s VSELP standard in clean conditions, while the backward system was just slightly inferior. For noisy channels, at bit error rates of 10^{-3}, both systems achieve MOS scores which were within 0.2 on the MOS scale from the scores obtained in clean conditions. In the backward system, the use of the short-term adaptation signal $u_{ls}(n)$ in short-term coefficient adaptation, resulted in a robust codec, which achieved good subjective quality even at BER as high as 10^{-2}.

The development of a low-rate high quality low-delay speech coder was a precursor to the primary motivation of this thesis: the study of packet recovery techniques in code excited linear prediction (CELP) based speech coders. The recovery techniques were based on speech classification and spectral extrapolation. The recovery system extrapolates independently the excitation signal and the short-term synthesis filter using an extrapolation strategy based on speech classification (voiced, unvoiced, transition, silence). The extrapolation of the short-term filter uses a least-squares fading

memory polynomial filter applied to reflection coefficients.

Quality evaluations of the recovery system applied to the LD-CELP G.728 standard and a VAR-CELP system for random and burst block erasures indicated that the system was robust up to a block erasure rate of 10%. Very little degradation in quality was observed at erasure rates up to 3%. The performance of the proposed PRM system is compared with a system in which recovery is based on simple repetition of the indices from the last correctly received packet. This simple recovery system is referred to as the reference system. The proposed LD-CELP-PRM system achieved results better by 0.45-0.6 on the MOS scale when compared to the reference system. In the VR-CELP-PRM system, the difference between the recovered and reference system was not as much as was the case for the recovery model tested on the LD-CELP coder.

6.1 Future Work

Some suggestions for possible future work are:

1. Improvements in residual excitation computation for transitions need to be addressed. Transition excitation modeling is yet an unsolved problem. The issue of incorrect modeling of transitions, represents a hurdle towards achieving excellent robustness to packet losses unless transition information is protected from such losses.

2. Improving predicted gain and short-term filter misalignments due to pitch errors in the recovery system applied to the LD-CELP codec. This is still a source of quality degradation in the LD-CELP-PRM system as a result of misalignments in the residual excitation signal. The misalignments can be reduced by making use of the future correctly received excitation to realign the residual excitation.

3. Adaptive codebook misalignments in the recovery system used in the VR-CELP codec can be further improved.

4. Looking at alternative extrapolation procedures for short-term spectral extrapolation. However, it must be stressed that correct computation of residual excitation is perceptually more important. In low-delay codecs, the ability to track accurately the short-term information relates directly to having obtained

an accurate residual excitation signal. Therefore, the focus of any packet recovery model should be towards excitation modeling, particularly in forward adaptation coders.

5. A preliminary test was performed to examine the effect on speech quality of varying the block size to 80 samples for the LD-CELP-PRM. There was a negligible change in quality. However, this has to be further studied to better understand the effect of block size on speech quality.

References

[1] H. Abut. *Vector Quantization.* IEEE Press, 1990.

[2] B. S. Atal and M. Schroeder. Stochastic coding of speech signals at very low bit rates. In *Intl. Conf. on Communications*, May 1984.

[3] T. P. Barnwell III. Recursive windowing for generating autocorrelation coefficients for LPC analysis. *IEEE Trans. Accoustics, Speech and Signal Processing*, ASSP-29(5), October 1981.

[4] J. Bellamy. *Digital Telephony.* Wiley, 1990.

[5] R. E. Blahut. *Principles and Practice of Information Theory.* Addison-Wesley, 1987.

[6] A. Buzo, A.H.Gray, R. M. Gray, and J. D. Markel. Speech coding based upon vector quantization. *IEEE Trans. Accoustic, Speech and Signal Processing*, ASSP-28(5), October 1980.

[7] CCITT Recommendation G.764. Voice packetization - packetized voice protocols. Technical report, International Telecommunications Union, 1990.

[8] J. H. Chen. A robust low-delay speech coder at 16 kb/s. In *IEEE Intl. Conf. Accoustic, Speech and Signal Processing*, 1989.

[9] J. H. Chen. High-quality 16 kb/s speech coding with a one-way delay less than 2 ms. In *IEEE Int. Conf. Accoustic, Speech, and Signal Processing*, April 1990.

[10] J. H. Chen and A. Gersho. Gain adaptive vector quantization with application to speech coding. *IEEE Trans. Communications*, COM-35, September 1987.

[11] J. H. Chen and A. Gersho. Real-time vector APC speech coding at 4800 bps with adaptive postfiltering. In *Intl. Conf. Accoustic, Speech and Signal Processing*, April 1987.

[12] J. H. Chen, N. Jayant, and R. V. Cox. Improving the performance of the 16 kb/s LD-CELP speech coder. In *IEEE Intl. Conf. Accoustics, Speech and Signal Processing*, 1992.

[13] J. H. Chen and M. S. Rauchwerk. An 8 kb/s low-delay CELP speech coder. In *IEEE Intl. Conf. Globecom*, 1991.

[14] V. Cuperman. Speech coding. *Advances in Electronics and Electron Physics*, 82(2), February 1991.

[15] 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 kb/s. In *IEEE Intl. Conf. Globecom*, 1989.

[16] V. Cuperman and R. Pettigrew. Robust low-complexity backward adaptive pitch predictor for low-delay speech coding. *IEE Proceedings-I*, 138(4), August 1991.

[17] G. Davidson and A. Gersho. Complexity reduction methods for vector excitation coding. In *Intl. Conf. Accoustic, Speech and Signal Processing*, April 1986.

[18] G. Davidson and A. Gersho. Multiple-stage vector excitation coding of speech waveforms. In *IEEE Conf. Accoustics, Speech and Signal Processing*, April 1988.

[19] G. Davidson, M. Yong, and A. Gersho. Real-time vector excitation coding of speech at 4800 bps. In *IEEE Intl. Conf. Accoustic, Speech and Signal Processing*, April 1987.

[20] M. DePrycker. *Asynchronous Transfer Mode: Solution for Broadband ISDN.* Prentice Hall, 1995.

[21] S. Dimolitsas. Standardization of speech coding technology for network applications. *IEEE Communications Magazine*, October 1993.

[22] N. Erdol, C. Castelluccia, and A. Zilouchian. Recovery of missing speech packets using the short-time energy and zero crossing measurements. *IEEE Trans on Speech and Audio Processing*, 1(3), July 1993.

[23] W. R. Erhart and J. D. Gibson. A speech packet recovery technique using a model based tree search interpolator. In *IEEE Intl. Workshop on Speech Coding*, October 1993.

[24] S. Furui. *Digital Speech Processing, Synthesis and Recognition.* Marcel-Dekker, 1989.

[25] W. Gardner, P. Jacobs, and C. Lee. QCELP: A variable-rate speech coder for CDMA digital cellular. In B. S. Atal, V. Cuperman, and A. Gersho, editors, *Speech and Audio Coding for Wireless and Network Applications.* Kluwer Academic, 1993.

[26] C. G. Gerlach. A probabilistic framework for optimum speech extrapolation in digital mobile radio. In *IEEE Intl. Conf. Accoustics, Speech and Signal Processing*, March 1993.

[27] A. Gersho. Advances in speech and audio compression. *Proceedings of the IEEE*, 82(6), June 1994.

[28] A. Gersho and R. Gray. *Vector Quantization and Signal Compression.* Kluwer Academic, 1991.

[29] A. Gersho and E. Paksoy. An overview of variable rate speech coding for cellular networks. In *Intl. Conf. on Selected Topics in Wireless Communications*, June 1992.

[30] I. Gerson and M. Jasiuk. Vector sum excited linear prediction (VSELP) speech coding at 8 kb/s. In *IEEE Intl. Conf. Accoustic, Speech and Signal Processing*, April 1990.

[31] J. D. Gibson, Y. C. Cheong, H.C. Woo, and W.-W. Chang. Backward adaptive prediction algcrithms in multi-tree speech coders. In B. S. Atal, V. Cuperman, and A. Gersho, editors, *Advances in Speech Coding.* Kluwer Academic, 1991.

[32] D. J. Goodman. Cellular packet communications. *IEEE Trans. Communications*, 38(8), August 1990.

[33] D. J. Goodman, G. B. Lockhart, O. J. Wassem, and W.-C. Wong. Waveform substitution techniques for recovering missing speech segments in packet voice communications. *IEEE Trans on Acoustics, Speech, and Signal Processing*, ASSP-34(6), December 1986.

[34] R. M. Gray. Vector quantization. *IEEE ASSP Magazine*, (2), April 1984.

[35] D. W. Griffin and J. S. Lim. Multi-band excitation vocoder. *IEEE Trans. Acoustic, Speech and Signal Processing*, 36(8), August 1988.

[36] K. Hellwig, P. Vary, D. Massaloux, and J.P. Petit. Speech codec for the european mobile radio system. In *IEEE Intl. Conf. Globecom*, November 1989.

[37] M. L. Honig and D. G. Messerschmitt. *Adaptive Filters: Structures, Algorithms and Applications.* Kluwer Academic, 1984.

[38] A. Husain and V. Cuperman. Low-delay vector excitation speech coding at 8 kb/s. In *IEEE Intl. Workshop on Intelligient Signal Processing and Communications Systems*, March 1992.

[39] A. Husain and V. Cuperman. Lattice low-delay vector excitation for 8 kb/s speech coding. In B. S. Atal, V. Cuperman, and A. Gersho, editors, *Speech and Audio Coding for Wireless and Network Applications*, pages 33–40. Kluwer Academic, 1993.

[40] A. Husain and V. Cuperman. Classification and spectral extrapolation based packet reconstruction for low-delay speech coding. In *IEEE Intl. Conf. Globecom*, November 1994.

[41] A. Husain and V. Cuperman. Techniques for missing packet reconstruction in speech coding. In *17th Biennial Symposium on Communications*, Queens University, May 1994.

[42] A. Husain and V. Cuperman. Reconstruction of missing packets for CELP-based speech coders. In *IEEE Intl. Conf. Accoustic, Speech and Signal Processing*, May 1995.

[43] A. Husain and V. Cuperman. Reconstruction of missing packets for CELP-based speech coders. *to be submitted to IEEE Trans on Speech and Audio Processing*, July 1996.

[44] IEEE. Proceedings of the IEEE Intl. Workshop on Speech Coding, 1991/93/95.

[45] INMARSAT. Inmarsat-M voice coding system description (draft version). Technical report, INMARSAT, December 1990.

[46] ITU-T Draft Recommendation G.728. *Coding of Speech at 16 kb/s using Low-Delay Code Excited Linear Prediction (LD-CELP)*, 1992.

[47] V. Iyengar and P. Kabal. A low-delay 16 kb/s speech coder. In *IEEE Intl. Conf. Accoustic, Speech and Signal Processing*, April 1988.

[48] N. S. Jayant and S. W. Christensen. Effects of packet losses in waveform coded speech and improvements due to an odd-even sample-interpolation procedure. *IEEE Trans. on Communications*, COM-29(2), February 1981.

[49] N. S. Jayant and P. Noll. *Digital Coding of Waveforms: Principles and Applications to Speech and Video.* Prentice-Hall, 1984.

[50] J. P. Campbell Jr, T. E. Tremain, and V. C. Welch. The DOD 4.8 kbps standard (proposed federal standard 1016). In B. S. Atal, V. Cuperman, and A. Gersho, editors, *Advances in Speech Coding.* Kluwer Academic, 1991.

[51] B-H. Juang and A. H. Gray Jnr. Multiple stage vector quantization for speech coding. In *IEEE Intl. Conf. Accoustics, Speech and Signal Processing*, May 1982.

[52] P. Kabal and R. P. Ramachandran. The computation of line spectral frequencies using Chebyshev polynomials. *IEEE Trans. Accoustics, Speech and Signal Processing*, ASSP-34(6), December 1986.

[53] M. R. Karim. Packetizing voice for mobile radio. *IEEE Trans. on Communications*, 42(2), February 1994.

[54] A. Kataoka and T. Moriya. A backward adaptive 8 kb/s speech coder using conditional pitch prediction. In *IEEE Intl. Conf. Globecom*, 1991.

[55] N. Kitawaki, H. Nagabuchi, M. Taka, and K. Takahashi. Speech coding technology for ATM networks. *IEEE Communications Magazine*, 21(1), January 1990.

[56] W. B. Kleijn. Encoding speech using prototype waveforms. *IEEE Trans. Accoustic, Speech and Signal Processing*, 1(4), October 1993.

[57] W. B. Kleijn, D. J. Krasinski, and R. H. Ketchum. Improved speech quality and efficient vector quantization in SELP. In *Intl. Conf. Accoustic, Speech and Signal Processing*, April 1988.

[58] K. Kondo and M. Ohno. Packet speech transmission on ATM networks using a variable rate embedded ADPCM coding scheme. *IEICE Trans. Communications*, E76-R(4), April 1993.

[59] P. Kroon and B. S. Atal. Pitch predictors with high temporal resolution. In *IEEE Intl. Conf. Accoustics, Speech and Signal Processing*, 1990.

[60] R. Kubichek. Technology-independent, user oriented, objective classification of voice transmission quality (study project T1Y1-20). Draft technical report, National Telecommunications and Information Administration, 1992.

[61] M. M. Lara-Barron and G. B. Lockhart. Packet-based embedded encoding for transmission of low bit-rate-encoded speech in packet networks. *IEE Proceedings-I*, 139(5), October 1992.

[62] W. P. LeBlanc. *Speech Coding at Low to Medium Bit Rates.* PhD thesis, Carleton University, October 1992.

[63] W. C. Y. Lee. *Mobile Communications Engineering.* McGraw-Hill, 1987.

[64] T. W. Leung, W. P. Blanc, and S. A. Mahmoud. Speech coding over frame relay networks. In *IEEE Intl. Workshop on Speech Coding*, October 1993.

[65] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Trans. Communications*, COM-28(1), January 1980.

[66] S. P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Information Theory*, IT-28, March 1982.

[67] J. Makhoul, S. Roucos, and H. Gish. Vector quantization in speech coding. *Proceedings of the IEEE*, 73(11), November 1995.

[68] M. W. Marcellin and T.R. Fischer. A trellis-searched 16 kb/s speech coder with low delay. In B. S. Atal, V. Cuperman, and A. Gersho, editors, *Advances in Speech Coding*. Kluwer Academic, 1991.

[69] J. D. Markel and A. H. Gray. Jnr. *Linear Prediction of Speech*. Springer-Verlag, 1976.

[70] F. A. Marvasti. Fast packet networks: Data, image, and voice signal recovery. In F. E. Froehlich and A. Kent, editors, *Encyclopedia of Telecommunications*. Marcel Dekker, 1994.

[71] J. Max. Quantizing for minimum distortion. *IRE Trans. Information Theory*, IT-6, March 1960.

[72] R. J. McAulay and T. F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Trans. Accoustic, Speech and Signal Processing*, 34(4), August 1986.

[73] S. Miki, K. Mano, T. Moriya, K. Oguchi, and H. Ohmuro. A pitch synchronous innovation CELP (PSI-CELP) coder for 2-4 kb/s. In *IEEE Intl. Conf. Accoustic, Speech and Signal Processing*, April 1994.

[74] T. B. Minde, H. Hermansson, E. Ekudden, T. Frankkila, I. Johansson, P. Mustel, J. Nystrom, J. Svedberg, Y. Timner, and A. Uvliden. An enhanced full-rate speech coder for GSM and PCS1900. In *IEEE Intl. Workshop on Speech Coding*, September 1995.

[75] S. E. Minzer. Broadband ISDN and asynchronous transfer mode (ATM). *IEEE Communications Magazine*, September 1989.

[76] N. Morrison. *Introduction to Sequential Smoothing and Prediction*. McGraw-Hill, 1969.

[77] R. W. Muise, T. J. Schonfeld, and G. H. Zimmerman. Experiments in wideband packet technology. In *Zurich Seminar Digital Communications*, 1986.

[78] H. Nakada and K. Sato. Variable rate speech coding for asynchronous transfer mode. *IEEE Trans. on Communications*, 38(3), March 1990.

[79] E. Paksoy, K. Srinivasan, and A. Gersho. Variable rate speech coding with phonetic segmentation. In *IEEE Intl. Conf. Accoustics, Speech and Signal Processing*, March 1993.

[80] R. Peng and V. Cuperman. Variable-rate low-delay analysis-by-synthesis speech coding at 8-16 kb/s. In *IEEE Intl. Conf. Accoustic, Speech and Signal Processing*, 1991.

[81] R. Pettigrew. Low-delay vector excitation coding of speech at 16 kb/s. Master's thesis, Simon Fraser University, January 1990.

[82] R. Pettigrew and V. Cuperman. Backward pitch prediction for low-delay speech coding. In *IEEE Intl. Conf. Globecom*, 1989.

[83] J. G. Proakis. *Digital Communications*. McGraw Hill, 1995.

[84] L. R. Rabiner, M. J. Cheng, A. E. Rosenberg, and C. A. McGonegal. A comparitive performance study of several pitch detection algorithms. *IEEE Trans. Accoustics, Speech and Signal Processing*, ASSP-24, October 1976.

[85] R. P. Ramachandran and P. Kabal. Stability and performance analysis of pitch filters in speech coders. *IEEE Trans. Accoustics, Speech and Signal Processing*, ASSP-35(7), July 1987.

[86] D. Raychaudhuri and N. D. Wilson. ATM-based transport architecture for multiservices wireless personal communication networks. *IEEE Journal on Selected Areas in Communications*, 12(8), October 1992.

[87] R. C. Reininger and J. D. Gibson. Backward adaptive lattice and transversal predictors in ADPCM. *IEEE Trans. Communications*, COM-33(1), January 1985.

[88] M. J. Sabin and R. M. Gray. Product code vector quantizers for waveform and voice coding. *IEEE Trans. Accoustics, Speech and Signal Processing*, ASSP-32, June 1984.

[89] R. Salami, C. Laflamme, J. P. Adoul, A. Kataoka, S. Hayashi, C. Lamblin, D. Massaloux, S. Proust, P. Kroon, and Y. Shoham. Description of the ITU-T 8 kb/s speech coding standard. In *IEEE Intl. Workshop on Speech Coding*, September 1995.

[90] M. Schwartz. *Telecommunication Networks*. Addison-Wesley, 1987.

[91] Y. Shoham. High-quality speech coding at 2.4 to 4 kb/s based on time-frequency interpolation. In *IEEE Intl. Conf. Accoustics, Speech and Signal Processing*, April 1993.

[92] K. Sriram, R. S. McKinney, and M. H. Sherif. Voice packetization and compression in broadband atm networks. *IEEE Journal on Selected Areas in Communications*, 9(3), April 1991.

[93] P. Strobach. New forms of Levinson and Schur algorithms. *IEEE Signal Processing Magazine*, January 1991.

[94] H. Yu Su and P. Mermelstein. Improving the speech quality of cellular mobile systems under heavy fading. In *IEEE Intl. Conf. Accoustics, Speech and Signal Processing*, March 1992.

[95] J. Suzuki and M. Taka. Missing packet recovery techniques for low-bit-rate coded speech. *IEEE Journal on Selected Areas in Communications*, 7(5), June 1989.

[96] T. E. Tremain and V. C. Welch. A new government standard 2400 bps speech coder. In *IEEE Intl. Workshop on Speech Coding*, 1993.

[97] V. J. Varma. Testing of 8 kb/s speech coders for usage in personal communications systems. Internal technical report, Bellcore, 1992.

[98] D. Veeneman and B. Mazor. Efficient multi-tap pitch prediction for stochastic coding. In *IEEE Intl. Workshop on Speech Coding*, September 1991.

[99] A. J. Viterbi and R. Padovani. Implications of mobile cellular CDMA. *IEEE Communications Magazine*, December 1992.

[100] C. R. Watkins and J. H. Chen. Improving 16 kb/s LD-CELP speech coder for frame erasure channels. In *IEEE Intl. Conf. Accoustics, Speech and Signal Processing*, May 1995.

[101] L. Watts and V. Cuperman. A vector ADPCM analysis-by-synthesis configuration for 16 kb/s speech coding. In *IEEE Intl. Conf. Globecom*, 1988.

[102] C. J. Weinstein and J. W. Forgie. Experience with speech communications in packet networks. *IEEE Journal on Selected Areas in Communications*, SAC-1(6), December 1983.

[103] H. C. Woo and J. D. Gibson. Low delay tree coding of speech at 8 kb/s. In *IEEE Intl. Conf. Globecom*, 1991.

[104] Y. Wu, H. B. Hansen, K. J. Larsen, H. Nielsen, and J. Sorenson. High performance coder: A possible candidate for the GSM half-rate system. In *IEEE Intl. Conf. Accoustic, Speech and Signal Processing*, 1991.

[105] Working Party SG XV/2. 8 kb/s speech coding. CCITT technical report, International Telecommunications Union, 1991.

[106] Working Party SG XV/2. Use of speech coding for FPLMTS. CCITT technical report, International Telecommunications Union, 1992.

[107] Working Party SG XV/2. Report of the march 1993 meeting on 16 kb/s speech coding. CCITT technical report, International Telecommunications Union, 1993.

[108] H. Sakai Y. Iiguni and H. Tokumaru. Convergence properties of simplified gradient adaptive lattice algorithms. *IEEE Trans. Accoustic, Speech and Signal Processing*, ASSP-33(6), December 1985.

[109] J. H. Yao, J. J. Shynk, and A. Gersho. Low-delay vector excitation coding of speech at 8 kb/s. In *IEEE Intl. Conf. Globecom*, 1991.

[110] R. Zopf. Real-time implementation of a variable rate CELP speech coder. Master's thesis, Simon Fraser University, May 1995.