ENCODING PROTOTYPE WAVEFORMS USING A PHASE CODEBOOK MODEL

by

Yingbo Jiang

B.S.E.E. University of Science and Technology of China, 1985

M.S.E.E. University of Science and Technology of China, 1988

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

October 1996

ISBN 0-612-16934-0

PARTIAL COPYRIGHT LICENSE

Title of Thesis/Project/Extended Essay

"Encoding Prototype Waveforms Using a Phase Codebook Model"

Author:

_____
(signature)

Yingbo Jiang
(name)

October 11, 1996
(date)

APPROVAL

Name: Yingbo Jiang

Degree: Doctor of Philosophy

Title of thesis : Encoding Prototype Waveforms Using a Phase Codebook Model

Examining Committee: Dr. Shawn Stapleton, Chairman

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. Peter Kroon
Bell Laboratories, Lucent Technologies

Date Approved: Oct 11, 96

Abstract

In recent years, there has been a renewed explosion of interest and activity in the area of digital speech compression, primarily due to the rapid development of very large scale integrated circuit technology that has enabled cost effective implementation of sophisticated speech compression algorithms. Speech compression has a variety of application such as digital cellular phones, personal communications systems, and multi-media communications.

Code excited linear prediction(CELP) is the dominant waveform speech coder at rates above 4 kbps while vocoder-based speech coders have become more prevalent at lower bit rates. *Interpolative speech coding* or *prototype waveform interpolation* (PWI) has emerged recently as a new class of coders belonging to the gray area between waveform coders and vocoders. Interpolative coding encodes only part of a speech signal, called prototype waveforms, and the missing part is recovered by interpolation between prototype waveforms. Interpolative coding has shown good potential to substantially reduce the bit rate.

This thesis proposes a phase codebook model to quantize phase information of speech prototype waveforms at low bit rates. The most challenging problem in phase quantization is the lack of a meaningful phase distortion criterion. Previous research efforts in phase quantization apply the *minimum mean square error*

(MSE) criterion on phase values in the frequency domain. This criterion results in poor phase quantization. In the proposed model, the minimum mean square error MSE criterion is applied to prototype waveforms in the time domain. The spectral phase of prototype waveforms is separated completely from the spectral magnitude, and quantized using a phase codebook. The model can perform closed-loop waveform alignment, together with the phase codebook search procedure. Experimental results are presented which indicate that the phase codebook model significantly outperforms direct waveform quantization schemes.

The phase codebook model provides an alternative way of prototype waveform quantization and facilitates efficient waveform interpolation. The model has been applied to prototype waveform interpolative coding with good results.

To My Parents

Acknowledgments

I would like to thank my supervisor Dr. Vladimir Cuperman for his guidance throughout the course of this research. I am grateful to Drs. Paul Ho and Jacques Vaisey, as members of my supervisory committee, for their helpful suggestions and comments.

Thanks to my colleagues, especially Bhaskar Bhattacharya, Aamir Husain, Feng-Hua Liu, and Peter Lupini for their friendship and assistance.

Thanks also to Brigitte Rabold, Chao Chen, Susan Stevens, and Marilyn Prouting for their excellent support.

Finally, I would like to thank B. C. Advanced Systems Institute, Simon Fraser University, NSERC, and the Center for Systems Science for providing financial support during my studies.

Contents

List of Symbols

E	Expectation operator
mse	mean square error
$d(x, y)$	distortion
$D(x, y)$	Average distortion
$\mathbf{C}$	Codebook
$\mathbf{Q}$	Quantization mapping
$\mathbf{R^k}$	k-dimensional Euclidean Space
$Cent(\mathbf{R_i^k})$	Centroid of a partition
A_l	Amplitude of the l^{th} sinusoid
A_s	Amplitude spectrum of the system function of speech production
$A(z)$	short-term prediction (inverse) filter
$B(z)$	long-term prediction filter
$H_l(k)$	Interpolation filter
$W(z)$	Perceptual weighting filter
a_k	k-th short-term prediction coefficient
$\epsilon(n_0)$	mean square error between original and reconstructed speech
γ_1	numerator bandwidth expansion coefficient
γ_2	denominator bandwidth expansion coefficient
ϕ_l	the initial phase of the l^{th} sinusoid
$\hat{\phi}(t)$	interpolated phase of sinusoids

ω_l	the harmonic frequency of the l^{th} sinosoid
τ	abstract continuous time axis
$\Theta_s(\omega)$	system phase of speech production function
ξ	best alignment shift in continuous time domain
$x(n)$	input (or original) signal
$\tilde{x}(n)$	quantized or predicted signal
$c(n)$	codevector excitation signal
c_m	cepstral coefficents
$e(n)$	residual error signal (discrete time)
$e(t)$	residual (excitation) signal (continous time)
$y(n)$	output (or reconstructed) signal
$s(n)$	input speech signal
$\hat{s}(n)$	reconstructed speech signal
$\mathbf{r}_x$	autocorrelation function of x (vector form)
$\mathbf{R}_{xx}$	autocorrelation matrix of x
$p(n)$	pitch period at time instant n (discrete time)
$p(t)$	pitch at time instant t (continuous time)
g_l	prototype waveform gain
g	excitation gain
t	continueous time axis
$z(t,\tau)$	instantaneous prototype waveform at time instant t
$u(t,\phi)$	normalized instantaneous prototype waveform t
C_l	l^{th} *consine* coefficient of Fourier representation
D_l	l^{th} *sine* coefficient of Fourier representation
$H_I(\omega)$	ideal interpolator filter's transfer function
$P_\rho(n)$	polyphase filter
$V_n(lM,k)$	DFT coefficients of gain-normalized and oversampled prototype
$V_{na}(lM,k)$	DFT coefficeints of aligned $V_n(lM,k)$

$\mathcal{R}_l(k)$	rapid evolving waveform in the frequency domain
$\mathcal{S}(k)$	slowly evolving waveform in the frequency domain

List of Abbreviations and Acronyms

ADPCM	Adaptive Differential Pulse Code Modulation
A-by-S	Analysis-by-Synthesis
CDMA	Code Division Multiple Access
CELP	Code Excited Linear Prediction
CS-ACELP	Conjugate Structure Algebraic Code Excited Linear Prediction
DFT	Discrete Fourier Transform
DoD	Department of Defense
ETSI	European Telecommunications Standard Institute
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
GLA	Generalized Lloyd Algorithm
GSM	Global System for Mobile Telecommunications
IDFT	Inverse Fourier Transform
IFFT	Inverse Fast Fourier Transform
IMBE	Improved Multi-band Excitation
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
LPC	Linear Prediction Coding
LSF	Line Spectral Frequencies
LSP	Line Spectral Pairs
MBE	Multi-Band Excitation
MSE	Mean Square Error
MS-NSTVQ	Multi-Stage Non-Square Transform Vector Quantization
MSVQ	Multi-Stage Vector Quantization
MOS	Mean Opinion Score
NSTVQ	Non-Square Transform Vector Quantization
PCM	Pulse Code Modulation
PCS	Personal Communications Services
PFA	the Prime Factor Algorithm
PSI-CELP	Pitch Synchronous Innovation Code Excited Linear Prediction
PW	Prototype Waveform
PWI	Prototype Waveform Interpolation
QCELP	Qualcomm Code Excited Linear Prediction
REW	Rapid Evolving Waveform
SEW	Slow Evolving Waveform
SNR	Signal-to-Noise Ratio
SegSNR	Segmental Signal-to-Noise Ratio
STC	Sinusoidal Transform Coder
TDMA	Time Division Multiple Access
TFI	Time-Frequency Interpolation
TIA	Telecommunications Industry Association

VAR-CELP	Variable Rate Code Excited Linear Prediction
VDVQ	Variable Dimension Vector Quantization
VQ	Vector Quantization
VSELP	Vector Sum Excited Linear Prediction
VXC	Vector Excitation Coding
WFTA	Winograd Fourier Transform Algorithm
ZIR	Zero Input Response
ZSR	Zero State Response

List of Tables

List of Figures

Chapter 1

Introduction

Speech compression has been an ongoing area of research for several decades since the vocoder was proposed in 1940s. In the last several years, however, there has been a renewed explosion of interest and activity in this area. This is primarily attributed to the rapid development of very large scaled integrated circuit technology, that has enabled cost effective implementation of sophisticated speech compression algorithms. Speech compression has a variety of application such as digital cellular phones, personal communications systems, and multi-media communications, where conserving either the bandwidth for transmission or media space for storage is required.

Roughly, speech coding algorithms can be classified into two main categories: waveform coders and vocoders. In waveform coders, the waveforms of original speech are approximated as close as possible by the reconstructed speech. On the other hand, vocoders employ human speech production models which do not try to reproduce the detail of speech waveforms, but rather their perceptually important parameters.

The dominant structure for waveform coders has been Code Excited Linear Prediction(CELP) [4]. At rates above 4 kbps, CELP coders have high speech quality. However, as the bit rate drops to 4 kbps or below, CELP speech coders suffer speech quality degradation simply because not enough bits are available to approximate the original speech waveform.

At rates below 4 kbps, vocoder-based speech coders have become more prevalent. An important class of vocoders is based on the sinusoidal model. In this model, the speech waveform is represented as a sum of sinusoids. In particular, sinusoidal transform coding (STC) [24] [42] [40] [43] [49] and multi-band excitation (MBE) [23] [25] coding are both very actively studied versions of sinusoidal coding.

In recent years, another promising approach to substantially reduce the rate for quasi-periodic speech segments is prototype waveform Interpolation (PWI). This is a class of coders belonging to the gray area between waveform coder and vocoder. In interpolative coding, only a part of the original speech, called *prototype waveforms*, is encoded, and the missing part of the speech is recovered by interpolation between known prototype waveforms.

1.1 Thesis Motivation and Contribution

Recent speech coders, using either the sinusoidal model or interpolative coding, bring new challenges to the field of speech coding. One of the challenges common to these coders is the quantization of the phase information of sinusoids at low bit rates.

In some variants of the sinusoidal model such as the original MBE [23], spectral phase information is extracted from the input speech spectrum and quantized.

Because the quantization of phase needs high bit rates, other versions of the sinusoidal model discard the measured phase in order to achieve low bit rates. For example, IMBE [25] encodes the spectral magnitude, discards the measured phase, and uses the predicted phase to reconstruct speech at the decoder. The STC coder [40] recovers phase information from spectral magnitudes using the Hilbert transform Relation while assuming that speech is a minimum phase signal. Interpolative coders have also faced the challenge of quantizing the phase information at low bit rates. PWI [30] encodes phase information implicitly with the spectral magnitudes, while *time frequency interpolation* (TFI) [54] quantizes the spectral magnitudes only.

This thesis proposes a phase codebook model to quantize phase information of speech prototype waveforms at low bit rates. The most challenging problem in phase quantization is the lack of a meaningful phase distortion criterion. Previous research efforts in phase quantization apply the minimum mean square error (MSE) criterion on phase values in the frequency domain. This criterion results in poor phase quantization. In the proposed model, the minimum mean square error (MSE) criterion is applied to prototype waveforms in the time domain. The spectral phase of prototype waveforms is separated completely from the spectral magnitude, and quantized using a phase codebook. The model can perform closed-loop waveform alignment together with the phase codebook search procedure. Experimental results are presented which indicate that the phase codebook model significantly outperforms direct waveform quantization schemes.

The phase codebook model provides an alternative way of prototype waveform quantization, and facilitates efficient waveform interpolation. The model has been applied to prototype waveform interpolative coding with good results.

1.2 Thesis Organization

This thesis is organized into six chapters. Chapter 2 provides a brief overview of the fundamentals of speech coding together with a few dominant speech coding algorithms. Chapter 3 presents an overview of the emerging interpolative coding which is necessary to facilitate a better understanding of Chapters 4 and 5.

In Chapter 4, a new phase codebook model is proposed and a detailed derivation of the model is presented. Different interpolation techniques are compared and used in the phase codebook model to significantly reduce computational complexity. The model is validated by comparing its performance with two different reference systems.

In Chapter 5, the phase codebook model is applied to prototype waveform interpolation coding. An efficient interpolation scheme of prototype waveforms based on the phase codebook model is presented. a speech coder based on interpolative coding is developed. Subjective listening results are also presented. Finally, conclusions are presented in Chapter 6.

Chapter 2

Fundamentals of Speech Coding

In this chapter, we will first review the concepts of linear prediction and vector quantization. Most of the current speech coders are based on these concepts. Second, we will discuss some of the current dominant speech coding algorithms, namely code-excited linear prediction (CELP), sinusoidal transform coding (STC), and multiband excitation coding (MBE). The description of these speech coders emphasizes the concepts and problems which are relevant to the proposed phase codebook model covered in chapter 4 and the speech coders covered in chapter 5. A comprehensive review of the state-of-the-art of speech coding can be found in [20]. Another emerging class of speech coding algorithms, interpolative coding, will be reviewed in more detail in the next chapter with the goal to introduce the concepts and notations that are important in later chapters.

2.1 Linear Prediction of Speech

In this section, the concept of linear prediction is briefly introduced. More detail can be found in [37].

Linear prediction, under various names and formulations, is widely applied in many fields. The first researchers to directly apply linear prediction techniques to speech analysis and synthesis were Saito and Itakura, and Atal and Schroeder in 1960s.

In a speech signal, the linear prediction of the current sample $s(n)$ is obtained as a linear combination of the past samples,

$$\hat{s}(n) = -\sum_{k=1}^{M} a_k s(n-k). \tag{2.1}$$

where, a_k are the linear prediction coefficients and M is the order of the linear predictor. The prediction error is defined as

$$e(n) = s(n) - \hat{s}(n) = s(n) + \sum_{k=1}^{M} a_k s(n-k) \tag{2.2}$$

The optimal set of a_k are chosen by minimizing the variance of the prediction error under the assumption that $s(n)$ is stationary and zero-mean. The variance can be written as

$$\sigma_e^2 = E\{e(n)^2\} = E\{(s(n) - \hat{s}(n))^2\} \tag{2.3}$$

Taking the derivative of eqn. (2.3) with respect to a_k, we obtain the following *Yule-Walker equations*

$$\sum_{k=1}^{M} a_k r_{|k-j|} = -r_j \quad j = 1, 2, \ldots, M \tag{2.4}$$

where r_k is the autocorrelation function of $s(n)$. In a matrix form, eqn. (2.4) can be written as

$$\mathbf{R_{ss}a} = -\mathbf{r_s} \tag{2.5}$$

where $\mathbf{R}_{ss}$ is the autocorrelation matrix of $s(n)$, $\mathbf{a} = [a_1, a_2, \ldots, a_M]^T$, is the linear prediction coefficient vector and $\mathbf{r}_s = [r_1, r_2, \ldots, r_M]^T$.

$$\mathbf{R}_{ss} = \begin{bmatrix} r_0 & r_1 & \cdots & r_{M-1} \\ r_1 & r_0 & \cdots & r_{M-2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{M-1} & r_{M-2} & \cdots & r_0 \end{bmatrix} \tag{2.6}$$

If the matrix $\mathbf{R}_{ss}$ is non-singular, then the optimal linear prediction coefficients are given by

$$\mathbf{a} = -\mathbf{R}_{ss}^{-1}\mathbf{r}_s \tag{2.7}$$

The autocorrelation matrix of a stationary sequence has a Toeplitz structure, hence efficient coefficient estimation algorithms such as the Levinson-Durbin algorithm [37] can be used.

In the frequency domain, the linear prediction model has the following transfer function

$$A(z) = \sum_{k=0}^{M} a_k z^{-k} \tag{2.8}$$

In speech coding, the prediction error signal $e(n)$ is called the *residual* signal. The relation between the residual and speech signals is illustrated in figure 2.1. The residual signal is obtained by passing the original speech signal $s(n)$ through filter $A(z)$. On the other hand, original speech $s(n)$ can be reconstructed by passing the residual signal $e(n)$ through filter $1/A(z)$. Hence the filter is known as the synthesis filter. This filter is the basis of many speech coding algorithms such as Code-Excited Linear Prediction (CELP).

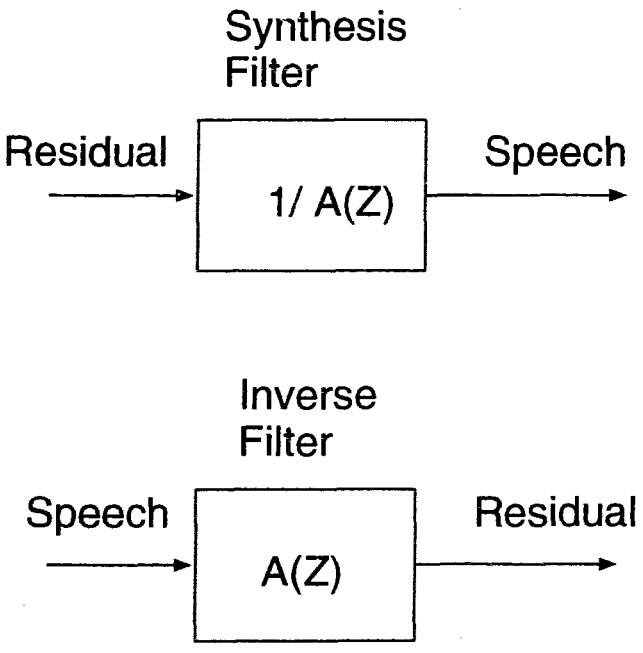

Figure 2.1: $A(z)$ in the Frequency Domain

2.2 Vector Quantization

In this section, we discuss briefly the concept of *vector quantization* (VQ). More detail can be found in [21], [22], and [1].

A fundamental result of Shannon's information theory is that better performance can always be achieved by coding vectors instead of scalars, even if the data source is memoryless. In other words, even if the scalars have been produced by preprocessing the original input data so as to make them uncorrelated or independent (for example, the Karhunen-Loeve transform), better performance is still achievable by vector quantization.

Vector quantization can be viewed as a form of pattern recognition where an input pattern is " approximated" by one of a predetermined set of standard patterns, or in other words, the input pattern is matched with one of a stored set of templates

or *codewords.*

If the input signal is a k-dimensional vector $\mathbf{x}$, a vector quantizer $\mathbf{Q}$ of dimension k and size N is a mapping from $\mathbf{x}$ in k-dimensional Euclidean space, $\mathbf{R^k}$, into a finite set $\mathbf{C}$, where $\mathbf{C} = \{\mathbf{y_j} : j = 1, 2, \ldots, N\}$, and $\mathbf{y_j} \in \mathbf{R^k}$. Each vector $\mathbf{y_j}$ in $\mathbf{C}$ is called a *codevector* or *codeword*, and $\mathbf{C}$ is called the *codebook.*

The mapping decisions partition $\mathbf{R^k}$ into N regions, $\mathbf{R_i^k}, i = 1, 2, \ldots, N$, called *cells.* The i^{th} cell is the subspace of $\mathbf{R^k}$ containing all vectors which are mapped by $\mathbf{Q}$ into $\mathbf{y_i}$

$$\mathbf{R_i^k} = \{\mathbf{x} \in \mathbf{R^k} : \mathbf{Q(x)} = \mathbf{y_i}\} \tag{2.9}$$

A *distortion measure*, $d(\mathbf{x}, \mathbf{Q(x)})$, is used to measure the distortion or error due to the mapping (quantization). A common distortion criterion is the Euclidean distance

$$d(\mathbf{x}, \mathbf{Q(x)}) = \mathbf{d(x, y_j)} = (\sum_{\mathbf{i=1}}^{\mathbf{k}} |\mathbf{x[i]} - \mathbf{y_j[i]}|^{\mathbf{2}})^{\frac{\mathbf{1}}{\mathbf{2}}} \tag{2.10}$$

For a given codebook, it can be shown that the *nearest neighbor* partition rule is optimal in the sense that the average distortion between the unquantized and quantized vectors is minimized [21]. The partition cell $\mathbf{R_i^k}$ using the nearest neighbor rule is defined as

$$\mathbf{R_i^k} = \{\mathbf{x} : d(\mathbf{x}, \mathbf{y_i}) \leq d(\mathbf{x}, \mathbf{y_j}), j = 1, 2, \ldots, N\} \tag{2.11}$$

On the other hand, If the partitions $\mathbf{R_i^k}$ are given, it can be shown that the optimal codeword $\mathbf{y_i}$ for each partition $\mathbf{R_i^k}$ is the *centroid* of $\mathbf{R_i^k}$, denoted by $Cent(\mathbf{R_i^k})$ [21]. The centroid of defined as

$$\mathbf{y_i} = Cent(\mathbf{R_i^k}) \text{ if } E[d(\mathbf{x}, \mathbf{y_i}) | \mathbf{x} \in \mathbf{R_i^k}] \leq E[d(\mathbf{x}, \mathbf{y}) | \mathbf{x} \in \mathbf{R_i^k}] \text{ for } \mathbf{y} \in \mathbf{R_i^k} \tag{2.12}$$

Eqns (2.11) and (2.12) represent necessary conditions for codebook optimality.

The LBG algorithm [34] utilizes the above two optimality conditions to design VQ codebooks by minimizing the average error over a training data set. The algorithm is similar to the *K-means algorithm* in pattern recognition and is also a generalization of Lloyd's algorithm for scalar quantization. The iterative *generalized Lloyd algorithm* (GLA) can be summarized as the following steps:

1. Given a training sequence $\mathbf{x_i}, i = 1, 2, \ldots, M$, and an initial codebook $\mathbf{C}(0)$. Set iteration counter $q = 0$.

2. For a given codebook $\mathbf{C}(q)$, partition $\mathbf{x}_i, i = 1, 2, \ldots, M$ into N celis, $\mathbf{R_j^k}(q), j = 1, 2, \ldots, N$, using eqn.(2.11).

3. For each partition cell $\mathbf{R_j^k}(q)$, compute $Cent(\mathbf{R_j^k}(q))$ using eqn. (2.12), and update the codewords of the codebook $\mathbf{C}(q)$ with the centroids to obtain a new codebook $\mathbf{C}(q+1)$.

4. Calculate the overall distortion

$$D(q+1) = \sum_{j=1}^{N} \sum_{\mathbf{x}_i \in \mathbf{R_j^k}} d(\mathbf{x}_i, \mathbf{y}_j(q+1)) \tag{2.13}$$

if $(D(q+1) - D(q))/D(q) < \delta$ then stop. Otherwise set q=q+1 and go to *Step 2.*

The average distortion produced by GLA-designed codebooks converges only to a local minimum; a different initial codebook can result in a different final codebook.

2.2.1 Sub-Optimal Vector Quantization

The *rate* of a vector quantizer in bit rates per sample is given by $r = (\log_2 N)/k$, where N is the codebook size, and k is the vector dimension. If we fix the rate r, N increases exponentially as vector dimension k increases. This also means the computational complexity increases exponentially.

In practice, due to limitations in complexity and/or available storage, sub-optimal VQ with multiple codebooks is often used instead of the optimal VQ with a large single codebook. A significant reduction in the computational complexity of VQ can be achieved by using codebooks with structures amenable to fast search procedures. There are many structured sub-optimal VQ schemes. Two of these schemes, namely, *Split VQ* and *multi-stage VQ*, are used the later chapters of this thesis.

- *Multi-stage VQ* divides the quantization procedure into successive stages, where the first stage performs a coarse quantization of the input vector. Then, a second stage quantizer quantizes the differential error between the original vector and the first stage quantization output. A similar procedure is repeated for each stage. The reconstructed vector is the sum of codevectors from these different stages.

- *Split VQ* divides the input vector into multiple subvectors, And each subvector is quantized by its own codebook. The reconstructed vector is the concatenation of reconstructed subvectors.

More details of other suboptimal VQ schemes can be found in [21].

2.3 Code-Excited Linear Prediction

Starting from this section to the end of the chapter, we discuss three important classes of speech coders.

One of the most important speech coding algorithms in use today is *code-excited linear prediction* (CELP). Currently, CELP based speech coding systems have provided a basis for most speech coding standardization activity, though recently, vocoder type models are playing an increasing role in evolving standards for the future.

The original CELP was proposed by Atal and Schroeder [4]. Since then, numerous advances to CELP coding have been developed to reduce complexity, increase robustness to channel errors, and improve quality. A comprehensive review of CELP coder literatures can be found in [20]. In this section, we will focus on the linear-prediction-based analysis-by-synthesis approach, which is the basis of CELP coders.

2.3.1 Analysis-by-Synthesis

The analysis-by-synthesis model used in CELP is shown in figure 2.2. The upper part of the figure is the synthesis part, and speech analysis is performed by synthesizing the waveforms for different codebook entries and comparing to the original. In other words, the synthesis procedure is part of the analysis procedure.

In the synthesis part, the synthesized speech $\hat{s}(n)$is reconstructed by passing a code vector c_j from the excitation codebook through the long-term prediction filter $1/B(z)$ and the short term prediction filter $1/A(z)$. The long-term filter $1/B(z)$ models the quasi-periodicity of voiced speech, and the short-term filter $1/A(z)$, called

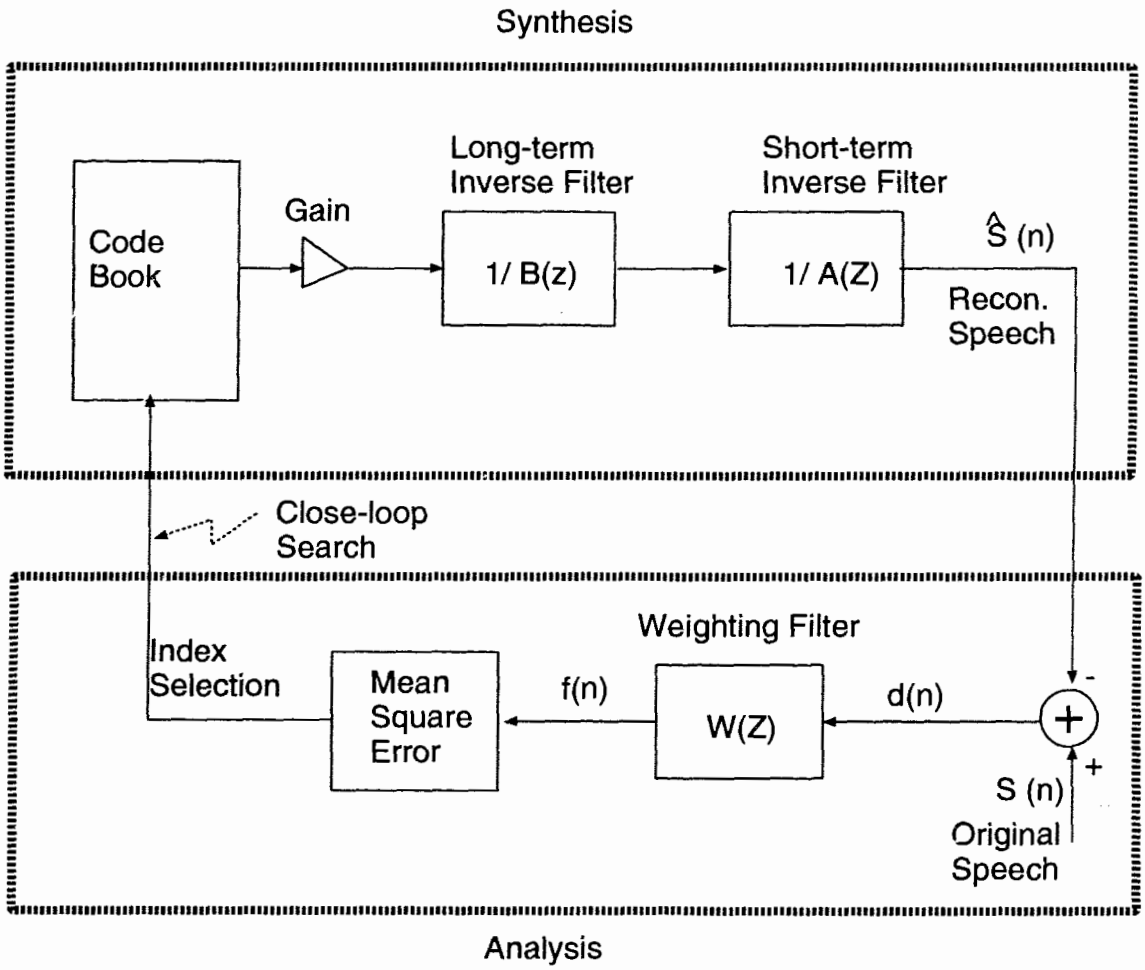

Figure 2.2: Block Diagram of Analysis-by-Synthesis CELP Coder

the synthesis filter, is the inverse of linear prediction filter $A(z)$ discussed in section 2.1. The coefficients a_i of $A(z)$ are the linear prediction coefficients. $1/A(z)$ attempts to model the voice production procedure in the human vocal track.

After the synthesis part obtains a possible synthesized speech signal by passing an excitation codevector through $1/B(z)$ and $1/A(z)$, the the synthesized speech $\hat{s}(n)$ is subtracted from the original to provide a differential signal $d(n)$. Then $d(n)$ is passed through the weighting filter $W(z)$ to provide a weighted differential signal $f(n)$. Finally a mean square error (MSE) is obtained by the summation of squared $f(n)$ in a given block (frame or subframe).

By repeating the above procedure, we obtain a MSE for each codevector of the excitation codebook. The codebook entry that generates the minimum MSE is chosen and the index of the entry is transmitted to the decoder. This completes the codebook search procedure. This encoding procedure is called *analysis-by-synthesis* because the speech analysis is performed by synthesizing—the encoder includes a complete decoder. Embedded in the A-by-S procedure, is the concept of the *closed-loop search* for the best entry in the excitation codebook. In CELP, A-by-S and closed-loop search significantly improve the quality of the reconstructed speech. A disadvantage of A-by-S is a significant increase in the computational complexity.

The weighting filter $W(z)$ in the diagram is defined as

$$W(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)} \tag{2.14}$$

where γ_1 and γ_2 are constants between 0 and 1. The function of the weighting filter is to exploit the masking properties of the human auditory system. The human ear is more sensitive to noise frequency components that lie in the valleys between formants, than to components in close proximity to the formants. The perceptual weighting is to attenuate the frequency components of the quantization noise in these valleys, thus giving better perceptual speech quality.

2.3.2 Complexity Reduction in Closed-Loop Search

The original CELP description shown in figure 2.2 only presents the basic conceptual idea, and a direct implementation of this diagram results in very high complexity. In order to efficiently handle the closed-loop search operation, usually the filtering in figure 2.2 is decomposed into zero-input response (ZIR) and zero-state response (ZSR) as shown in figure 2.3 [16].

The output of the synthesis filter can be written as the sum of zero state response (ZSR) and the zero input response (ZIR) by using the linearity property of the synthesis filter. The ZIR does not depend on the choice of the code vector in the excitation codebook. The ZSR is the output of the filter with memory set to zero, and it does not depend on the excitation vector of previous frames. For the long-term predictor, the same decomposition can be applied if the pitch period is greater than the processed block size (frame or subframe). This procedure greatly simplifies the codebook search procedure.

Another technique to reduce the computational complexity of the search procedure is to move the weighting filter into branches as shown in figure 2.3. The weighting filter now can be combined with the synthesis filter to form *the weighted synthesis filter*, resulting in reduced filtering operations.

2.3.3 CELP based Speech Coding Standards

Since the original CELP was proposed, the number of studies of CELP coding algorithms has grown steadily. Numerous techniques for reducing computational complexity and enhancing the performance of CELP coders have been emerged. As a results, a number of CELP based speech coders have been adopted as international

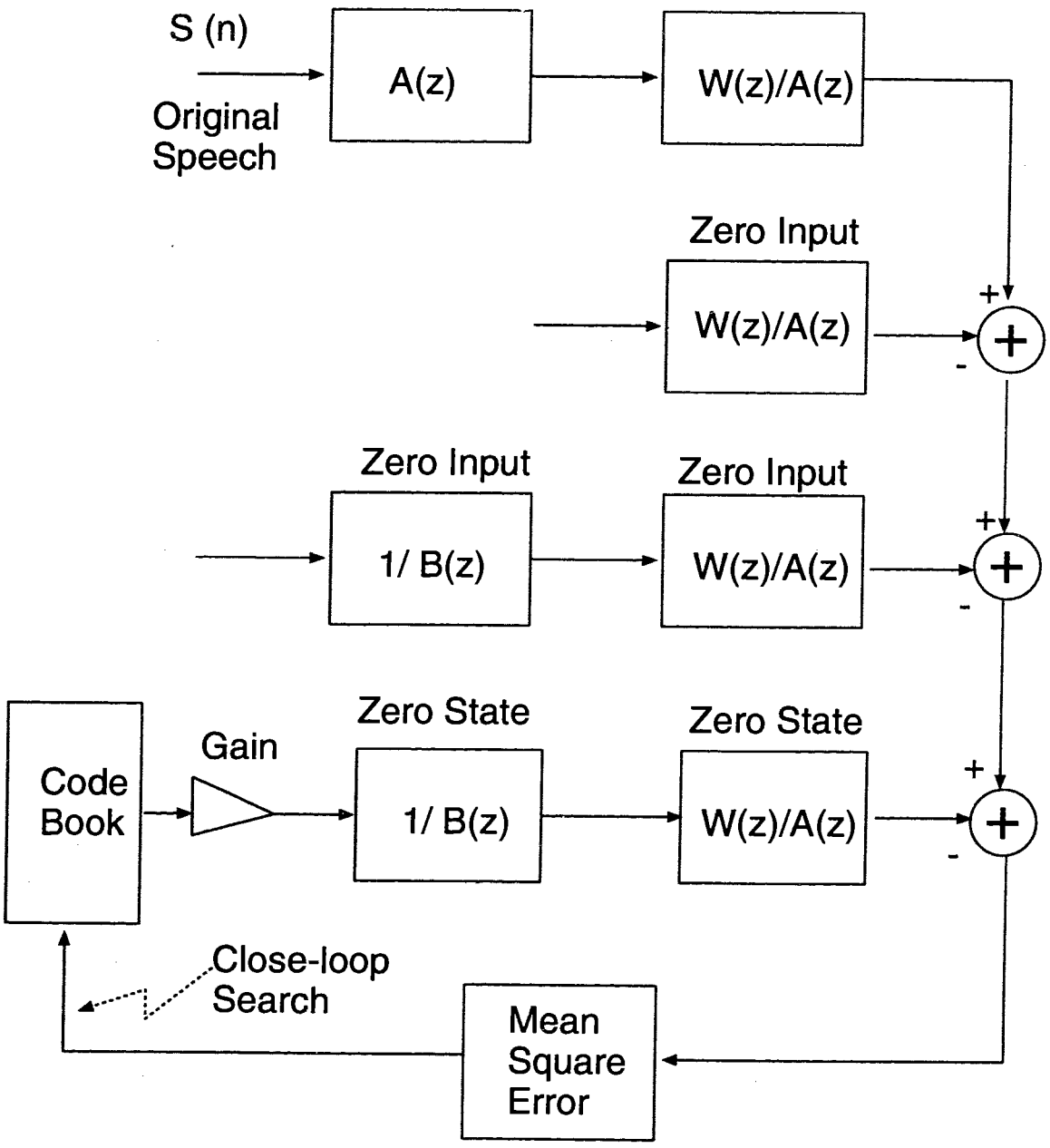

Figure 2.3: CELP Structured Rearranged (1) separate the zero state and zero input responses (2) move location of $W(z)$

standards. These CELP based standards have bit rates from 4.6 kbps to 16 kbps.

The first CELP based standard is the Department of Defense FS 1016 CELP codec [29]. FS 1016 operates at an encoding rate of 4.6kbps, with an extra 200 b/s set aside for synchronization, error correction, and future algorithm modifications. A main feature of this CELP coder is a sparse stochastic codebook, which enables complexity reduction in the stochastic codebook search procedure.

Another important CELP based coder is the *vector sum excitation linear prediction* (VSELP) by Gerson and Jasiuk. It is been adopted as a standard for North American TDMA digital cellular telephony (IS-54). A modified version of it is also used for the Japanese Digital Cellular (JDC) TDMA standard. The JDC has adopted a half-rate standard for the Japanese TDMA digital cellular system called *pitch synchronous innovation CELP* (PSI-CELP) [44].

At 16 kbps, ITU-T adopted the *low-delay CELP algorithm* developed by Chen et al, [12] [13] as a standard (G.728). A variable rate coding algorithm called Qualcomm CELP (QCELP) was also chosen as the CDMA digital cellular telephony standard IS-95.

The most recent ITU standard at 8 kbps (G.729) is based on *conjugate structured algebraic CELP (CS-ACELP)* [52].

2.4 Sinusoidal Transform Coding

Sinusoidal coders have emerged as an important class of vocoders in recent years, and represent a viable alternative to CELP at rates below 4 kbps. The main feature of sinusoidal coders is that reconstructed voiced speech is, at the decoder, generated

as a sum of sinusoids. The frequencies and phases of these sinusoids try to represent and track the evolving short-term spectral character of original speech.

The conceptual introduction of this approach is due to Hedelin [24]. Since then, several variants of sinusoidal coding have been studied such as *sinusoidal transform coding* (STC) [42] [40] [43] [49] , *multiband excitation coding* (MBE) [23] [25], and *harmonic coding* [3] [38]. In this section, we describe the STC developed by McAulay and Quatieri. (MBE will be described in the next section). We will focus only on issues which are relevant to this thesis such as synthesis models and phase models for sinusoidal coding. The reader is referred to [40] for a complete discussion of STC.

2.4.1 Speech Synthesis in STC

Given an estimated set of magnitudes, frequencies, and phases for each frame, a straightforward synthesis model used in sinusoidal coding for the k^{th} frame of speech is

$$\hat{s}(n) = \sum_{l=1}^{L} A_l^k \cos(n\omega_l^k + \phi_l^k) \tag{2.15}$$

where L is the number of sinusoids used for synthesis in the kth frame, A_l^k and ω_l^k specify the amplitude and frequency of the l^{th} sinusoidal oscillator, and ϕ_l^k specifies the initial phase of each sinusoid. However, if each of consecutive frames is synthesized using eqn. (2.15) without interpolation between parameters of neighboring frames, this scheme will generate discontinuity at frame boundaries. To solve this problem, we have to interpolate the parameters of the current frame, $\{A_l^k, \omega_l^k, \phi_l^k\}$, with those that are obtained on the previous frame,$\{A_l^{k-1}, \omega_l^{k-1}, \phi_l^{k-1}\}$.

The magnitudes can be easily interpolated as following

$$\hat{A}^k(n) = A^{k-1} + (A^k - A^{k-1})(n/T) \;\; n = 0, 1, \ldots, T-1 \tag{2.16}$$

where T is the frame length, and subscript l is dropped for convenience.

However, interpolation of the frequencies and phases is more complicated because the phase ϕ^k and ϕ^{k-1} are obtained modulo 2π. Hence, phase unwrapping is required to ensure that the frequency tracks are "maximally smooth" across frame boundaries. A cubic polynomial for phase interpolation is required [2]:

$$\hat{\phi}(t) = \zeta + \gamma t + \alpha t^2 + \beta t^3 + 2\pi M \tag{2.17}$$

where the term $2\pi M$,M an integer, accounts for the phase unwrapping. To simplify the following derivations, assume a continuous variable t with $t = 0$ corresponding to the center of frame $k-1$ and $t = T$ corresponding to the center of frame k. Since the starting phase and frequency of the interpolation are known at $t = 0$, (ϕ^{k-1} and ω^{k-1}), eqn. (2.17) can be rewritten as

$$\hat{\phi}(t) = \phi^{k-1} + \omega^{k-1}t + \alpha t^2 + \beta t^3 + 2\pi M \tag{2.18}$$

Now by writing eqn. (2.18) and its derivative for the phase and frequency of frame k, ϕ^k and ω^k, to eqn.(2.18), it can be shown that values of α and β have to satisfy the relations

$$\begin{bmatrix} \alpha(M) \\ \beta(M) \end{bmatrix} = \begin{bmatrix} 3/T^3 & -1/T \\ -2/T^3 & 1/T^2 \end{bmatrix} \begin{bmatrix} \phi^k - \phi^{k-1} - \omega^{k-1}T + 2\pi M \\ \omega^k - \omega^{k-1} \end{bmatrix} \tag{2.19}$$

The phase unwrapping parameter M is chosen to make the unwrapped phase maximally "smooth". A reasonable criterion for "smoothness" is to minimize the second derivative of $\hat{\phi}(t)$ with respect to time t. After some derivation, M is found to be [42]:

$$M = round\left(\frac{1}{2\pi}\left[(\phi^{k-1} + \omega^{k-1}T - \phi^k) + (\omega^k - \omega^{k-1})\frac{T}{2}\right]\right) \tag{2.20}$$

where *round* means taking the closest integer.

As a result of this phase unwrapping procedure, each frequency track will have associated with it an instantaneous unwrapped phase. The synthesized waveform

for the kth frame will be given by

$$\hat{s}(n) = \sum_{l=1}^{L} \hat{A}_l(n) \cos(\hat{\phi}_l(n)) \tag{2.21}$$

From eqns. (2.15) and (2.21), the parameter set required for sinusoidal coding is $\{A_l^k, \omega_l^k, \phi_l^k\}$, where $1 \leq l \leq L$. This parameter set is exceedingly large for speech coding applications at 4 kbps and below. One simple modification to the synthesis model for reducing the required parameter set is to assume that the frequencies of the sinusoids for a given frame are integer multiples of the lowest frequency (called the *fundamental* or *pitch* frequency). In this case, it is not necessary to transmit the number of sinusoids, L, or the frequency of each sinusoid, ω_l. Instead only the fundamental frequency ω_0 is transmitted. However, magnitudes A_l^k, and the phases ϕ_l^k $1 \leq l \leq L$ still have to be transmitted. In order to avoid direct transmission of the phases that requires high bit rates, McAulay and Quatieri have developed a sine wave phase model with more efficient parametric presentation, which will be described in the following subsection.

2.4.2 Phase Model in STC

The sinusoidal speech synthesis model [40] explicitly identifies the phase components due to the excitation, the glottis, and the vocal tract.

For a basic vocoder model, the sequence of excitation pitch pulses can be written as a sum of sine waves (in the complex form)

$$e(n) = \sum_{l=1}^{L} e^{j(n-n_0)\omega_l} \tag{2.22}$$

where n_0 corresponding the time of occurrence of the pitch pulse nearest the center of the current analysis frame. The occurrence of this temporal event, called the

onset time, ensures that the underlying excitation sine waves will add up with each other at the time of pitch pulse. Assume $H_g(\omega)$, $H_v(\omega)$ are the transfer functions of the glottal pulse and the vocal tract respectively, and let $H_s(\omega) = H_g(\omega)H_v(\omega)$ be the composite transfer function, called *the system function.* Passing the excitation sequence in eqn. (2.22) through the filters, we have the output speech signal in the complex form

$$\hat{s}(n) = \sum_{l=1}^{L} H_s(\omega_l) e^{j(n-n_0)\omega_l} \tag{2.23}$$

On the other hand, original speech can be represented in terms of sine waves as

$$s(n) = \sum_{l=1}^{L} A_l e^{j(n\omega_l + \theta_l)} \tag{2.24}$$

Ideally, $\hat{s}(n)$ should be as close as possible to $s(n)$ according to some meaningful criteria. A reasonable criterion is the minimum mean squared error (MSE)

$$\epsilon(n_0) = \frac{1}{N+1} \sum_{n=-N/2}^{N/2} |s(n) - \hat{s}(n; n_0)|^2 \tag{2.25}$$

where n_0 is the onset time. The eqn. (2.25) can be expanded as

$$\epsilon(n_0) = \frac{1}{N+1} \sum_{n=-N/2}^{N/2} \{|s(n)|^2 - 2\mathrm{Re}\,[s(n)\hat{s}^*(n; n_0)] + |\hat{s}(n; n_0)|^2\} \tag{2.26}$$

The first term of eqn. (2.26) is the power in the measured signal, which can be defined as

$$P_s = \frac{1}{N+1} \sum_{n=-N/2}^{N/2} |s(n)|^2 = \sum_{l=1}^{L} A_l^2 \tag{2.27}$$

The second term of eqn. (2.26) can be rewritten using the spectral magnitudes:

$$\frac{1}{N+1} \sum_{n=-N/2}^{N/2} s(n)\hat{s}^*(n; n_0) = \sum_{l=1}^{L} A_l A_s(\omega_l) e^{j[\theta_l + n_0\omega_l - \Theta_s(\omega_l)]} \tag{2.28}$$

where $A_s(\omega)$ and $\Theta_s(\omega)$ are the magnitude and phase of the system transfer function $H_s(\omega)$ in eqn. (2.23). Similar to the first term, the third term of eqn. (2.26) is the power of synthesized speech defined as

$$\frac{1}{N+1} \sum_{n=-N/2}^{N/2} |\hat{s}(n; n_0)|^2 = \sum_{l=1}^{L} A_s^2(\omega_l) \tag{2.29}$$

Using eqns. (2.27)–(2.29), we can rewrite eqn. (2.25) as

$$\epsilon(n) = P_s - 2\sum_{l=1}^{L} A_l A_s(\omega_l)\cos[\theta_l + n_0\omega_l - \Theta_s(\omega_l)] + \sum_{l=1}^{L} A_s^2(\omega_l) \tag{2.30}$$

Eqn. (2.30) was obtained assuming that the system magnitude $A_s(\omega)$ and phase $\Theta_s(\omega)$ were known. To obtain the best estimation of onset n_0, the system magnitude and phase have to be estimated first. One estimator for the magnitude of the system function can simply be obtained by linear interpolation between the available magnitudes of the current frame defined as following

$$A_s(\omega) = A_{l-1} + \frac{A_l - A_{l-1}}{\omega_l - \omega_{l-1}}(\omega - \omega_{l-1}) \;\; \omega_{l-1} \leq \omega \leq \omega_l \tag{2.31}$$

Because the measured (or original) phase requires high bit rates to quantize, alternatives have to be used to recover the phase. One early method is to simply set the system phase $\Theta_s(\omega)$ to zero, then the optimal onset time n_0 can be estimated from eqn. (2.30) [49] [41] [48]. Another way of recovering the system phase is to assume that the system function is of minimum phase. Then the phase can be recovered from the system magnitude using the Hilbert transform [46] as shown below.

From the system magnitude $A_s(\omega)$, the cepstral coefficients $c_m, m = 0, 1, \ldots$ can be determined using

$$c_m = \frac{1}{\pi}\int_0^{\pi} \log A_s(\omega)\cos(\omega m)d\omega \;\; m = 0, 1, \ldots \tag{2.32}$$

and, hence, the system phase, $\Theta_s(\omega)$ can be obtained using

$$\Theta_s(\omega) \;\; = \;\; -2\sum_{m=1}^{\infty} c_m \sin(m\omega) \tag{2.33}$$

Use of eqn. (2.33) is incomplete, however, since the minimum phase analysis brings an ambiguity to the system phase $\Theta_s(\omega)$. The ambiguity can be accounted for by

generalizing the phase in eqn. (2.33) using

$$\Theta_s(\omega) + \beta\pi \quad \beta = 0 \text{ or } 1 \tag{2.34}$$

Applying the expressions for the system amplitude and phase in eqn. (2.31) and eqn. (2.34) in eqn.(2.30), the MSE is

$$\epsilon(n_0, \beta) = P_s - 2\sum_{l=1}^{L} A_l^2 \cos[\theta_l + n_0\omega_l - \beta\pi - \Theta_s(\omega_l)] + \sum_{l=1}^{L} A_l^2 \tag{2.35}$$

because only the second term of eqn. (2.35) is relevant to the phase model, it suffices to choose n_0 and β to maximize the second term. Also the value of β, 0 or 1, only changes the sign of the second term. Hence the minimization of eqn. (2.35) is equivalent to maximizing $|\rho(n_0)|$ where

$$\rho(n_0) = \sum_{l=1}^{L} A_l^2 \cos[\theta_l + n_0\omega_l - \Theta_s(\omega_l)] \tag{2.36}$$

The above derivation shows that if the magnitudes of the sine waves are known, then the MSE criterion can lead to a technique which estimates the pitch onset time n_0 assuming that the glottal pulse and vocal track networks are of minimum phase. However, because of the minimum phase assumption, the sine wave phase model has its inadequacy at high frequency of voiced speech and introduces reverberance in unvoiced speech. In order to cope with this inadequacy, the model has been modified to introduce a voicing transition frequency below which voiced speech is synthesized and above which unvoiced speech is synthesized. The voicing transition frequency depends on voicing probability of speech, which is defined similar to the value of ρ in eqn.(2.36).

2.5 Multiband Excitation Coding

Multiband excitation coding (MBE) was proposed by Griffin and Lim [23]. As a member of sinusoidal coding family, MBE is similar to STC in the sense that both

of them treat speech as a summation of sinusoids. But they differ in the parameters estimation as well as the synthesis of reconstructed speech. In STC, voiced speech is synthesized below a voicing transition frequency, and unvoiced speech is synthesized above the voicing transition frequency. The main innovation of the MBE is that speech is separated into many frequency bands, and voiced and unvoiced decision is made for each individual frequency band. This represents the excitation signal better and results in better speech quality compared to the original vocoder, which has only one single frequency band.

A refined version of MBE, called *improved multiband excitation* (IMBE) [25], was adopted as the Inmarsat standard.

2.5.1 Speech Synthesis in MBE

In MBE, the voiced component of speech is synthesized in the time domain while the unvoiced component of speech is synthesized in the frequency domain. The two components are added up to obtain the reconstructed speech. The voiced signal is synthesized as the sum of sinusoidal oscillators with frequencies at the harmonics of the fundamental and with magnitudes set by the spectral envelope parameters. Similar to STC, this technique has the advantage of allowing the fundamental frequency to vary continuously from frame to frame.

A block diagram of the synthesis model for MBE is shown in figure 2.4. As we can see in the figure, the voiced portion of speech is synthesized from voiced magnitude envelope samples by adding up the outputs of a bank of sinusoidal oscillators. Each oscillator has a frequency equal to a multiple of the *fundamental frequency*. In the continuous time domain, the voiced portion of the synthesized speech can be written

as

$$\hat{s}_v(t) = \sum_{l=1}^{L} \hat{A}_l(t) \cos(\hat{\phi}_l(t)) \tag{2.37}$$

where L is number of oscillators, and time-varying amplitudes $\hat{A}_l(t)$ is a result of linear interpolation between frames assuming that the amplitudes of unvoiced harmonics are zeros. This equation appears to be a continuous time domain version of eqn. (2.21) in STC. However, the phase function $\hat{\phi}_l(t)$ is obtained in a different way as follows:

$$\hat{\phi}_l(t) = \int_0^t \omega_l(\tau) d\tau + \phi_0 \tag{2.38}$$

where ϕ_0 is the initial phase, and the frequency track $\omega_l(t)$ is linearly interpolated between l^{th} harmonics of the current frame and that of the next frame using

$$\omega_l(t) = l\omega_0(0)\frac{T-t}{T} + l\omega_0(T)\frac{t}{T} + \Delta\omega_l \qquad 0 \le t \le T \tag{2.39}$$

where T is the frame length, $\omega_0(0)$ and $\omega_0(T)$ are the fundamental frequency of the current and next frames respectively, and $\Delta\omega_l$ is a frequency deviation. The initial phase ϕ_0 and frequency deviation$\Delta\omega_l$ have to be chosen so that the $\phi_l(0)$ and $\phi_l(T)$ are equivalent to the measured harmonic phases in the current and next frames (modulo 2π). $\Delta\omega_l$ should be chosen to be the smallest deviation required to match the measured phases. If either of l^{th} harmonics of the current frame or that of the next frame is marked unvoiced, $\Delta\omega_l$ is simply set to be zero because only one measured phase is required to be matched.

The unvoiced portion of speech, on the other hand, is synthesized in the frequency domain as shown in the lower branch of figure 2.4. A white noise sequence is windowed, and FFT is applied to produce samples of Fourier transform. In each unvoiced frequency band, the magnitudes of Fourier transform are replaced by the unvoiced spectral envelope, which is a result of linear interpolation between the envelope sample magnitudes $A_l(t)$. Then inverse transform can be taken to obtain the time-domain unvoiced portion of synthesized speech.

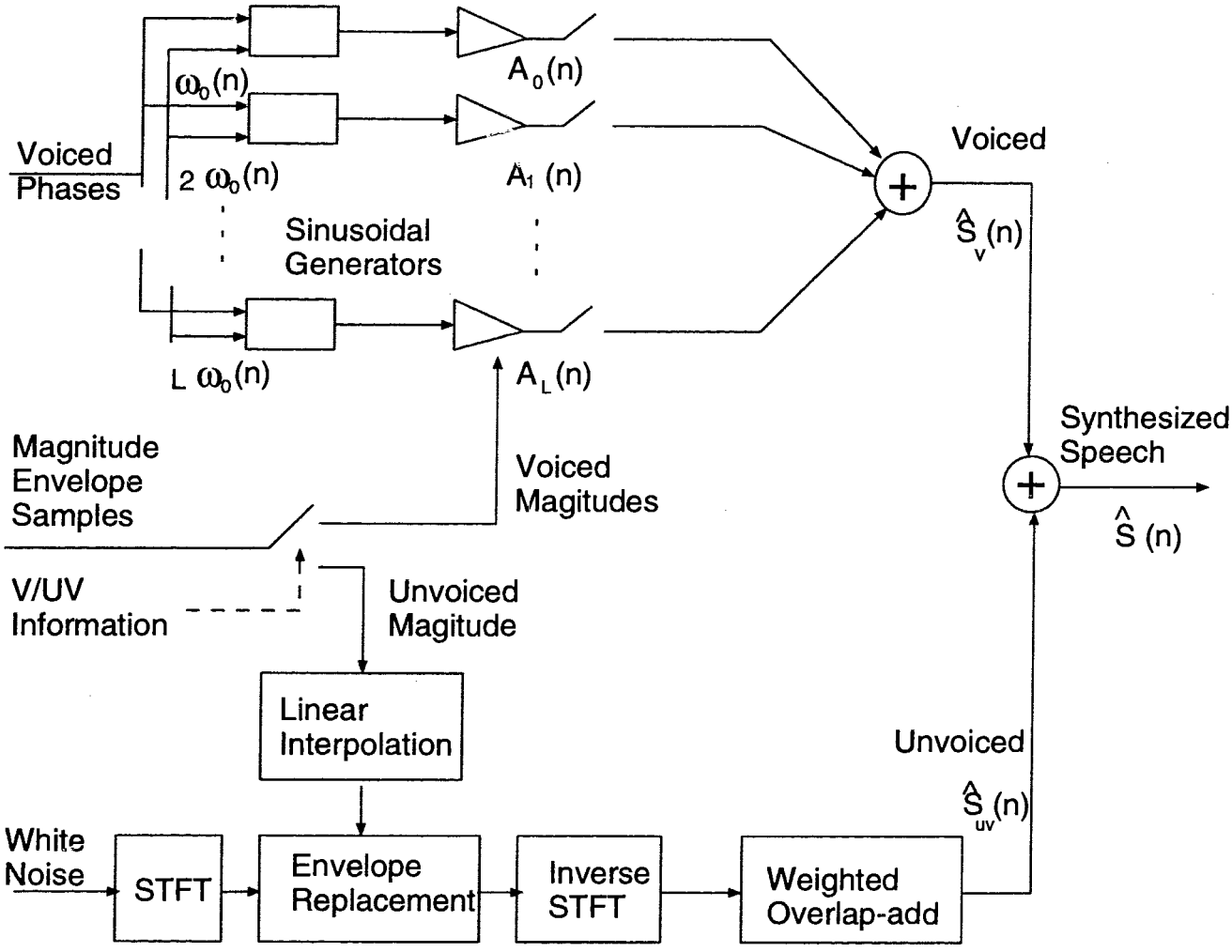

Figure 2.4: Block Diagram of the MBE Synthesis Model

Finally, the synthesized speech is generated by adding up the voiced and unvoiced portions.

2.5.2 Phase Quantization in MBE

In the original MBE [23], the phase information is preserved for the purpose of obtaining high quality synthesized speech. Two classes of phase information are considered important in MBE: the phase difference between consecutive frames and relative phases between harmonics in voiced regions. To encode the phases, the phase of the current frame are predicted from that of the previous frame, and the difference between the predicted and measured phase for the current frame is quantized using a Lloyd-Max quantizer.

The phases of harmonics in frequency regions declared unvoiced do not need to be coded because they are not required by the speech synthesizer. Only the phases of voiced bands are encoded, hence the bit rate required for phase quantization varies from 0 bit/sec for completely unvoiced speech to 2250 bit/sec for complete voiced speech [23].

2.5.3 The IMBE Coder

Improved multiband Excitation (IMBE) [25] is a refined version of MBE. Compared to MBE (8 kbps), it has a reduced bit rate of 4.15 kbps, and it uses predicted phases for voiced speech synthesis instead of the measured phases.

Similar to MBE, the unvoiced and voiced components of the reconstructed speech signal are synthesized separately. The unvoiced component speech synthesis is the

same as that of MBE. However, the synthesis method for the voiced component is different from that of MBE simply because the measured phases are not available in IMBE. Following is a brief description of IMBE's synthesis scheme for voiced components.

Assume N is the synthesis frame length, the phase of l^{th} harmonic in frame k is replaced by a predicted phase defined as

$$\phi_l^k = \phi_l^{k-1} + \left[\frac{\omega_0^{k-1} + \omega_0^k}{2}\right] N/2. \tag{2.40}$$

where ϕ_l^{k-1} is the predicted phase of frame $k-1$, and ω_0^k and ω_0^{k-1} are the fundamental frequencies of frames k and $k-1$ respectively.

To avoid "buzziness" in the reconstructed speech as a result of too much phase coherence in eqn. (2.40), the IMBE algorithm includes a mechanism for adding random noise to the phases of the upper voiced harmonics in proportion to the percentage of bands declared unvoiced.

After obtaining the predicted phase, IMBE uses the quadratic phase interpolation described in eqns. (2.38) and (2.39) to preserve phase continuity at frame boundaries by allowing a small discontinuity in frequency, $\Delta\omega$. In the discrete time domain, the detailed interpolation equations can be written as

$$\phi_l(n) = \phi_l^{(k-1)} + [\omega_0^{(k-1)} l + \Delta\omega]n + [\omega_0^{(k)} - \omega_0^{(k-1)}]\frac{ln^2}{2N} \tag{2.41}$$

$$\Delta\phi = \phi_l^{(k)} - \phi_l^{(k-1)} - [\omega_0^{(k-1)} - \omega_0^{(k)}]\frac{lN}{2} \tag{2.42}$$

$$\Delta\omega = \frac{1}{N}\left[\Delta\phi - 2\pi(\frac{\Delta\phi + \pi}{2\pi})\right] \tag{2.43}$$

where $0 \leq n < N$. Note that eqn. (2.43) sets $\Delta\omega$ to the smallest possible value which, when used in eqn. (2.42) will guarantee phase continuity at the boundary samples $n = 0$ and $n = N$.

2.6 Summary

In this chapter, the basic concepts of linear prediction and vector quantization have been briefly described. We discuss a dominant speech coding algorithm, CELP, which utilizes both concepts of linear prediction and vector quantization. While CELP coders generate high quality speech above the rate of 4 kbps, sinusoidal vocoders have emerged in recent years as alternatives to CELP at low bit rates. We describe in this chapter two versions of versions of sinusoidal coders, namely STC and MBE.

Chapter 3

Interpolative Speech Coding

Initially proposed by Kleijn [30], interpolative coding is a relatively new technique which lies in the grey area between waveform coding and vocoders and is seen as a merging of CELP and sinusoidal coding. Because CELP (a dominant waveform coder) and sinusoidal coding (an important class of vocoders) are complementary, interpolative coding has good potential to provide high quality speech at low bit rates. This chapter describes current available interpolative coding schemes, which will facilitate understanding for chapters 4 and 5.

3.1 The Concept and Basic Structure

CELP can generate high quality speech at bit rates above 4.8kbps using the analysis-by-synthesis procedure. This procedure is essentially a waveform matching technique to reconstruct synthesized speech as closely as possible to original speech. As the bit rate goes down below 4.8 kbps, the accuracy of the matching decreases thereby creating noisy synthesized speech.

Voiced speech is a quasi-periodic signal; CELP takes advantage of the quasi-periodicity by employing a long-term pitch filter or an adaptive codebook. To exploit better this periodicity, interpolative coding encodes only parts of a speech signal, called *prototype waveforms*, and the missing parts of the speech signal are recovered by interpolation between encoded prototype waveforms. A prototype waveform is defined as the waveform of a single pitch cycle. Figure 3.1 shows the concept of prototype waveform interpolation. The solid lines in the figure present prototype waveforms, while the dashed lines present recovered waveforms. The recovered waveforms are obtained by interpolating the waveform shapes as well as the pitches of prototype waveforms.

Various interpolative coding systems [54] [31] [8] [39] [50] [58] have been proposed since the introduction of the PWI coder [30]. These systems differ in detailed waveform representation, quantization, and interpolation. However, the basic structure of different interpolative coders is similar and can be illustrated as in figure 3.2. In this block diagram, prototype waveforms can be either extracted from input speech or residual. The basic structure can be divided into four functional blocks: (1) prototype waveform extraction (2) waveform alignment (3) representation and quantization of prototype waveforms (4) interpolation between prototype waveforms. These functional blocks are inherently dependent on each other. For example, a specific implementation of prototype waveform alignment depends on the specific representation and quantization of prototype waveforms. However, we attempt to separate these functional blocks as much as possible for the purpose of easy understanding of the basic structure. In the rest of this section, functional blocks (1) and (2) will be discussed together with a general form of interpolation between prototype waveforms. In sections 3.2 and 3.3, functional blocks (3) and (4) will be discussed in detail.

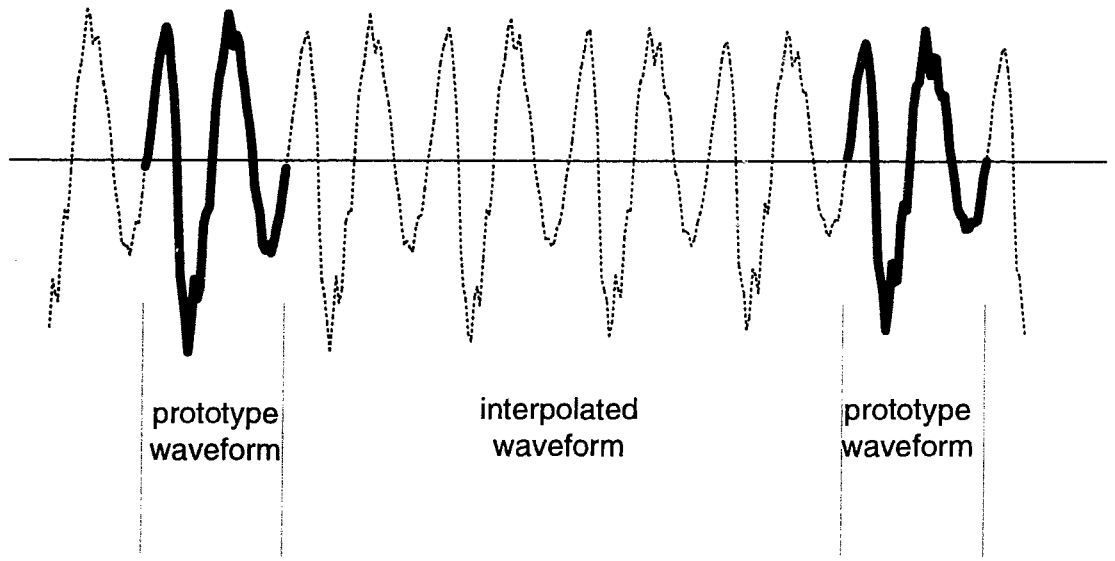

Figure 3.1: The Concept of Prototype Waveform Interpolation

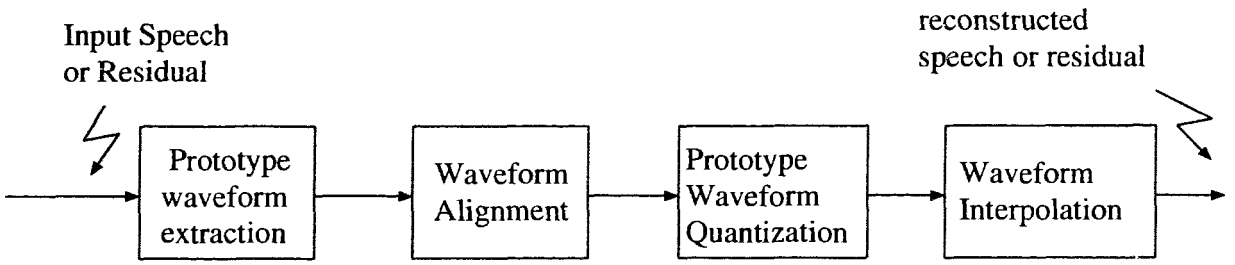

Figure 3.2: The Basic Structure of Prototype Waveform Interpolative Coding

3.1.1 Single versus Multiple Prototypes

Earlier interpolative coding schemes, such as prototype waveform interpolation (PWI) [30] and time frequency interpolation (TFI) [54], belong to a class called *single prototype waveform interpolation.* In the past two or three years, another class of interpolative coding, called *multiple prototype waveform interpolation*, has emerged [32] [9]. Single prototype systems typically transmit a prototype waveform at intervals of 10ms-20ms. Single PWI systems (PWI here is used as a general term) can encode only voiced speech, and a different coding algorithm is required to encode unvoiced speech. In other words, the basic structure shown in figure 3.2 is only applicable to voiced speech. However, multiple PWI systems have a much higher prototype update rate, and the update rate is high enough that the same algorithms can be used to encode both voiced and unvoiced speech.

In multiple PWI, because of the high update rate, prototypes extracted will overlap each other on the time axis as shown in figure 3.3. In this figure, prototype 1 and prototype 2 are extracted from different starting points, but part of prototype

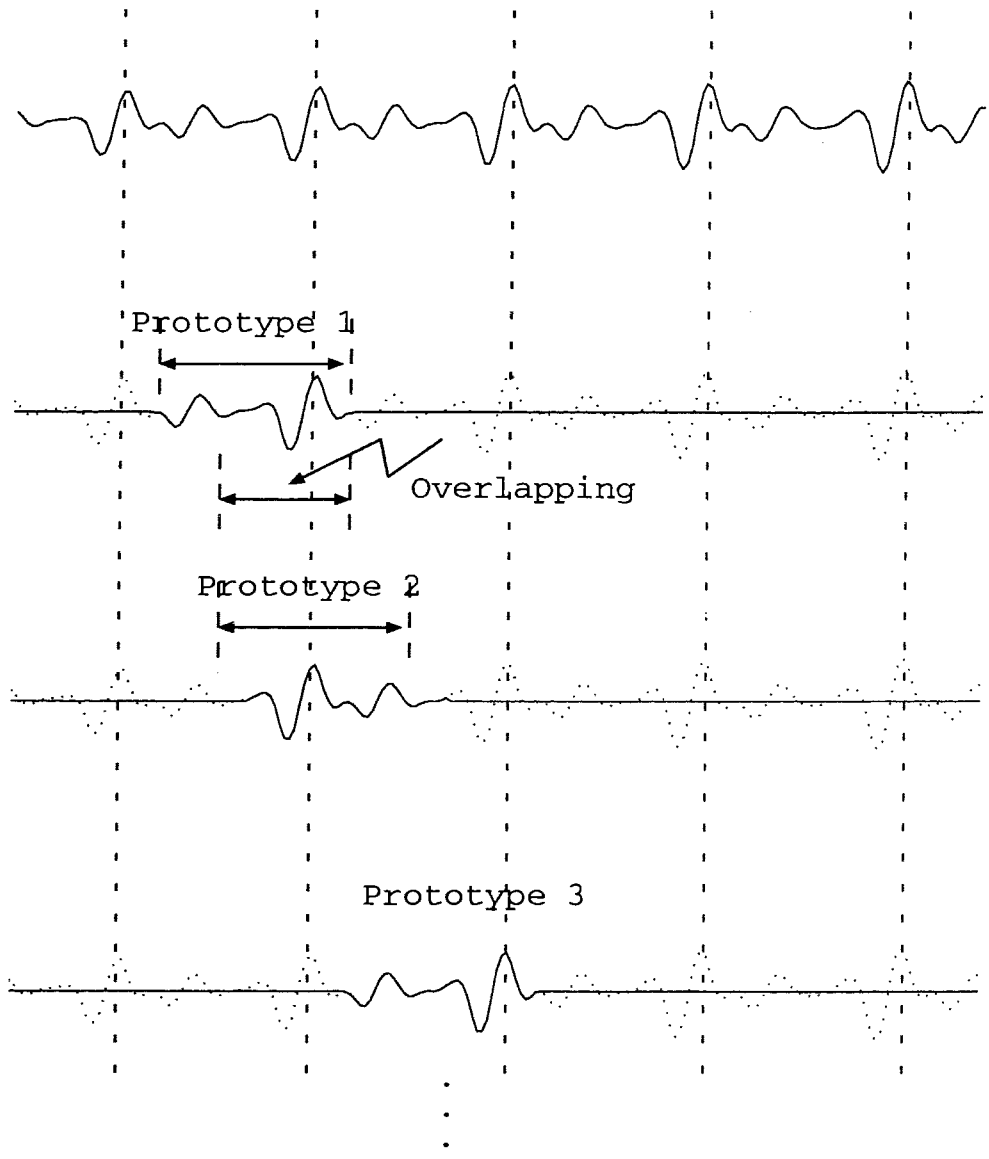

Figure 3.3: Prototype Waveforms Overlap with Each Others In Multiple PWI

1 is the same as part of prototype 2. Because of this overlapping, more complicated interpolation schemes are required in multiple PWI. Section 3.3 provides a more detailed description of interpolation schemes, and section 5.1 presents an efficient interpolation scheme based on the proposed phase codebook model.

3.1.2 Practical Prototype Extraction

As shown in figure 3.2, prototype waveforms can be extracted either from a speech signal or from a residual signal depending on the detailed structure of the interpolative coder. A prototype extraction algorithm searches through a given segment of a speech or residual signal to obtain a prototype waveform, i.e., one of the pitch cycles of the given segment. The objective of the algorithm is to maintain continuity at prototype boundaries if we repeat periodically the prototype waveform on the time axis.

Kleijn [30] has proposed an extraction method on the speech signal using a criterion which maximizes the prediction gain. First an arbitrary location point is defined. Starting from this point, take an interval of speech with the length of the interval close to the estimated pitch period. Then repeat this interval to generate a periodic signal and perform linear prediction on the periodic signal. The length which results in a maximum short-term prediction gain is chosen to be the length of the prototype waveform. This length minimizes boundary discontinuity because the linear predictor cannot predict discontinuity at boundaries of repeated intervals. The disadvantage of this method is the high computational complexity.

The above maximum prediction gain scheme cannot be directly applied to a residual signal. However, prototype waveforms can be extracted using pitch-pulse markers from an upsampled residual signal. The pitch pulses are first located, and

a prototype waveform boundary is selected in the middle of two pitch pulses. This will likely result in a low energy level at the prototype waveform boundaries because most residual energy resides around pitch pulses.

3.1.3 Alignment of Prototype Waveforms

Waveform alignment is an important concept in interpolative coding. Because a prototype-extracting algorithm can locate the starting point of a prototype at any location in a pitch cycle, waveform alignment is necessary both to have better prototype quantization efficiency and to reconstruct synthesized speech using prototype waveform interpolation. Figure 3.4 illustrates the concept of alignment between prototype waveforms. In the figure, prototype waveform 2 is aligned with prototype waveform 1.

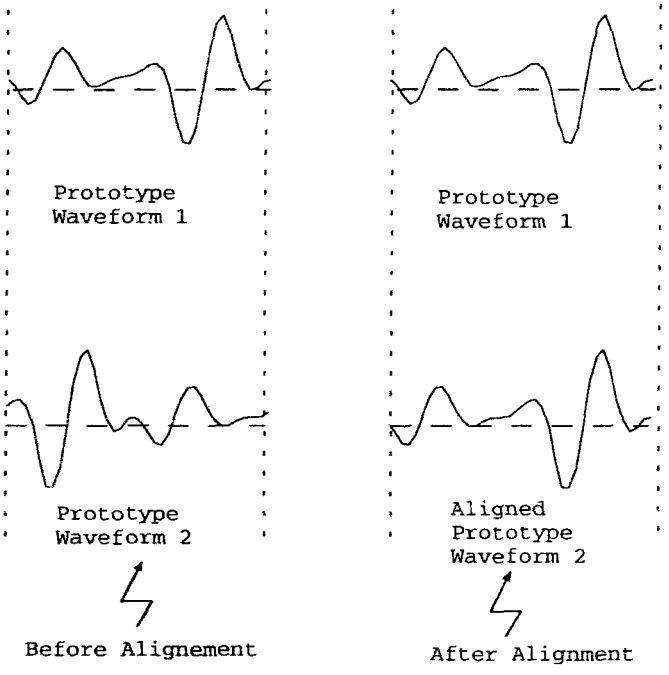

Figure 3.4: The Concept of Alignment between Prototype Waveforms

Note that the prototype waveforms 1 and 2 in figure 3.4 are of the same length,

which is not generally the case because of the varying pitch. To align two prototype waveforms with different lengths in the time domain, one has to normalize the prototype waveforms to a fixed length (i.e. 2π), then the circular cross-correlation function between these normalized prototypes can be used to find the best alignment shift between two prototypes.

Before introducing the definition of the circular cross-correlation function, let us introduce the concept of the *instantaneous prototype waveform* [30].

A prototype waveform can be seen as one cycle of a periodic signal, with its period (pitch) and waveform shape evolving with time. For a given time instant t, the pitch and shape of the prototype waveform can be frozen. By introducing an abstract time axis τ, the frozen prototype waveform can be seen as one cycle of a periodic function $z(t, \tau)$. Associated with each time instant t, is a pitch value $p(t)$–the pitch period of $z(t, \tau)$. Normalizing the pitch period of $z(t, \tau)$ in the time axis τ, we obtain a periodic function, called the *instantaneous prototype waveform* at time instant t, as follows:

$$u(t, \phi) = z(t, p(t)\frac{\phi}{2\pi}) \tag{3.1}$$

where ϕ is the same as the continuous time axis τ with a scale factor. $u(t, \phi)$ is periodic with period 2π:

$$u(t, \phi) = u(t, \phi + 2k\pi) \qquad k = 0, \pm 1, \pm 2, \ldots \tag{3.2}$$

Note that $u(t, \phi)$ can be seen as a two dimensional signal of the time axis t and the abstract time axis ϕ. When the time instant t is fixed, we have an instantaneous prototype waveform, $u(t, \phi)$, along abstract time axis ϕ.

Now suppose two prototype waveforms $u(t_{i-1}, \phi)$ and $u(t_i, \phi)$ are available at time instants $t = t_{i-1}$ and $t = t_i$ respectively, and we want to align these two prototype

waveforms. The circular cross-correlation function between $u(t_{i-1}, \phi)$ and $u(t_i, \phi)$ is defined as

$$R(u(t_{i-1}, \phi), u(t_i, ((\phi - \xi))_{2\pi}) = \int_0^{2\pi} u(t_i, \phi) u(t_i, ((\phi - \xi))_{2\pi}) d\phi \tag{3.3}$$

where $((\phi - \xi))_{2\pi}$ means *modulo* 2π. The best circular shift ξ is obtained using

$$\xi = \arg \max_{\xi'} \left\{ \int_0^{2\pi} u(t_i, \phi) u(t_i, ((\phi - \xi'))_{2\pi}) d\phi \right\} \tag{3.4}$$

This technique is a general time-domain alignment scheme. The specific implementation of the time-domain alignment depends on how the prototype waveforms are represented. In [30], since prototypes are represented as a set of Fourier-series coefficients, the time-domain correlation becomes a procedure which exhaustively searches for a best phase shift angle from 0 to 2π (See section 3.2 for details). The disadvantage of time-domain based alignment algorithms is high computational complexity.

Another way of performing alignment is in the frequency domain. Frequency domain schemes are more efficient computationally due to the fact that integration in the continuous time domain (or convolution in the discrete time domain) becomes multiplication in the frequency domain. In this thesis, a frequency domain alignment scheme is used (See section 5.2.5). Burnett and Bradley [9] have also used a frequency domain scheme.

3.1.4 General Form of Interpolation Between Prototypes

Practical schemes of interpolation between prototypes depend on how prototype waveforms are represented and quantized. For example, a prototype waveform can be represented as a set of Fourier series coefficients. Because this Fourier series representation implicitly includes the spectral phase of the prototypes, the corresponding

interpolation scheme can use the available spectral phase in the interpolation procedure, and it can be quite different from other interpolation schemes used by a representation without spectral phase.

However, before going into detailed representations and quantization of prototype waveforms in the following sections, a discussion of the general form of interpolation between prototypes is helpful. Following is the derivation of the general form of interpolation between two prototype waveforms [30].

An "ideal" interpolation should not generate discontinuity and should have a smooth pitch contour. Let $u(0,\phi)$ and $u(T,\phi)$, as defined in eqn. (3.1), be two instantaneous prototype waveforms at time instant $t = 0$ and $t = T$ respectively, and we want to perform waveform interpolation between these two prototype waveforms. Waveform alignment discussed in the previous subsection is required before interpolation. For simplicity, assume these two prototype waveforms have been aligned. The interpolated instantaneous prototype waveform at time instant t, where $0 \leq t < T$, can be written as

$$u(t,\phi) = (1 - \alpha(t))u(0,\phi) + \alpha(t)u(T,\phi) \qquad 0 \leq t \leq T \tag{3.5}$$

where $\alpha(t)$ is a monotonically increasing interpolation function and satisfies $\alpha(0) = 0$ and $\alpha(T) = 1$. Only when both $u(0,\phi)$ and $u(T,\phi)$ have the same normalized pitch period, the interpolation eqn. (3.5) makes sense.

If the same interpolation function $\alpha(t)$ is applied to the interpolation of pitch values, the pitch value at time instant t is

$$p(t) = (1 - \alpha(t))p(0) + a(t)p(T) \qquad 0 \leq t \leq T \tag{3.6}$$

The pitch $p(t)$ changes all the time along time axis t from 0 to T, and we have to reconstruct a signal in which the instantaneous interpolated prototype waveform at time instant t satisfies:

1. The instantaneous waveform shape is obtained using eqn.(3.5)

2. The instantaneous pitch of the waveform is $p(t)$ in eqn. (3.6)

The desired signal $e(t)$, can be obtained by the concatenation of infinitesimal segments of the instantaneous prototype waveforms satisfying the above two conditions. The first infinitesimal segment is obtained by un-normalizing the normalized prototype waveform $u(0, \phi)$ in an infinitesimal interval dt around time instant $t = 0$:

$$e(0 + dt) = u\left(0, \phi(0) + \frac{2\pi}{p(0)}dt\right) \tag{3.7}$$

By maintaining the continuity of the phase between the boundaries of these infinitesimal intervals, we have the desired reconstructed signal:

$$e(t) = u(t, \phi(t)) = u\left(t, \phi(0) + \int_0^t \frac{2\pi}{p(t')}dt'\right) \qquad 0 \leq t \leq T \tag{3.8}$$

Eqn. (3.8) is the general form of waveform interpolation. It applies to both single PWI and multiple PWI systems. In practice, this continuous-time-domain general form of waveform interpolation is computationally expensive to approximate in the discrete time domain. However, we will show in chapter 5 that this form of interpolation can be efficiently implemented in the discrete time domain based on the proposed phase model.

There are different ways of implementing the general form of prototype waveform interpolation depending on how prototype waveforms are represented and quantized. The next section discusses the representation and quantization of prototype waveforms, followed by a section describing different approaches to interpolate prototype waveforms.

3.2 Representation and Quantization of Prototypes

In interpolative coding, one of the most important questions is how to represent and quantize prototype waveforms. This section discusses some typical representation and quantization schemes using in interpolative coding.

3.2.1 Direct Waveform Quantization

One straightforward scheme is direct waveform quantization. Direct waveform quantization can be implemented either by scalar quantization or by vector quantization(VQ).

Scalar waveform quantization is straightforward. For example, Taori, Sluijter, and Kathmann have proposed to use simple differential coding techniques with a fixed second-order predictor and a logarithmic quantizer, resulting in a bit rate of 6.75 kbps for voiced regions [50].

Vector quantization has better quantization efficiency compared with scalar quantization. However, direct VQ of prototype waveforms faces the problem of the variable length of prototype waveforms as the pitch values vary in time. Because of the variable length of prototype waveforms, vector quantization (VQ) schemes cannot be used directly. A way to overcome this problem is to oversample prototype waveforms to a fixed length. Another way is to employ *variable dimension vector quantization* (VDVQ) [17], which was initially developed for spectral magnitude quantization. Implementations of the oversampling scheme and VDVQ will be discussed and compared with the proposed phase codebook model in chapter 4. Finally, *non-square*

transform vector quantization (NSTVQ) [36] is also applicable (See chapter 4.)

3.2.2 Sine and Cosine Representation

In general, prototype waveforms can be extracted either in the speech domain or in the residual (excitation) domain depending on the exact structures of the speech coders. In the PWI proposed by Kleijn [30], prototype waveforms are extracted in the residual signal domain.

Since a prototype waveform can be seen as a cycle of a periodic signal, Kleijn represents an excitation prototype waveform as a set of Fourier-series coefficients:

$$u(t,\phi) = \sum_{l=0}^{L}[C_l(t)\cos(l\phi) + D_l(t)\sin(l\phi)] \tag{3.9}$$

where the number of L depends on the pitch $p(t)$ and the signal bandwidth defined by the Nyquist frequency of the sampled signal. The Discrete Fourier Transform (DFT) can also be used instead of Fourier series.

At each update point, a prototype waveform is extracted from a residual signal using a prototype extraction procedure (See section 3.1.2). Assume $u(0,\phi)$ and $u(T,\phi)$ are two successive prototype waveforms at time instants $t = 0$ and $t = T$ respectively. Usually, consecutive prototype waveforms are not properly aligned. To align prototype waveforms, eqn. (3.4) can be used to find the best circular phase shift ξ which maximizes the cross correlation R between these two prototype waveforms. In terms of Fourier series coefficients, eqn.(3.4) can be rewritten as

$$\begin{aligned}\xi \;=\; \arg\max_{\xi'} \sum_{l=0}^{L} \{&(C_l(0)C_l(T) + D_l(0)D_l(T))\cos(l\xi') + \\ &(D_l(0)C_l(T) - C_l(0)D_l(T))\sin(l\xi')\}\end{aligned} \tag{3.10}$$

By comparing eqn. (3.10) with the definition of Fourier-series coefficients in eqn.

(3.9), the cosine and sine coefficients of the properly aligned prototype at $t = T$ are

$$\begin{aligned} \tilde{C}_l(T) &= C_l(T)\cos(l\xi) - D_l(T)\sin(l\xi) \\ \tilde{D}_l(T) &= C_l(T)\sin(l\xi) + D_l(T)\cos(l\xi) \end{aligned} \tag{3.11}$$

Quantization of Sine and Cosine Coefficients

Assume two prototype waveforms $u(0,\phi)$ and $u(T,\phi)$ have been properly aligned using eqns. (3.10)-(3.11). The PWI starts quantization with differential coding between consecutive prototype waveforms based on the assumption that successive prototypes are similar to each other. In other words, the quantized version of $u(0,\phi)$ (with a scaling factor λ_0) is subtracted from $u(T,\phi)$ to obtain a differential signal. As shown in eqn. (3.9), $u(t,\phi)$ is represented by *sine* and *cosine* coefficients, and the coefficient vectors at different time instants generally do not have the same dimension because of the varying pitch $p(t)$. In [30], this problem is simply solved by adding zeros to shorter vectors. The differential vector is quantized by M codebooks ($M = 2$). Each codeword of these codebooks has a dimension equal to the largest number of harmonics, L. The quantized waveform of $u(T,\phi)$ can be written as

$$\hat{u}_(T,\phi) = \lambda_0\hat{u}(0,\phi) + \sum_{m=1}^{M} \lambda_m c_{k_m}^m(\phi) \tag{3.12}$$

where $\hat{u}(0,\phi)$ is the quantized $u(0,\phi)$, $c_{k_m}^m$ is the codevector with index k_m from the m^{th} codebook, and λ_m are gains. An analysis-by-synthesis technique similar to that in CELP is used for searching the best gains λ_m and the best codebook entries k_m. Because the synthesized speech is not necessarily synchronized with the original speech, alignment between the synthesized and original speech is required before MSE criterion is applied. The PWI performs alignment on the excitation signal with a single pulse. Also, the M-stage codebooks in the PWI are orthogonalized to gain better quantization efficiency and to better control similarities between

evolving prototype waveforms. More details of codebook searching and similarity measurement can be found in [30].

3.2.3 The Scheme in Mixed-Domain Coding

It is interesting to compare the prototype waveform quantization scheme of the PWI by Kleijn and a mixed-domain coding by DeMartin and Gersho [39]. Both schemes employ analysis-by-synthesis techniques, and both extract prototype waveforms in the residual domain. However, in terms of prototype waveform quantization, the mixed-domain method is more similar to CELP coders. It reconstructs prototype waveforms from three components : (a) the previous (quantized) pitch cycle with suitable time scaling (b) a selection from a single-impulse codebook, and (c) a selection from a noise codebook. Each component has a corresponding gain factor. The purpose of (a) is the same as the differential coding of the PWI, though implemented in a different way. The addition of (b) and (c) is to compensate for the difference between current and previous prototype waveforms. This is essentially a time-domain quantization scheme similar to CELP.

The mixed-domain coding selects the three components of the excitation prototype waveforms in a closed-loop manner. It differs from CELP in the updating of the memory of the synthesis filter; the mixed-domain scheme refreshes the memory with backward extended periodic excitation while CELP updates the memory of the synthesis filter using the excitation from the previous coding block (frame or subframe).

3.2.4 Magnitude-Only Quantization Schemes

In the sine-and-cosine representation in section 3.2.2, prototype waveforms are described as sets of Fourier series coefficients which constitutes a frequency domain description. Similarly, discrete Fourier transform (DFT) coefficients can also be used to represent prototype waveforms [54] [9] [28]. DFT coefficients can be further decomposed into spectral magnitudes and phases. However, spectral phases are difficult to quantize using conventional quantization schemes [47]. On the other hand, extensive research has been conducted on spectral magnitude quantization. For this reason, similar to some sinusoidal coders, some interpolative coders simply drop the phase information on the arguable assumption that phase information is not important to the human ear.

In one interpolative coder, TFI [54], Shoham has proposed a magnitude-only quantization scheme. The spectral magnitude is quantized by a weighted, variable-size, multi-stage vector quantizer after differential coding between the current and previous magnitude vectors. The differential coding is switched ON in continuous voiced segments and OFF in unvoiced-to-voiced transitional segments. Fixed-value gain (0.7) is used in differential coding, while the gains for multi-stage VQ are vector quantized.

Many magnitude quantization schemes used in sinusoidal coding can be readily used in magnitude quantization of prototype waveforms. In both sinusoidal coding and interpolative coding, associated with spectral magnitude quantization is the problem of variable vector dimension.

Several techniques have been developed to avoid the difficulties associated with quantization of variable dimension vectors. The Inmarsat multi-band excitation (IMBE) codec [25] uses a complicated hybrid scalar/vector quantization. Branstein

[7] presents a method which fits a fixed-order all-pole model to spectral magnitude vectors with variable dimension. Recently, a technique called *variable dimension vector quantization* (VDVQ) [17] has been proposed by Das and Gersho. NSTVQ [36] by Lupini and Cuperman is another approach which shows performance comparable to VDVQ. Both NSTVQ and VDVQ have been shown to outperform the IMBE scheme and the all-pole scheme.

Another possible approach is to warp the variable dimension magnitude vectors to a fixed length using a mechanism similar to the oversampling technique described in chapter 4. The validity of this scheme can be easily seen after the discussion of oversampling in the proposed phase codebook model.

3.2.5 SEW and REW decomposition in Multiple PWI

Single prototype waveform interpolation has a low update rate of prototype waveforms, resulting in lower bit rates compared to conventional coding schemes in which complete speech frames are quantized. However, a low update rate means a slow evolution of prototype waveforms. This generates a reconstructed speech signal with high periodicity, making single PWI applicable only to voiced speech. An increase in the update rates not only allows encoding non-periodic signals such as unvoiced speech but also improves the quality of voiced speech. However, increasing the update of the prototype waveform increases the bit rate.

To increase the update rate while keeping bit rates low, Kleijn and Haagen [32] [31] have proposed that a prototype waveform be decomposed into a *slowly evolving waveform* (SEW) and a rapidly evolving waveform (REW). The purpose of decomposing prototype waveforms into SEWs and REWs is to reduce the quantization bit rate by exploiting the human auditory system. It is noted that unvoiced speech

can be encoded in a perceptually accurate manner at a very low bit rate [33]. This is the case despite the fact that unvoiced speech have a very high information rate from an information-theoretic viewpoint. On the contrary, the bit rate required for voiced speech is usually relatively high, despite the slow evolution of the pitch cycle waveform. These observations show that the human ear is more sensitive to slowly evolving feature of the waveform. The decomposition of prototype waveforms into SEWs and REWs facilitates the use of different quantization schemes for voiced-like SEW and noise-like REW [31].

Recall the definition of the instantaneous prototype waveform $u(t,\phi)$ in eqn. (3.1). At each update point t_i, there is an instantaneous prototype waveform $u(t_i,\phi)$. In [32], the signal power of $u(t_i,\phi)$ is first computed, and the prototype $u(t_i,\phi)$ is gain-normalized. The gain is transferred to the logarithm domain, low-pass filtered, and downsampled to a rate of 80Hz. The downsampled signal is then quantized using a differential quantizer. In the following derivation, we assume $u(t_i,\phi)$ has already been gain normalized for convenience.

In general, the prototype waveform $u(t_i,\phi)$ at time instant t_i is not aligned with its previous prototype waveform $u(t_{i-1},\phi)$. If prototype waveforms are represented by sine and cosine coefficients (See section 3.2.2), the alignment can be performed using eqns. (3.10) and (3.11). For time instant t_i, the aligned coefficients can be obtained as

$$\begin{aligned} \tilde{C}_l(t_i) &= C_l(t_i)\cos(l\xi_i) - D_l(t_i)\sin(l\xi_i) \\ \tilde{D}_l(t_i) &= C_l(t_i)\sin(l\xi_i) + D_l(t_i)\cos(l\xi_i) \end{aligned} \tag{3.13}$$

where ξ_i satisfies

$$\begin{aligned} \xi_i = \arg\max_{\xi'} \sum_{l=0}^{L} &\{(C_l(t_{i-1})C_l(t_i) + D_l(t_{i-1})D_l(t_i))\cos(l\xi') + \\ &(D_l(t_{i-1})C_l(t_i) - C_l(t_{i-1})D_l(t_i))\sin(l\xi')\} \end{aligned} \tag{3.14}$$

Now assume all prototype waveforms are aligned properly, and the *tilde* sign is dropped for convenience.

The Fourier series representation of the SEW can be obtained by linear-phase low-pass filtering of the following time sequences:

$$\ldots,\ C_l(t_{i-1}),\ C_l(t_i),\ C_l(t_{i+1}),\ \ldots \tag{3.15}$$

and

$$\ldots,\ D_l(t_{i-1}),\ D_l(t_i),\ D_l(t_{i+1}),\ \ldots \tag{3.16}$$

Note that these sequences are defined along time axis t. There is a *cosine* sequence and a *sine* sequence for each harmonic l. A *sine* sequence and a *cosine* sequence consist of *sine* and *cosine* coefficients from a specific harmonic l respectively.

Similar to the SEW, the REW is obtained by high-pass filtering of the above two sequences.

In [9], Burnett and Bradley replace the low-pass filtering operation with averaging the *sine* and *cosine* sequences along time axis t. In other words, the the low-pass filter is a rectangular window with a window size of L, where L is the number of prototype waveforms in a frame. Therefore, the SEW of a frame is the average of prototype waveforms in the given frame. Similarly, the REW is defined as the difference between a prototype waveform and the SEW of the given frame.

Quantization of SEW and REW

Because of the low-pass filtering , a SEW sequence can be downsampled, and SEW's update rate can be reduced to a value similar to that of prototype waveforms in a single PWI system.

In [32], the SEW is found to be important to speech quality whereas only the magnitudes of REW are quantized with low accuracy. The low-frequency part of the spectral magnitude of the SEW is vector quantized, and the high-frequency part is recovered assuming the overall (SEW and REW magnitudes) frequency response is flat. One of four phase spectra is selected for a SEW on the basis of the REW spectrum transmitted.

In [8], an open-loop quantization and closed-loop quantization scheme are proposed. The open-loop scheme uses vector quantization of the REW. Because the energy of prototype waveforms is normalized, there is an inherent linkage between the energies of the REW and SEW. A SEW at the decoder is reconstructed from a pulse shape (depending on the current pitch period) and a gain factor (depending on the energy of the quantized REW). Therefore no bits are required for the quantization of SEW. The closed-loop scheme selects an REW/SEW vector combination on the basis of spectral magnitude comparison with the normalized prototype waveforms.

3.3 Interpolation Between Prototypes

Similar to prototype waveform quantization, interpolation between prototype waveforms is another important issue which has a significant impact on speech quality. Based on different representations of prototype waveforms, different interpolation schemes have to be applied.

There are two classes of prototype waveform representation: with and without phase information. If the phase information is available, the general form of waveform interpolation (section 3.1.4) is applicable. However, if phase information is not available, other interpolation schemes are required. This section discusses inter-

polation schemes for both classes. Finally, some simplified schemes of the general form of interpolation are discussed.

3.3.1 Interpolation for Fourier Series and DFT Representation

When prototype waveforms are represented by sets of Fourier series (or DFT) coefficients as in the PWI by Kleijn [30], the interpolation between two prototype waveforms, $u(t_{i-1}, \phi)$ and $u(t_i, \phi)$, becomes an interpolation between Fourier series coefficients because of the linearity of the Fourier series expansion:

$$\begin{aligned} C_l(t) &= (1-\alpha(t))C_l(t_{i-1}) + \alpha(t)C_l(t_i) \\ D_l(t) &= (1-\alpha(t))D_l(t_{i-1}) + \alpha(t)D_l(t_i) \end{aligned} \tag{3.17}$$

where $C_l(t)$ and $D_l(t)$ are the interpolated Fourier series coefficients at time instant t, $t_{i-1} \leq t \leq t_i$, $C_l(t_{i-1})$ and $D_l(t_{i-1})$ are the coefficients at time instant t_{i-1}, and $C_l(t_i)$ and $D_l(t_i)$ are those at time instant t_i. Applying eqn.(3.8), the desired reconstructed signal $e(t)$ can be written as:

$$e(t) = u(t, \phi(t)) = \sum_{l=0}^{L} C_l(t)\cos(l\phi(t)) + D_l(t)\sin(l\phi(t)) \tag{3.18}$$

where $\phi(t)$ is the phase track of the instantaneous prototype waveform at time instant t:

$$\phi(t) = \phi(t_{i-1}) + \int_{t_{i-1}}^{t} \frac{2\pi}{(1-\alpha(t'))p(t_{i-1}) + \alpha(t')p(t_i)} dt' \tag{3.19}$$

The denominator in eqn. (3.19) is the result of linear interpolation between the pitch values of time instants t_{i-1} and t_i — $p(t_{i-1})$ and $p(t_i)$. Either numerical or analytical evaluation of eqn.(3.19) is possible [30]. In practice, numerical integration is a viable option to calculate $\phi(t)$. If the interpolation function $\alpha(t)$ is linear, an

explicit expression of ϕ can be found:

$$\phi(t) = \begin{cases} \phi(t_{i-1}) + 2\pi \frac{-Qp(t_{i-1}) + \sqrt{Q^2p^2(t_{i-1}) + 2Q(p(t_i) - p(t_{i-1}))(t - t_{i-1})}}{p(t_i) - p(t_{i-1})} & p(t_{i-1}) \neq p(t_i) \\ \phi(t_{i-1}) + 2\pi \frac{t - t_{i-1}}{p(t_{i-1})} & p(t_{i-1} = p(t_i) \end{cases} \tag{3.20}$$

where Q is defined as:

$$Q = 2\frac{t_i - t_{i-1}}{p(t_{i-1}) + p(t_i)}. \tag{3.21}$$

In eqn. (3.18), the trigonometric operations are computationally expensive. Sen and Kleijn [53] have proposed a recursive procedure to evaluate these trigonometric operations. The trigonometric identities used in the recursion are:

$$\begin{aligned} \cos(l\phi) &= 2\cos(\phi)\cos((l-1)\phi) - \cos((l-2)\phi) \\ \sin(l\phi) &= 2\cos(\phi)\sin((l-1)\phi) - \sin((l-2)\phi) \end{aligned} \tag{3.22}$$

This recursive procedure reduces the number of trigonometric operation from $2L$ to 2 for each reconstructed sample of $e(t)$.

3.3.2 Interpolation Without Spectral Phase

Magnitude-only prototype waveform quantization schemes are used in some interpolative coding systems such as [54]. In these coders, phase information is not available for the interpolation between prototype waveforms. A common method of interpolation is to let the phase run freely under the constraint that there will be no discontinuity in the reconstructed signal.

This method is similar to sinusoidal coding. Eqns.(2.38) and (2.39) from MBE are applicable to magnitude-only prototype waveform interpolation if $\Delta\omega_l$ is replaced with zero, and ω_0 with $2\pi/p(t)$. The reason the frequency deviation $\Delta\omega_l$ is set to zero is that no measured phase is available, and there is no need for any deviation

to match a measured phase. As a matter of fact, in [54], prototype waveform interpolation is performed based on eqns.(2.38) and (2.39).

In multiple prototype waveform interpolation, SEW is usually assigned a specific phase spectrum by some heuristic techniques [32] [9], and its interpolation is handled by schemes similar to those in section 3.3.1. As an example of heuristic techniques for phase spectra, in [32], one of four phase spectra is selected based on the REW spectrum. However, the reconstructed rapid evolving waveform (REW) is usually recovered from the spectral magnitudes only. In this case, random processes are used to imitate the random-noise-like characteristics of REW.

3.3.3 Simplified Interpolation Schemes

Because the general form of interpolation is computationally expensive, simplification of interpolation is desirable. One simplification of the general form is obtained when an instantaneous prototype waveform is updated only once per pitch cycle. Another possible simplification is that a prototype waveform can be truncated or padded with zeros to account for pitch changes in interpolation intervals. As a matter of fact, before the introduction of the PWI, researchers used similar schemes such as the pitch-synchronous overlap-and-add procedures in the area of time-scaling of speech [51] [11]. The disadvantage of these simplified block-wise schemes is the discontinuity at cycle boundaries, and the extra steps required to smooth the discontinuity. One practical scheme to smooth the discontinuity is to concatenate cycles at low-signal-energy areas of speech.

Recently, another simplified method was proposed by Taori etc. ([50]) to overcome the discontinuity problem at prototype waveform boundaries. This approach slightly modifies the definition of a prototype waveform: a speech segment of two

successive pitch cycles are extracted and windowed (raise-cosine window). Then the two cycles are separated, and overlapped with each other to form a modified prototype waveform. The operation smears the prototype waveform. However, when this operation is repeated, there is no discontinuity in the speech waveform at boundary points.

Chapter 4

The Phase Codebook Model

This chapter analyzes in detail the proposed phase codebook model. First, the motivation and objective of the model are described. In section 4.2, the phase codebook model is derived based on the minimum mean square error (MSE) criterion, and the properties of the basic model are given. In order to reduce the computational complexity of the model, three oversampling (or interpolation) techniques of prototype waveforms are discussed in section 4.3. In section 4.4, the oversampling techniques are used to reduce computational complexity of the model, resulting in a modified phase codebook model. Finally, performance of the phase codebook model is compared with those of direct waveform quantization schemes in section 4.5.

4.1 Motivation and Objective

In the discussion on sinusoidal coding and interpolative coding in chapters 2 and 3, it was shown that a major problem in sinusoidal coders and interpolative coding is the difficulty of quantizing the spectral phase information.

In some sinusoidal coding approaches such as original MBE [23] (section 2.5), spectral phase is extracted from the input speech spectrum and quantized. Unfortunately, encoding phase directly requires too many bits. Pearlman and Gray have found that encoding the phase of DFT coefficients needs more bits than encoding the spectral magnitudes [47]. Because of the high bit rate of phase quantization, other versions of the sinusoidal model discard the measured phase in order to obtain over-all low bit rates. For example, IMBE [25] encodes the spectral magnitude only and use the predicted phase to reconstruct speech at the decoder. An early version of STC [42] assumes zero-phase of the system function. Later versions [40] recover phase information from spectral magnitudes using the Hilbert transform based on the assumption that speech is a minimum phase signal (section 2.4). PWI [30] encodes Fourier series coefficients. These coefficients implicitly include phase information. TFI [54] quantizes the spectral magnitudes only and uses predicted phase to reconstruct synthesized speech (section 3.3).

The discarding of phase information in both sinusoidal coding and interpolative coding is based on the assumption that phase is not important to human ears. While some of the coders without phase information generate good quality speech, the assumption is arguable. The mechanism of the human auditory system is not well understood, and some experiments such as [45] [18] [19] have shown the importance of phase information.

The objective of this thesis is to develop a model to efficiently quantize spectral phase, especially the spectral phase of prototype waveforms. The proposed phase codebook model intends to provides a way of quantizing the spectral phase of prototype waveforms at low bit rates and to provide an alternative way of prototype waveform quantization.

4.2 The Phase Codebook Model

The proposed phase codebook model uses a phase vector from a phase codebook to represent the phase information of a prototype waveform. The phase vector is selected by applying the minimum MSE criterion to the difference between the original and reconstructed prototype waveforms.

Assume the original and synthesized speech prototype waveforms are $s(n)$ and $\hat{s}(n)$ respectively, where $0 \leq n < p$. $S(k)$ and $\hat{S(k)}$, where $0 \leq k < p$, are the DFTs of $s(n)$ and $\hat{s}(n)$. The size of the prototype waveform, p, varies from frame to frame according to the fundamental (pitch) period. With a circular alignment shift m, the minimum MSE (MMSE) is

$$\begin{aligned} MSE &= \min_{0 \leq m < p} \left(\sum_{n=0}^{p-1} (s(n) - \hat{s}((n-m))_p)^2\right) \\ &= \sum_{n=0}^{p-1} s^2(n) + \sum_{n=0}^{p-1} \hat{s^2}(n) - 2 \max_{0 \leq m < p} \sum_{n=0}^{p-1} s(n)\hat{s}((n-m))_p \end{aligned} \tag{4.1}$$

where $((n-m))_p$ means $(n-m)$ modulo p, and m represents the circular shift required for alignment between the original and synthesized prototype waveforms. The first and second terms in Equation (4.1) are the energies of the original and synthesized prototype waveforms respectively. From Parseval's Theorem, the signal energy depends only on its spectral magnitudes, not the spectral phases. In other words, once the spectral magnitudes are quantized, the energy of the synthesized prototype waveform—the second term in eqn. (4.1) – is determined. Only the third term – the circular cross-correlation between $s(n)$ and $\hat{s}(n)$— has to be considered in the minimization, and it can be represented in the frequency domain as:

$$\begin{aligned} z(m) &= \sum_{n=0}^{p-1} s(n)\hat{s}((n-m))_p \\ &= \frac{1}{p} \sum_{k=0}^{p-1} S(k)\hat{S}^*(k) \exp^{j2\pi mk/p} \quad 0 \leq m < p \end{aligned} \tag{4.2}$$

In other words, the minimization of MSE between $s(n)$ and $\hat{s}((n-m))_p$ is equivalent to finding the maxima in the inverse discrete Fourier transform (IDFT) of $S(k)\hat{S}^*(k)$.

Based on the above derivation, the phase codebook model can be illustrated in figure 4.1. In this figure, $\hat{S}(k)$ is a combination of quantized spectral magnitude vector and a spectral phase vector from the phase codebook. For each phase vector, we can find a maxima in the IDFT of $S(k)\hat{S}^*(k)$. Corresponding to the maxima is the best circular shift m', and a best match between $\hat{s}((n-m'))_p$ and $s(n)$. In other words, $\hat{s}(n)$ is aligned with $s(n)$ when $\hat{s}(n)$ is circularly shifted by m' samples.

The above procedure is repeated for all the possible phase codebook vectors, and all the correlations $z(m')$ are computed. By comparing all the $z(m')$, we can find the overall maxima. Corresponding to the overall maxima is the best phase codevector. This concludes the phase codebook search procedure.

Figures 4.2 and 4.3 show typical matches between original and synthesized prototype waveforms using the phase codebook model. The solid lines are typical original prototype waveforms, and the dashed lines are synthesized prototypes. the phase codebook size for the experiments in Figures 4.2 and 4.3 is 1024 (10bits).

Closed-loop Prototype Waveform Alignment

As we can see from the above derivation, prototype waveform alignment and the phase codebook search are performed simultaneously by using simple peak picking applied to the IDFT of $S(k)\hat{S}^*(k)$. In other words, the proposed model leads to a closed-loop waveform alignment, in which a prototype waveform is aligned during the quantization procedure. The closed-loop alignment is different from the open-loop ones performed by other interpolative coders (chapter 3), where prototype waveforms are always aligned before quantization.

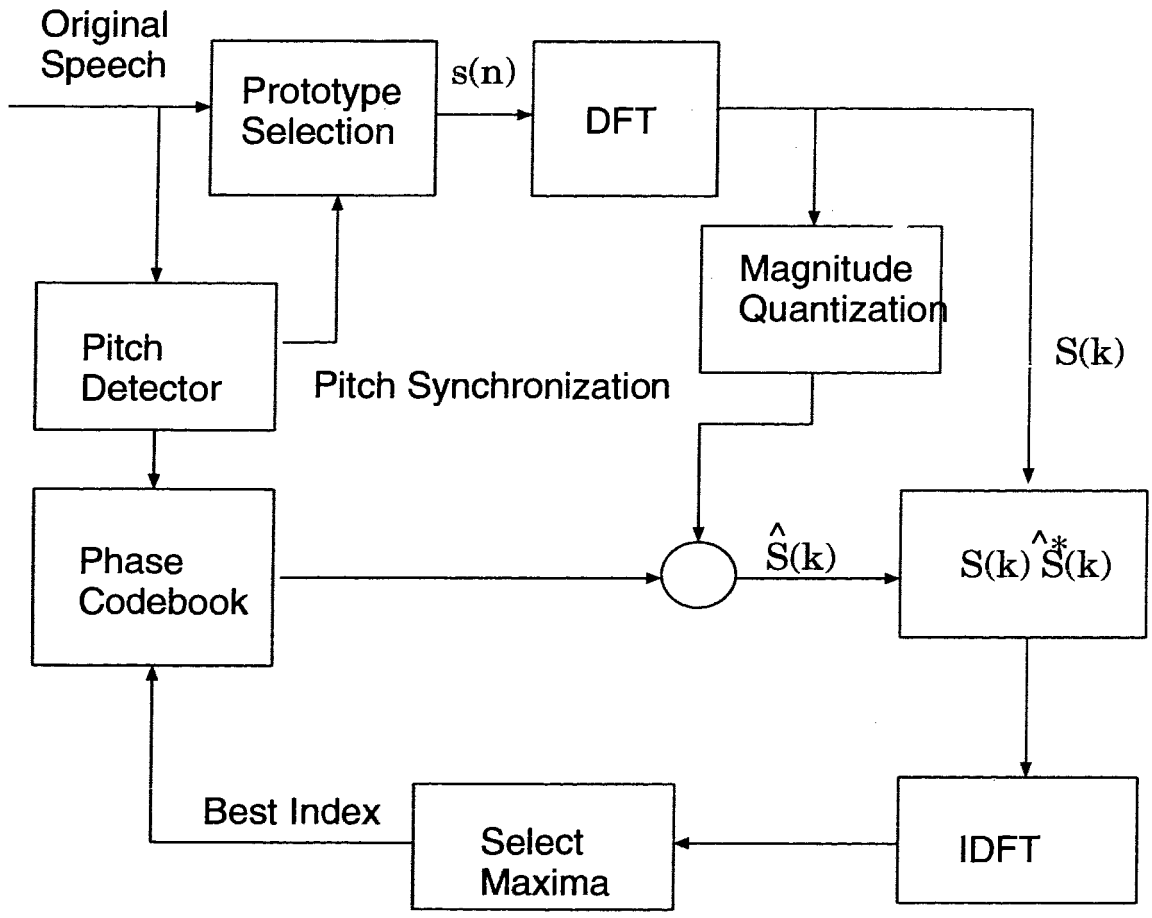

Figure 4.1: Block Diagram of The Phase Codebook Model

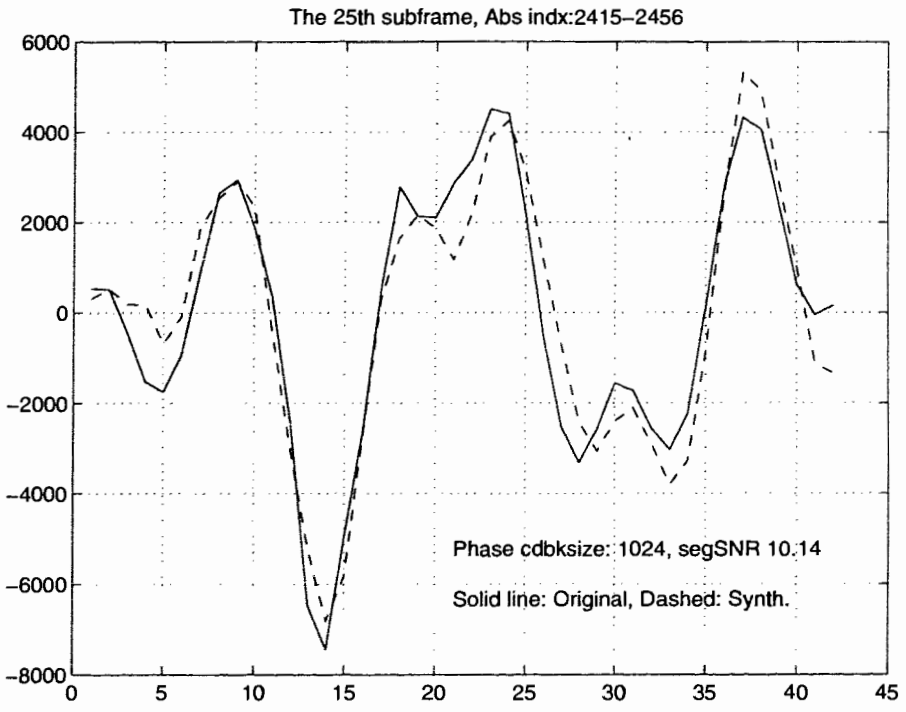

Figure 4.2: Prototype Waveforms Matched Using Phase Codebook

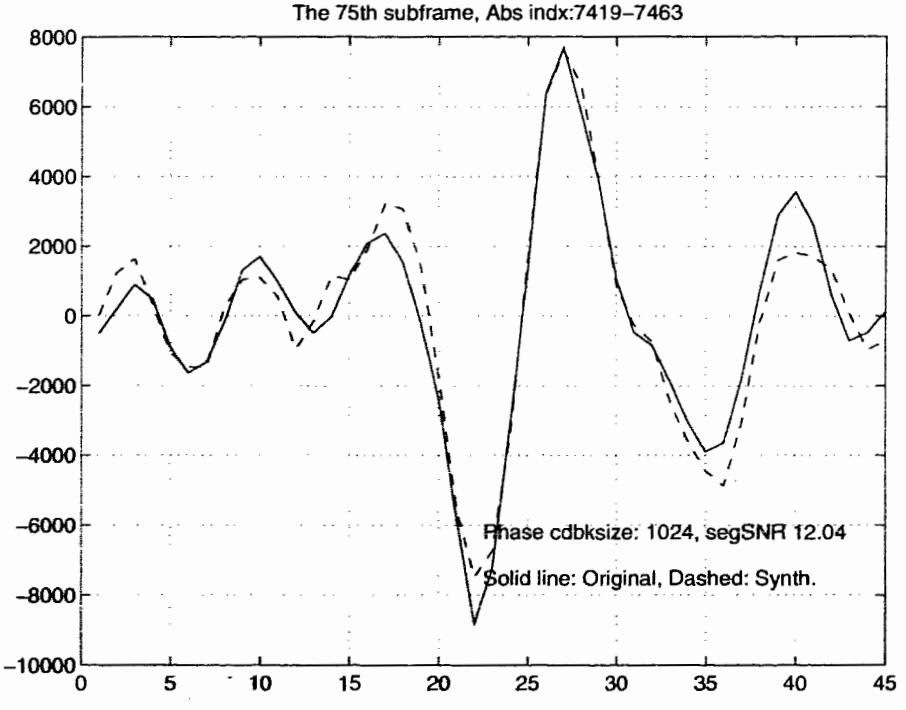

Figure 4.3: Prototype Waveforms Matched Using Phase Codebook

Figure 4.4 shows an example of the closed-loop waveform alignment performed by the phase codebook model. The two dashed lines are the synthesized prototype waveforms with and without alignment. The best alignment shift is obtained together with the best phase codebook entry during the phase codebook search procedure.

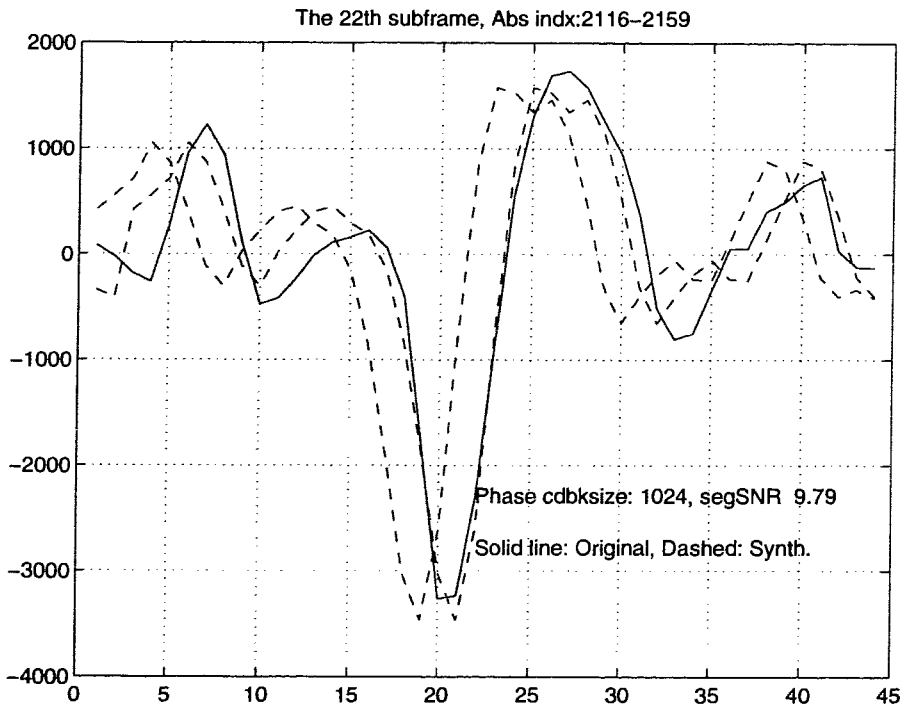

Figure 4.4: Waveform Alignment in the Phase Codebook

Quantization Parameter Set

For each prototype waveform, the quantization parameter set of the model is:

- pitch
- gain (energy)
- spectral magnitudes
- spectral phase index

- circular alignment shift

In many interpolative speech coders, the transmission of the circular alignment shift is not required. Sometimes, the gain can be quantized together with spectral magnitudes.

Prototype Waveform and Pitch Extraction

As shown in figure 4.1, pitch and prototype boundary detection algorithms have to be used to extract pitch-size prototype waveforms. Prototype boundary detection algorithms similar to those in PWI [30] (See section 3.1.2) and any pitch algorithms can be used in the proposed model. In our experiments, a modified version of a pitch detection algorithm [5] is used, and a simple minimum energy criterion is used to find the boundaries of a prototype waveform.

Quantization of Variable-Length Spectral Magnitude Vectors

The pitch value p varies in time. In the model, pitch-size DFTs generate spectral magnitude and phase vectors with variable dimensions. The variable dimension makes it difficult to directly take advantage of the benefits of vector quantization. This variable dimension problem appear also in direct waveform VQ (section 3.2.1) and in magnitude-only prototype waveform quantization (section 3.2.4). The magnitude-only quantization schemes in section 3.2.4, such as variable dimension vector quantization (VDVQ) [17] and non-square transform vector quantization (NSTVQ) [36], are applicable to the magnitude quantization in the phase codebook model. Differential coding similar to that in [54] is also applicable.

NSTVQ has been used for the quantization of spectral magnitudes in our exper-

iments. More details of NSTVQ and VDVQ are presented in section 4.5, where the performance of the phase codebook is evaluated.

Phase Codebook Generation

Two approaches have been experimented with to generate the phase codebook.

The first method uses a random variable with uniform distribution between $(-\pi, \pi)$. Prototype waveforms are real signals, hence their phase vectors are anti-symmetric. To generate a synthesized prototype waveform spectrum $\hat{S}(k)$, $0 \leq k < p$, the left half of the phase vector $\Phi(n)$, where $1 \leq n < \lfloor (p-1)/2 \rfloor$, is filled with random values. The right half of the phase vector is obtained using the anti-symmetric constraint. Since $S(0) = \sum_{n=0}^{p-1} s(n)$ and $s(n)$ is real, the phase of DC-term, $\Phi(0)$, has to be zero or π. The same zero or π constraint applies to $S(p/2)$ if p is even.

The second method employs phase vectors extracted from typical speech prototype waveforms. Similar to spectral magnitude vectors of prototype waveforms, phase vectors in the phase codebook obtained from real speech prototype waveforms are of variable length as well. There are a few ways to extend the shorter phase vectors. For example, padding random phase values or repeating a part of the phase vectors. It was found that the difference in performance between these methods is small. The reason seems to be that the low frequency part of phase vectors plays a more important role compared to the high frequency part.

The scheme based on extracted speech waveforms performs slightly better in SNR than the first one. There is no distinguishable difference between the two schemes in terms of speech quality.

Computational Complexity

The phase codebook model has high computational complexity. The phase codebook search procedure contributes most of the complexity. To quantize a prototype waveform, a full search of the phase codebook is required. For each phase codebook entry, a pitch-size IDFT is required to obtain the local maxima. DFT has complexity of $O(p^2)$. With a phase codebook size of 512, the worst possible pitch size of 150, and the prototype waveform update rate of 50 Hz, the computational complexity for each prototype has the order of

$$512 \times 150 \times 150 \times 50 = 576 MFLOPS$$

This complexity is extremely high, and is beyond the capacity of the current available general purpose digital signal processors such as TMS320 Family [26] [27] [59]. Also it is difficult to implement arbitrary-size DFT algorithms in hardware.

Applying fast Fourier transform (FFT) algorithms seems to be an straightforward solution to reduce the complexity. However, the pitch value p varies in time, so does the length of prototype waveforms. A pitch-size DFT is required by the phase codebook model, which prevents us from applying fast Fourier transform (FFT) algorithms directly.

However, by employing the concept of oversampling (or upsampling), the most expensive DFT computation can be replaced by an FFT in the inner loop of the codebook search procedure, resulting in a modified phase codebook model. The modified model is described later in this chapter.

Discussion

In the *prototype waveform interpolation* (PWI) method [30] (see section 3.2.2), Kleijn decomposes a prototype waveform into a set of Fourier series coefficients. Similarly, Burnett and Bradley [9] decompose a prototype waveform into a set of DFT coefficients. These decompositions schemes either encode the real-part and imaginary-part of Fourier coefficients or encode *sine*-part and *cosine*-part of the coefficients. This implies that the phase information is encoded together with spectral magnitudes.

On the other hand, the proposed model decomposes a prototype waveform into a spectral magnitude vector and a phase vector. The magnitude and phase are completely separated. This complete separation allows us to employ the best magnitude quantization techniques (see section 3.2.4) to quantize spectral magnitudes of a prototype waveform. The phase information, on the other hand, is quantized using the phase codebook model.

The proposed model performs waveform alignment and phase codebook search simultaneously, and it is much more computationally efficient compared to alignment in the time domain employed by PWI. More importantly, the proposed model performs closed-loop waveform alignment whereas separate open-loop waveform alignment is always required in PWI.

The model can be used on either original or residual speech. Also it is suitable for either a passband or full-band speech signal.

4.3 Oversampling of Prototype Waveforms

As mentioned in the previous section, the basic phase codebook model has high computational complexity. A pitch-size DFT is required in the basic model, and FFT algorithms cannot be directly used. However, if prototype waveforms are oversampled to a fixed length, the expensive DFT computation can be replaced by an FFT in the inner loop of the phase codebook search procedure. Furthermore, oversampling provides not only higher resolution in prototype waveform alignment but also efficient ways of interpolation between prototype waveforms.

This section analyzes in detail three different schemes of oversampling prototype waveforms, and these schemes are the basis for the discussion of the modified phase codebook model in the next section. The three schemes for oversampling are:

- Polyphase filtering in the time domain
- All-pass filtering with fractional shifts in the frequency domain
- Zero-padding in the frequency domain

These schemes are different in computational complexity and in operation as well. However, under certain constraints, they are equivalent in terms of oversampling or interpolation[1] results. These constraints includes strict periodicity of the processed signal, and an ideal low-pass filtering operation. In practice, polyphase filtering introduces interpolation errors because of non-ideal low-pass filtering operation, while the other two schemes perform ideal interpolation on periodic signals. When used before extracting prototype waveforms, polyphase filtering introduces

[1]The term *interpolation* used here and in the rest of this chapter is equivalent to "oversampling". It is commonly used in signal processing literature. But it is different from the "interpolation" between prototype waveforms in the previous chapter

less distortion to the boundaries of prototype waveforms because the speech signal is not strictly periodic.

4.3.1 Polyphase Filtering

Increasing the sampling rate (oversampling) of a signal $x(n)$ by an integer factor L implies that we must interpolate $L-1$ new sample values between each pair of sample values of $x(n)$. From the Nyquist sampling theorem, the ideal interpolation can be performed using the following two steps:

1. The input signal $x(n)$ is "filled in" with $L-1$ zero-valued samples between each pair of samples of $x(n)$, resulting a new signal $w(n)$.
2. The signal $w(n)$ is filtered by an ideal low-pass filter $h_I(n)$ with a cut-off frequency of $\pm\pi/L$.

The signal $y(n)$ obtained by these two steps is an ideal oversampled version of $x(n)$ [46].

The well-known interpolation procedure—polyphase filtering—is theoretically the same as the two-step interpolation procedure described above. However, these two procedures differ in their implementation structures and computational complexity: polyphase filtering has a more efficient structure, and consequently it requires L times less computation to obtain $y(n)$ [14] [15] [56] [57].

The brief derivation of polyphase filtering in this subsection follows [14]. For more details, the reader is referred to [14], [15], [56].

For a *1-to-L* interpolator, there are L polyphase filters and they can be defined as

$$P_\rho(n) = h(\rho + nL), \rho = 0, 1, 2, \ldots, L-1, and\ all\ n \tag{4.3}$$

where $h(k)$ is a low pass filter. Given an input signal $g(n)$, polyphase filtering is equivalent to applying $h(k)$ to filter a sequence $\hat{g}(n)$ defined as

$$\hat{g}(k) = \begin{cases} L\ g(k/L) & k = 0, \pm L, \pm 2L, \ldots \\ 0 & otherwise \end{cases} \tag{4.4}$$

An ideal interpolator filter $h(k)$ must approximate the ideal low pass characteristics defined as

$$H_I(\omega') = \begin{cases} L & |\omega'| \leq \pi/L \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

where the frequency variable ω' refers to $h(k)$, and the subscript I means the filter is ideal. Assuming the frequency variable $\omega = \omega' L$ refers to the polyphase filters $P_\rho(n)$, it is possible to derive the equivalent ideal characteristic $P_{\rho,I}$ required in the polyphase filters

$$\begin{aligned} P_{\rho,I} &= \frac{1}{L} e^{j\omega\rho/L} H_I(\omega/L) \\ &= e^{j\omega\rho/L},\ \rho = 0, 1, 2, \ldots, L-1 \end{aligned} \tag{4.6}$$

Eqn. (4.6) shows that the ideal polyphase filters $P_{\rho,I}$ should approximate all pass filters with linear phase shifts corresponding to fractional advances of ρ/L samples.

If we consider the interpolation filter in the time domain, we obtain an alternative picture of the ideal interpolation filter. By taking the inverse transform of the ideal filter characteristic defined by eqn. (4.6), we get the well-known $sin(x)/x$ characteristic

$$h_I(k) = \frac{\sin(\pi k/L)}{\pi k/L}, k = 0, \pm 1, \pm 2 \pm 3 \ldots \tag{4.7}$$

Either by sampling eqn. (4.7) or by taking the inverse transform of eqn. (4.5), we obtain the ideal time response of the polyphase filters

$$P_{\rho,I}(n) = \frac{\sin(\pi(n+\rho/L))}{\pi(n+\rho/L)}, \quad \rho = 0,1,\ldots,L-1, and\ all\ n \tag{4.8}$$

Since these filters are theoretically infinite in duration, they must be approximated, in practice, with finite-duration filters. Thus the interpolation "error" between the outputs of a practical system and an ideal system will always be non-zero except the output of the $zero^{th}$ polyphase filter $P_0(n)$, where the output is known to be equal to the input [14].

4.3.2 Interpolation of Periodic Signal in the Frequency Domain

If interpolation is required for a periodic signal, the interpolation "error" introduced by practical polyphase filters can be avoided using a DFT based approach derived as following.

Let $x(n)$ be a period of a period of a periodic signal $\tilde{x}(n)$ with infinite duration

$$\tilde{x}(n) = \tilde{x}(n+P) \quad -\infty < n < \infty \tag{4.9}$$

$$x(n) = \tilde{x}(n) \quad 0 \le n < P \tag{4.10}$$

where P is the period of the periodic signal. An *1-to-L* interpolation for $x(n)$ can be obtained by applying a set of all-pass filters with a fractional shift of $l/L, (l = 0,1,\ldots,L-1)$ samples.

$$H_l(k) = \begin{cases} e^{j2\pi kl/(LP)} & 0 \le k \le \lfloor (P-1)/2 \rfloor \\ 0 & P/2 \text{ if } P \text{ is even} \\ e^{-j2\pi(P-k)l/(LP)} & \lceil (P+1)/2 \rceil \le k < P \end{cases} \tag{4.11}$$

Note that eqn. (4.11) and eqn. (4.5) are similar. However, eqn. (4.11) is defined in the discrete frequency domain (or the DFT domain) whereas eqn. (4.5) is defined in the continuous frequency domain. And more importantly, as we will see in the following, eqn. (4.11) is implementable in practice, and can perform ideal interpolation.

The concept of the DFT domain interpolation is shown in figure 4.5. In this figure, $x(n)$, where $0 \leq n < P$, is a cycle of the periodic signal $\tilde{x}(n)$ defined by eqn. (4.9). Given $x(n)$, an oversampled version of $x(n)$, denoted by $y(n)$, is obtained in two steps:

1. obtain L intermediate sequences $x_l(n)$, where $0 \leq l < L$ and $0 \leq n < P$, from $x(n)$. Each sequence $x_l(n)$, where $0 \leq n < P$, has length P.

2. obtain $y(n)$ by "interleaving" these L sequences $x_l(n)$.

Figure 4.6 shows how the L intermediate sequences $x_l(n)$ are obtained and how the "interleaving" is performed. In figure 4.6, $X(k)$, where $0 \leq k < P$, is the DFT of $x(n)$,$0 \leq n < P$. By multiplying $X(k)$ and interpolators $H_l(k)$, we obtain $X_l(k)$ as:

$$X_l(k) = X(k)H_l(k), \quad 0 \leq l < L; \quad 0 \leq k < P \tag{4.12}$$

where $H_l(k)$ are the interpolators defined in eqn. (4.11). Note that $X_l(k)$ have the same magnitudes as $X(k)$, but with extra fractional phase shifts determined by $H_l(k)$. Taking the IDFT of $X_l(k)$, we have $x_l(n)$ as

$$x_l(n) = \frac{1}{P}\sum_{k=0}^{P-1} X_l(k)e^{\frac{j2\pi nk}{P}} \quad 0 \leq n < P; \quad 0 \leq l < L \tag{4.13}$$

Finally, the desired $y(n)$, the ideal oversampled version of $x(n)$, is obtained by "interleaving" the set of sequences $x_l(n)$

$$y(m) = y(nL + l) = x_l(n), \quad 0 \leq n < P; \quad 0 \leq l < L \tag{4.14}$$

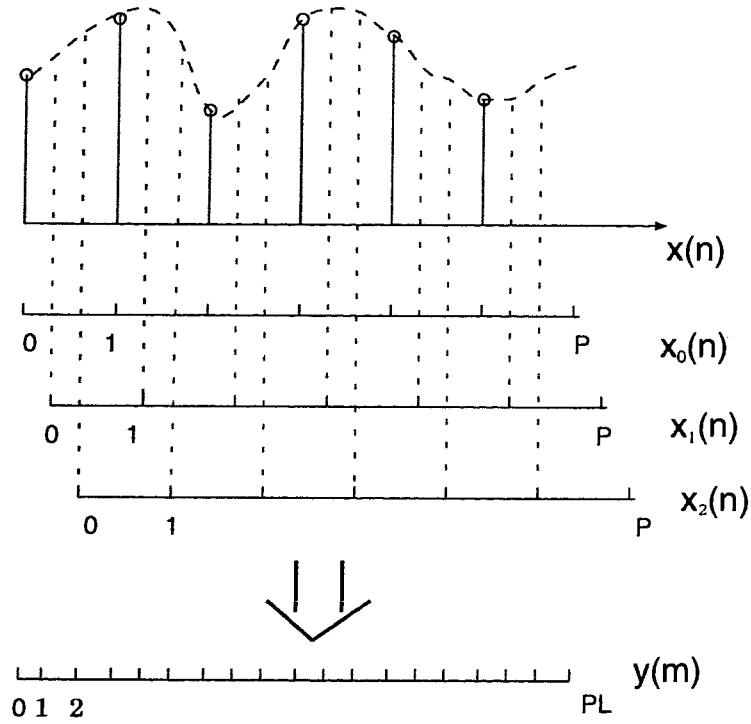

Figure 4.5: The Concept of Interpolation in the Frequency Domain (L = 3)

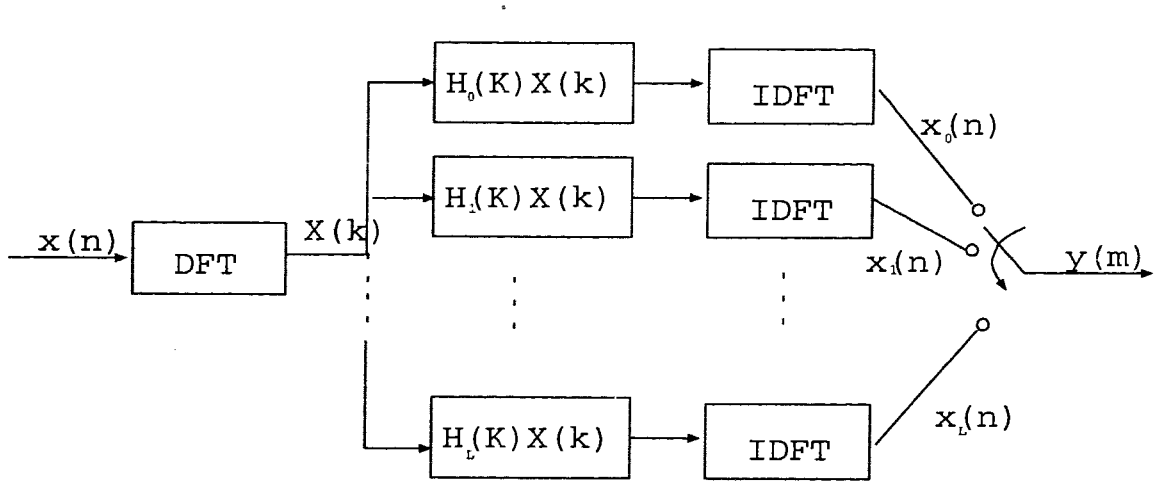

Figure 4.6: The Block Diagram of the Frequency Domain Interpolation

where

$$m = nL + l, \;\; 0 \leq l < L \;\; 0 \leq n < P \tag{4.15}$$

Based on the Nyquist sampling theorem, it can be shown that by applying the interpolators $H_l(k)$, as defined by eqn. (4.11), the above procedure (figure 4.6) performs ideal interpolation under the constraint that if P is even, $X(P/2)$ has to be zero. However, this constraint is normally satisfied by a speech (or residual) signal because $X(P/2)$ is the magnitude value at exact half of the Nyquist sampling rate, and it is usually filtered out by the anti-alias filter before analogue-to-digital (A/D) conversion. The reader is referred to Appendix A for detailed derivation of the interpolation procedure.

In the Appendix A, it is shown that if the interpolator $H_l(k)$ are applied to $x(n)$ to obtain an oversampled version $y(n)$, then DFTs of $x(n)$ and $y(n)$, $X(k)$ and $Y(k)$, satisfy

$$Y(k) = \begin{cases} L\,X(k) & 0 \leq k \leq \lfloor (P-1)/2 \rfloor \\ L\,X((k))_p & P(L-1) + \lceil (P+1)/2 \rceil \leq k < PL \\ 0 & otherwise \end{cases} \tag{4.16}$$

where $((k))_p$ means modulo p. Eqn.(4.16) shows that the low frequency part of $Y(k)$ is equal to $X(k)$ within a scaling factor L while the high frequency part is equal to zero.

As an example, let us look at the interpolation procedure where the interpolation factor $L = 2$. Given a signal $x(n)$, and its DFT, $X(K)$, we perform multiplication in the frequency domain using eqn. (4.12):

$$X_0(k) = H_0(k)X(k), \quad O \leq k < P \tag{4.17}$$

$$X_1(k) = H_1(k)X(k), \quad 0 \leq k < P \tag{4.18}$$

Taking inverse DFT, we has

$$x_o(n) = IDFT(X_0(k)), \quad 0 \leq n < P \tag{4.19}$$

$$x_1(n) = IDFT(X_1(k)), \quad 0 \leq n < P \tag{4.20}$$

Note that if $X(k)$ satisfy the constraint ($X(P/2) = 0$ in case P is even), then

$$X_0(k) = X(k) \quad 0 \leq k < P \tag{4.21}$$

$$x_0(n) = x(n) \quad 0 \leq n < P \tag{4.22}$$

Applying eqn. (4.14), the ideal *1-to-2* interpolated signal of original signal $x(n)$ becomes

$$y(n) = \begin{cases} x(n/2) & \text{n is even} \\ x_1((n-1)/2) & \text{n is odd} \end{cases} \quad 0 \leq n < 2P \tag{4.23}$$

The relation between $X(k)$ and $Y(K)$ in eqn. (4.16) is shown in figure 4.7. It is interesting to note that $X_0(k)$ and $X_1(K)$ add up in the low frequency region, and cancel out in the high frequency region to form $Y(k)$. This phenomenon exists in interpolation where $L > 2$, though in a more complicated way defined by eqn.(4.16).

4.3.3 Zero-padding in the Frequency Domain

Zero-padding in the DFT domain is naturally related to the phase shifting interpolator $H_l(k)$. It is actually an extension of the phase shifting interpolator.

Suppose we have a prototype waveform with pitch length $P = 50$, and want to oversample it to a fixed length of $N = 128$. In signal processing literature, this is called a *50-to-128* interpolation or *sampling rate conversion* by a rational factor 50/128. The phase shifting interpolator $H_l(k)$ cannot be applied directly due to the fact that 128/50 is not an integer value. However, a two-step procedure can perform arbitrary *P-to-N* interpolation.

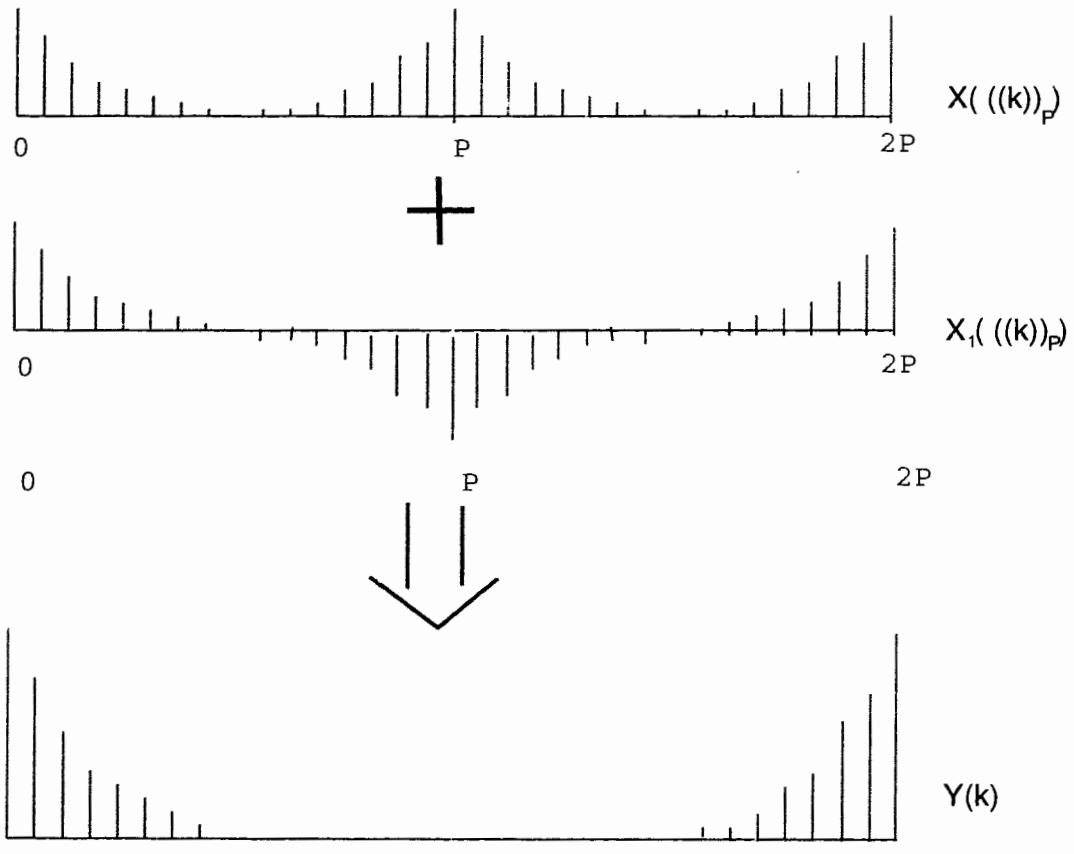

Figure 4.7: The Relation between Y(k) and X(k)(L = 2)

Letting (P, N) be the greatest common divisor of P and N, and letting $x(n)$, where $0 \leq n < P$, be a cycle of a periodic signal defined by eqn. (4.9), the following two steps are required to obtain $y(n)$—an oversampled version of $x(n)$:

1. Perform *1-to-N/(P,N)* interpolation on $x(n)$, resulting in an intermediate signal $w(n), 0 \leq n < pN/(P, N)$
2. Perform *p/(P,N)-to-1* decimation on $w(n)$ to obtain $y(n), 0 \leq n < N$

Now let us look at the two-step procedure in the frequency domain. **Step** 1 pads zeros in the middle of $X(k)$, the DFT of $x(n)$, to obtain $W(k)$, the DFT of $w(n)$. On the other hand, **Step** 2 takes out zeros right in the middle of $W(k)$ to obtain $Y(k)$. As long as $N > P$, the shape non-zero part of spectrum $Y(k)$ is the same as $X(k)$ within a scaling factor. Combining these two steps, we obtain the *zero-padding in the DFT domain* interpolation scheme as following:

Simply by padding $N - P$ zeros right in the middle of $X(k), 0 \leq k < P$, and scaling $X(K)$ by a factor of N/P, we can obtain Y(k) as

$$Y(k) = \begin{cases} N/P\ X(k) & 0 \leq k \leq \lfloor (P-1)/2 \rfloor \\ 0 & \lfloor (P-1)/2 \rfloor < k < \lceil (P+1)/2 \rceil + N - P \\ N/P\ X(k-N+P) & \lceil (P+1)/2 \rceil + N - P \leq k < N \end{cases} \quad (4.24)$$

The IDFT of $Y(k)$, $y(m), 0 \leq m < N$ is therefore an oversampled version of $x(n), 0 \leq n < P$. The interpolation will be ideal under the same constraint as the interpolator $H_l(k)$ scheme. The constraint is that $X(P/2) = 0$ when P is even.

This zero-padding procedure can perform arbitrary *P-to-N* interpolation on $x(n)$. By combining the derivation of the interpolation scheme based on $H_l(k)$ in the previous subsection and the two-step procedure in this subsection, we can easily see the validity of the zero-padding scheme.

Figure 4.8 shows an example of *P-to-N* interpolation using the zero-padding scheme. The solid line presents an original prototype waveform with $P = 40$, and the dashed line is the oversampled prototype waveform with $N = 128$. Their corresponding spectral magnitudes are shown in figure 4.9. These two magnitude spectra are equivalent with a scaling factor at the low frequency part of $Y(k)$.

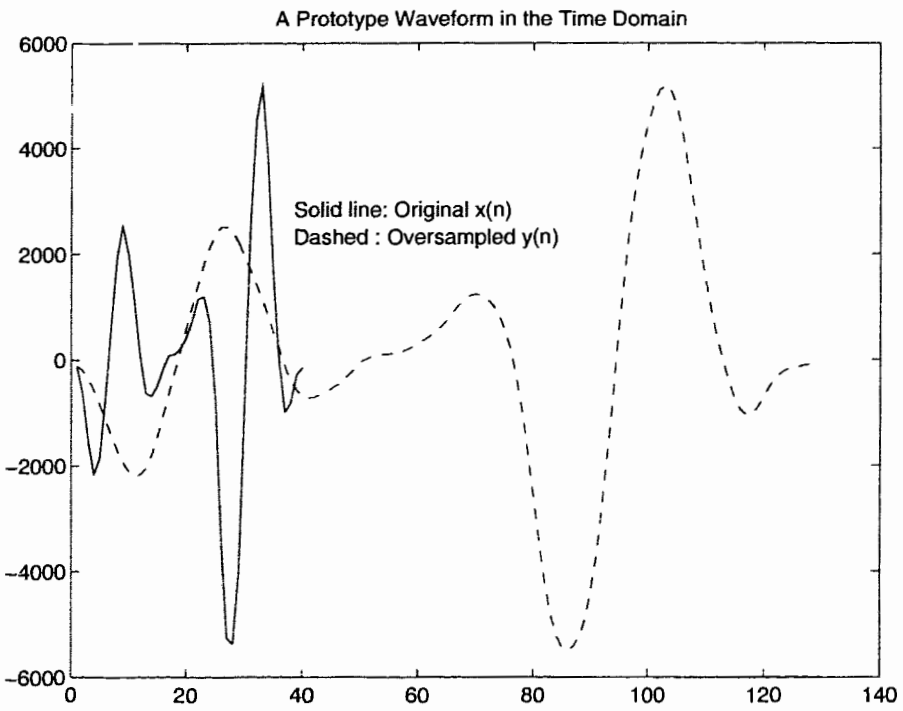

Figure 4.8: Original and Oversampled Waveforms

4.3.4 Comparison of the Three Interpolation Schemes

In the previous subsections, we have discussed three possible techniques to oversample prototype waveforms, namely, polyphase filtering, phase-shifting interpolators $H_l(k)$, and zero-padding in the DFT domain.

Polyphase filtering is the only interpolation technique applicable to non-periodic signals. Phase-shifting interpolators and zero-padding can only apply to a period in

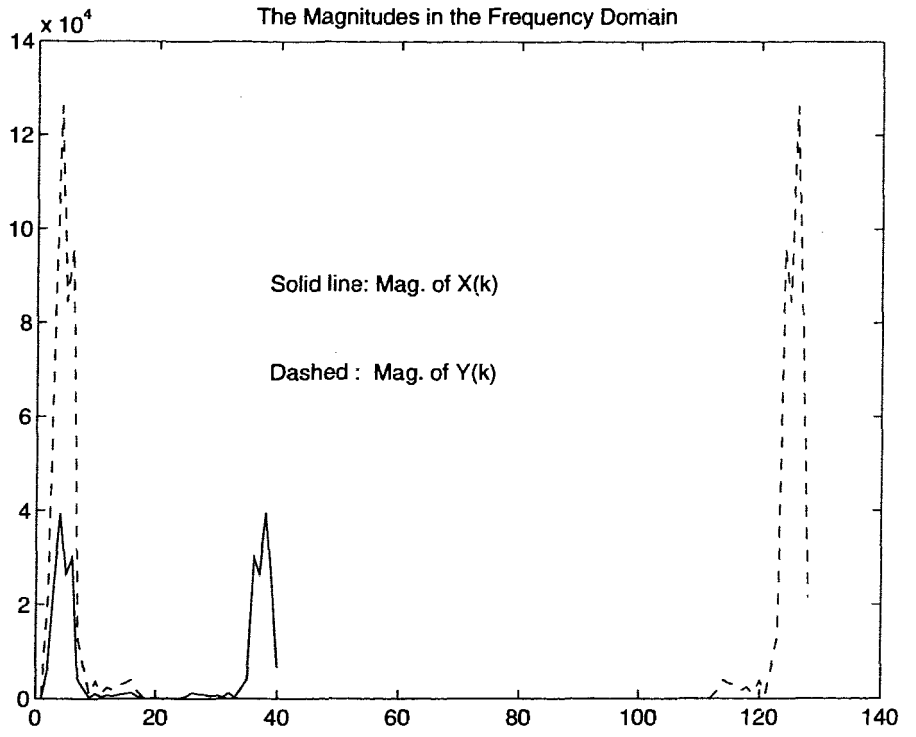

Figure 4.9: The Magnitude Spectra of Original and Oversampled Waveforms

a periodic signal. However, polyphase filtering is computationally expensive. The other two schemes are more efficient because the convolution operations of polyphase filtering in the time domain become multiplications in the frequency domain.

Voiced speech or residual signals are not strictly periodic. As we have discussed in chapter 3, the shape and length (pitch) of prototype waveforms evolve with time. Hence, polyphase filtering has advantage over the other two schemes when we want to extract prototype waveforms from a speech or residual signal. For example, in [30], input speech signal is oversampled by a factor of 10, and the prototype waveforms are extracted from the interpolated signal. The oversampling enables the prototype waveform extraction algorithms to locate the boundary of prototype waveforms more precisely.

However, the polyphase filtering is unable to perform arbitrary *P-to-N* sampling rate conversion when the pitch P varies in time. To oversample prototype wave-

forms to a fixed-length N using polyphase filters, first we have to perform *1-to-M* interpolation (M is an integer and $PM \gg N$) on a speech or residual signal. Then we assume that the oversampling rate is so high that consecutive samples of the oversampled signal do not change much, and the value of any point between two consecutive samples can simply be replaced by either of the two consecutive samples. This method of warping a prototype waveform from length P to N introduces interpolation errors because of the above assumption. Also, ideal polyphase filters in eqn.(4.8) are theoretically infinite in duration, and have to be approximated, in practice, with finite-length filters. To reduce the interpolation error, we can increase interpolation rate M, and use long polyphase filters. However, this greatly increases complexity of filtering operations (convolution).

Both phase-shifting interpolators $H_l(k)$ and zero-padding in the frequency domain provide ideal interpolation for a periodic signal. Interpolators $H_l(k)$ is identical to the zero-padding scheme in case that N/P is an integer. However, the pitch P of prototype waveforms varies, and N/P cannot alway be an integer. The interpolation scheme using $H_l(k)$ has to employ a two step procedure—*1-to-N/(P,N)* oversampling followed by *P/(P,N)-to-1* decimation (See section 4.3.2)—to obtain an arbitrary *P-to-N* rate conversion.

The zero-padding scheme has the flexibility to easily oversample prototype waveforms with different pitch values to a fixed length, and it is computationally efficient. The potential disadvantage of it is that the prototype waveforms are assumed to exact periodic.

4.4 The Modified Phase Codebook Model

This section describes an improved phase codebook model in terms of computationally complexity using the oversampling techniques discussed in the previous section.

4.4.1 Description of the Modified Phase Codebook Model

The most computational intensive part of the model is the phase codebook search procedure. As we can see in the basic model shown in figure 4.1, each phase codebook entry requires one pitch-size IDFT.

Now suppose the original prototype waveform is $x(n)$, where $0 \leq n < P$. Applying one of interpolation schemes in the previous section, we have the oversampled waveform $y(n)$, where $0 \leq n < N$, and $N \geq P$.

The minimization procedure in eqn. (4.1) is applicable to the difference between $y(n)$ and the oversampled synthesized waveform $\hat{y}(n)$. The cross-correlation term $z(m)$ in eqn. (2.30) becomes

$$\begin{aligned} z(m) &= \sum_{n=0}^{N-1} y(n)\hat{y}((n-m))_N \\ &= \frac{1}{N}\sum_{k=0}^{N-1} Y(k)\hat{Y}^*(k)e^{j2\pi mk/N} \quad 0 \leq m < N \end{aligned} \tag{4.25}$$

Note that N is fixed for all prototype waveforms, and proper IFFT algorithms can be used. This reduces the computational complexity by a factor of $P^2/(N \log N)$.

The research on FFT algorithms is comprehensive [10], FFT algorithms based on index mapping and polynomial algebra such as *Cooley-Tukey FFT*, *the split-radix FFT*, *the prime factor algorithm* (PFA), and *Winograd Fourier transform algorithm* (WFTA), provide efficient calculation of DFT. They can readily be used in this

modified model.

Oversampling of prototype waveforms not only provides computational efficiency of the model, but also provides better prototype waveform alignment because of better resolution in the time domain (from P to N samples for each prototype waveforms).

The block digram of the modified phase codebook model is shown in figure 4.10. In order to exploit the advantages of different oversampling schemes, both polyphase filtering and zero-padding schemes can be used at different stages of the modified model.

Spectral Magnitude Quantization

The effect of the modified model on the spectral magnitudes can be clearly seen in figure 4.9. As we can see in the figure, oversampling prototype waveforms in the time domain only has an scaling effect on prototype spectral magnitudes. The zeros at the high frequency part of $Y(k)$ can simply be ignored in the quantization procedure. Hence, the techniques discussed in section 4.2 are applicable to the modified model.

Phase Codebook Generation

In the modified phase codebook model, speech prototypes are interpolated to facilitate the use of FFT algorithms in the phase codebook search procedure. It has been shown in section 4.3 that oversampling in the time domain has only a scaling effect on the spectral magnitudes, but no effect on the phase vector. Hence, the same phase codebook used in the original model can be used in the modified model

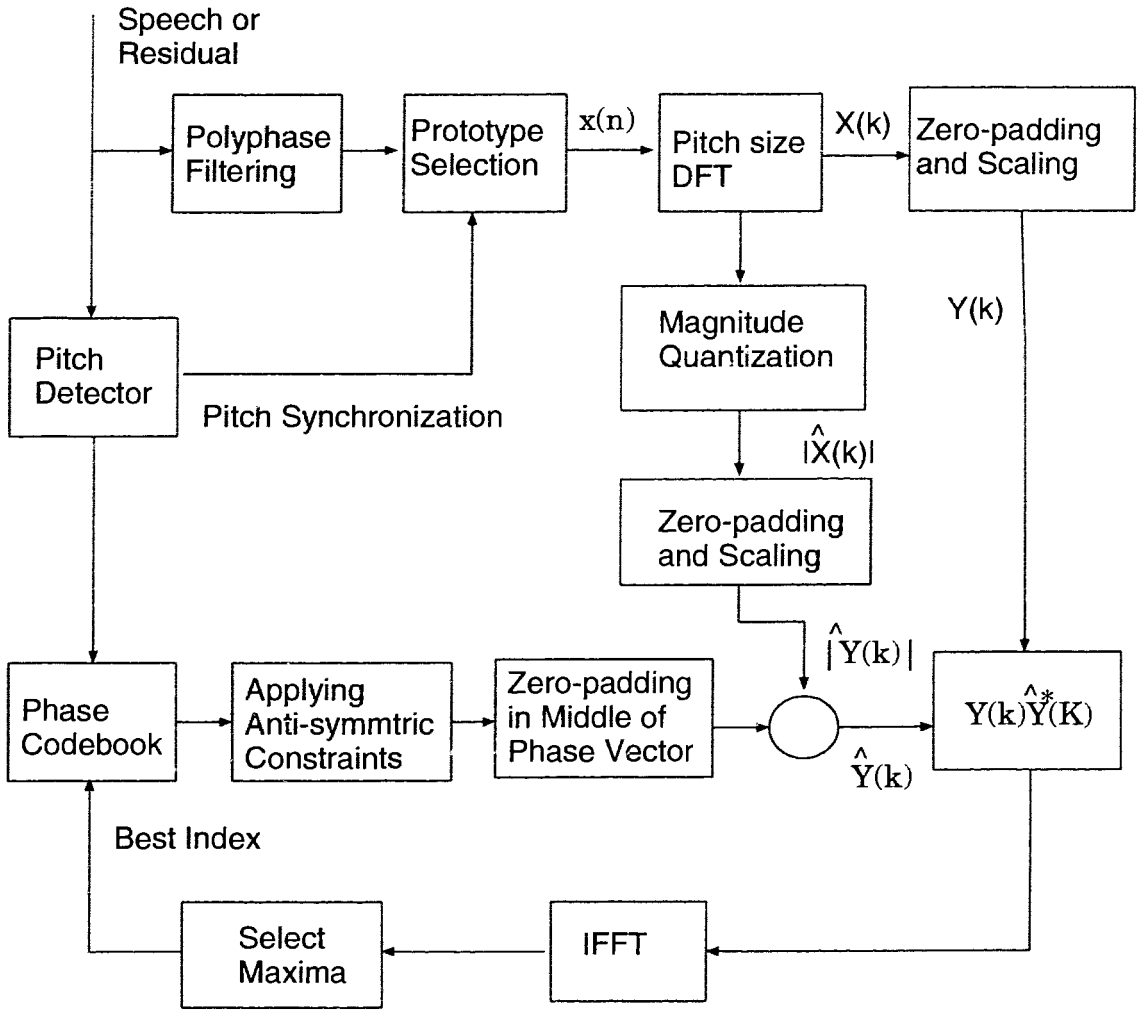

Figure 4.10: Block Diagram of the Modified Phase Codebook Model

with a slight modification. The slight modification is that in the modified model, $N - P$ zeros are padded right into the middle to an original phase vector of size P, where N is the size of FFT.

4.5 Quantization Performance of The Model

In order to validate the proposed phase codebook model, this section compares the quantization performance of the model with that of two direct waveform VQ schemes. The first direct VQ scheme employs variable dimension vector quantization (VDVQ) [17] to handle the variable length. The second method oversamples prototype waveforms to a fixed length, and then applies split VQ to the fixed length waveforms.

The next two subsections describe these two reference systems, following by a subsection with the details of magnitude quantization in the phase codebook model. Finally, the performances of these systems are compared.

4.5.1 Reference System 1: Variable Dimension VQ

Variable dimension VQ (VDVQ) was originally used for the quantization of spectral magnitudes in sinusoidal coding [17].

Given an input vector $\mathbf{S}$ with a variable dimension P, the encoder first maps $\mathbf{S}$ into a N dimensional extended vector $\mathbf{X}$ by assigning the P components of $\mathbf{S}$ to the components of $\mathbf{X}$. Note that $\mathbf{X}$ has N ($N \geq P$) components, and only P out of N components in $\mathbf{X}$ are assigned with components of $\mathbf{S}$ while the others are assigned

with zeros. The exact values assigned to $\mathbf{X}$ are decided as follows:

$$\mathbf{X}[k] = \begin{cases} \mathbf{S}[j], & \text{if } k = \lceil \frac{jN}{P} \rceil; \quad \text{j any positive integer} \\ 0 & otherwise \end{cases} \tag{4.26}$$

for $1 \leq k \leq N$. In a similar way, a *selection vector* $\mathbf{Q}$ is defined as

$$\mathbf{Q}[k] = \begin{cases} 1 & \text{if } k = \lceil \frac{jN}{P} \rceil; \quad \text{j any positive integer} \\ 0 & otherwise \end{cases} \tag{4.27}$$

for $1 \leq k \leq N$.

Let us define a codebook with each of its code vectors, $\mathbf{Y}_j$, having dimension N. To quantize the input vector $\mathbf{S}$, we define the distortion measure between $\mathbf{S}$ with its associated selector vector $\mathbf{Q}$ and a code vector $\mathbf{Y}_j$ in the codebook. This measure is based on matching the input vector $\mathbf{S}$ components to the corresponding subset of the code vector $\mathbf{Y}_j$. Thus,

$$d(\mathbf{X}, \mathbf{Y}_j) = \frac{1}{P} \sum_{k=1}^{N} \mathbf{Q}[k]\, d_1(\mathbf{X}[k], \mathbf{Y}_j[k]) \tag{4.28}$$

where $d_1(s, y)$ is a specified distortion between two scalars s and y.

The codebook is iteratively designed in a manner similar to the usual generalized Lloyd algorithm (GLA) described in section 2.2. More details about the nearest neighbor rule and centroid rule can be found in [17].

4.5.2 Reference Systems 2: Oversampling Waveforms in the time Domain

The second reference system performs waveform quantization in two steps: (1) oversamples variable dimensional input vectors to a fixed length and (2) applies ordinary VQ to the fixed-length oversampled vectors. The oversampling schemes described in section 4.3 are used.

4.5.3 Quantization using the Phase codebook model

In the proposed phase codebook model, NSTVQ is used for quantization of variable-length spectral magnitude vectors.

NSTVQ for Spectral Magnitude Quantization

Because of the variable length problem of spectral magnitude vectors, usual waveform quantization schemes cannot be directly used in the quantization of spectral magnitude vectors. However, the schemes we discussed in section 4.2 are applicable. In our experiment, *non-square transform vector quantization* (NSTVQ) [36] [35] is used. Following is a brief description of NSTVQ. More detail can be found in [35].

The block diagrams of the NSTVQ encoder and decoder are shown in figure 4.11. There are two steps in the NSTVQ approach: (1) convert a magnitude vector S which has a variable dimension P, to a vector X with fixed-dimension N (2) vector quantize the fixed-length vector X using an ordinary vector quantizer.

In the first step, the variable vector dimension P is used by the switch, shown in the figure, to select from a set of L non-square transformation matrices which are fixed and known at both the encoder and decoder. The selected matrix, $\mathbf{B}_l$, is called the forward transformation matrix and has dimension $N \times P$. The variable-length vector is transformed into a fixed-length vector $\mathbf{X}$ using the following equation:

$$\mathbf{X} = \mathbf{B}_l\mathbf{S} \tag{4.29}$$

In the second step, the N-dimensional vector $\mathbf{S}$ is vector quantized using a ordinary fixed-length vector quantizer.

In the decoding process, the input vector dimension P is required. However, the

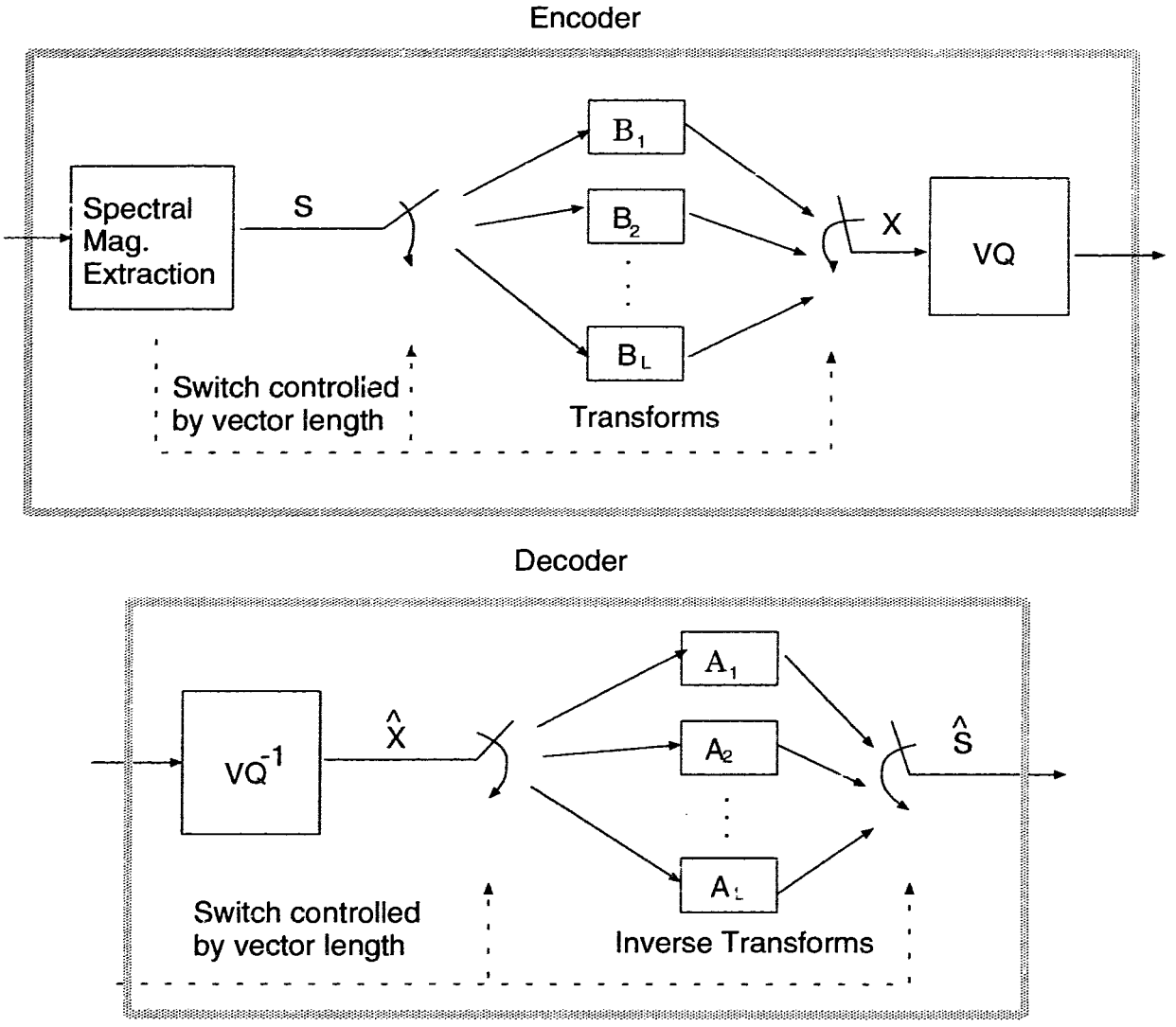

Figure 4.11: The Block Diagram of Non-Square Transform Vector Quantization (NSTVQ)

value P is the pitch value which is always transmitted to the decoder, therefore no extra bits are required for vector dimension transmission. The quantized magnitude vector $\mathbf{S}$ of dimension P is recovered using the following equation:

$$\hat{\mathbf{S}} = \mathbf{A}_l \hat{\mathbf{X}} \tag{4.30}$$

When the dimension of vector S is larger that the fixed length vector dimension N, there will be distortion between the original and reconstructed vectors even when the fixed-length vectors are not quantized. The distortion due to this dimension conversion is called modeling distortion. The forward transformation matrix which minimizes the modeling distortion is simply the transpose of the inverse transformation matrix for the corresponding vector dimension.

In [35], a few choices of transformation matrices are presented such as Karhunen-Loeve Transform (KLT) matrix, first and second forms of the discrete cosine transform (DCT-I) and (DCT-II), a transform from orthogonal polynomial basis functions (OPT), and the discrete Hartley transform (DHT). Based on results in [35], DCT-II is chosen as the transform in our experiments.

4.5.4 Comparison Between the Model and Reference Systems

The proposed model is compared with two direct waveform-quantization reference schemes described in the previous subsection.

Table 4.1 compares the segmental signal-to-noise ratio (segSNR) of the proposed model with those of the direct waveform VQ schemes. These systems are compared at a high rate of 72 bits/prototype and a low rate of 36 bits/prototype. The specific

numbers of bits, 72 and 36, are chosen because of their relative easiness in bit assignment for all of these systems.

In the first reference system, variable dimension VQ (VDVQ) is used. A variable dimension prototype waveform is mapped into a fixed-length($N = 256$) vector using eqns. (4.26) and (4.27). The extended vector is then split into 4 (36 bits/prototype case) or 8 subvectors (72 bits/prototype case). These subvectors are quantized using a split VQ (eqn.(4.28) is the distortion criterion).

In the second reference system, a variable dimension prototype waveform is oversampled to a fixed length ($N = 256$). Then each extended vector is split into 4 subvectors (36 bits/prototype case) or 8 subvectors (72 bits/prototype case), and quantized by an ordinary split VQ with 9-bit codebooks.

In the phase codebook model, the spectral magnitudes are quantized using multistage NSTVQ (MS-NSTVQ) described in the previous subsection. A 4-*stage* NSTVQ codebook with 6 bits/stage is used for the 36-bit case while a 10-*stage* NSTVQ codebook with 6 bits/stage for the 72-bit case. The results of table 4.1 indicate that the proposed system outperforms both of the reference schemes by a wide margin.

A significant part of the performance gain in table 4.1 is due to the implicit prototype waveform alignment in the proposed system. The transmission of the alignment information would require about 8 bits (in many PWI systems this information does not need to be transmitted). To compensate for the effect of alignment, Table 4.2 shows a comparison in which the direct VQ systems are allocated 8 bits more than the proposed system. The results show that the proposed system still preserves a performance advantage over the direct VQ systems. For example, for quantization with 36 bits per prototype, the proposed system achieves a segSNR

	Bits/prototype	female	male	Overall
Ref. Sys. 1: Split VDVQ	36(9bitx4cdbk)	6.06	4.22	5.21
Ref. Sys. 2: Oversampling+VDVQ	36(9bitx4cdbk)	6.31	4.57	5.50
Mag. MS-NSVQ+ Phase Cdbk	36 (6bitx4stg +12)	8.88	5.96	7.53
Reference 1	72 (9bitx8cdbk)	9.53	6.47	8.11
Reference 2	72 (9bitx8cdbk)	9.44	6.94	8.28
Spectral MS-NSVQ + Phase Cdbk	72 (6bitx10stg+12)	13.17	8.68	11.09

Table 4.1: Comparisons between The Phase Codebook Model and Direct Waveform VQ Schemes

	Bits/prototype	female	male	Overall
Ref. Sys. 1: Split VDVQ	44(11bitx4cdbk)	7.51	5.44	6.55
Ref. Sys. 2: Oversampling+VQ	44(11bitx4cdbk)	7.89	5.83	6.94
Mag. MS-NSVQ+ Phase Cdbk	36 (6bitx4stg +12)	8.88	5.96	7.53
Reference 1	80 (10bitx8cdbk)	10.93	7.45	9.32
Reference 2	80 (10bitx8cdbk)	10.57	7.83	9.30
Spectral MS-NSVQ + Phase Cdbk	72 (6bitx10stg+12)	13.17	8.68	11.09

Table 4.2: Comparisons Between The Phase Codebook Model and Direct Waveform VQs With Extra Bits

improvement of about 2 dB with respect to the second reference system ; even if an extra 8 bits are allocated to the second reference system to account for the alignment, the proposed system retains a performance gain of 0.6 dB.

Table 4.3 shows the comparison between the proposed system and the reference systems with alignment. In the direct waveform quantization schemes, the waveform is aligned based on the main peak of the pitch cycle before being quantized. The proposed system outperforms both of the direct waveform VQ with alignment. For example, the phase codebook model outperforms the the second system with alignment about 0.8 dB for a bit allocation of 36 bits/prototype and by 1.6 dB for 72 bits per prototype.

The effect of codebook size for phase and magnitude codebooks is shown in figure 4.12. By increasing the phase codebook size from 7 bits to 12 bits, the performance

	Bits/prototype	female	male	Overall
Ref. Sys. 1 with Alignment	36(9bitx4cdbk)	7.35	5.44	6.47
Ref. Sys. 2 with Alignment	36(9bitx4cdbk)	7.56	5.74	6.72
Mag. MS-NSTVQ+ Phase Cdbk	36 (6bitx4stg +12)	8.88	5.96	7.53
Ref. Sys. 1 with Alignment	72 (9bitx8cdbk)	10.83	7.46	9.26
Ref. Sys. 2 with Alignment	72 (9bitx8cdbk)	10.75	7.90	9.43
Mag. MS-NSTVQ + Phase Cdbk	72 (6bitx10stg+12)	13.17	8.68	11.09

Table 4.3: Comparisons Between The Phase Codebook Model and Direct Waveform VQ With Alignment

improves by as much as 3.5 dB. Within the allocation ranges considered in figure 4.12, assigning bits to phases appears to be more efficient that assigning the same amount of bits to magnitudes.

To test the subjective speech quality of the phase codebook model, informal listening tests have been conducted where original speech prototype waveforms were replaced by the quantized waveforms using either the phase codebook model or the reference systems. The speech quality of the systems are consistent with the objective signal-to-noise ratios.

From the performance comparisons between the proposed model and the reference systems, it is shown that the proposed phase codebook model reaches its objective —efficient quantization of the phase information of the prototype waveforms at low bit rates. The performance gain of the phase codebook model seems to be due to the simultaneous waveform alignment and quantization procedures. In other words, the closed-loop waveform alignment plays an important role in improving the quantization efficiency of the proposed model.

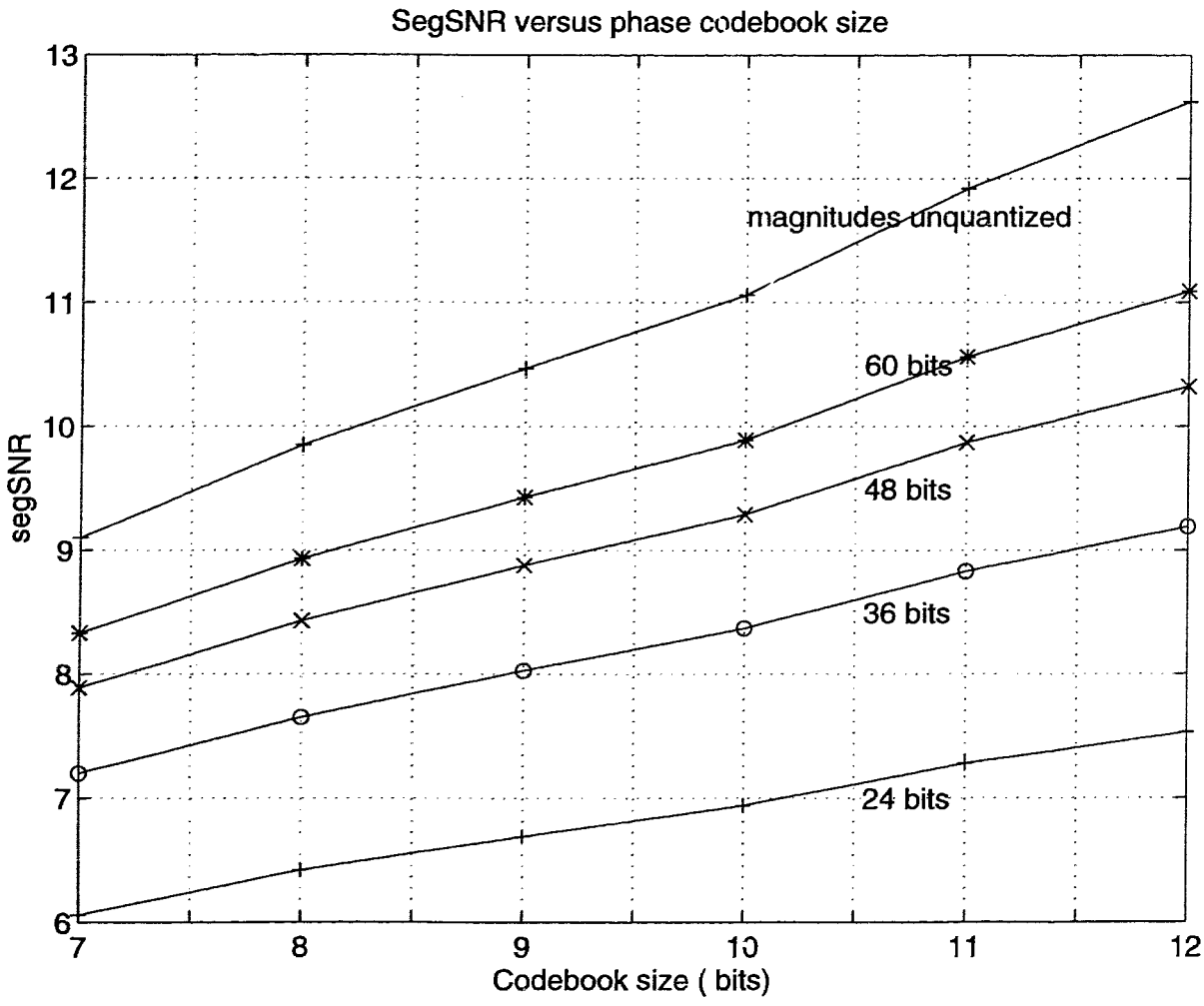

Figure 4.12: Effect of phase and magnitude codebook size on performance

4.6 Summary

As a main chapter of the thesis, this chapter focuses on the proposed phase codebook model. The first section discusses the motivation and objective— to find an efficient way of quantizing spectral phase information.

In section 4.2, the detailed derivation of the basic phase codebook model is

given, and its properties are discussed. It has been shown that waveform alignment of prototype waveforms can be performed simultaneously with the phase codebook search procedure (closed-loop waveform alignment). To reduce the computational complexity of the basic model, three different waveform oversampling schemes are discussed in section 4.3. The polyphase filtering scheme can perform interpolation on non-periodic signals while other two interpolation schemes based on interpolator $H_l(k)$ and zero-padding in the frequency domain have better computational efficiency. By using these oversampling techniques, a modified phase codebook model is obtained in section 4.4.

Finally, in section 4.5, the proposed phase codebook model is compared with two direct waveform quantization systems, and experimental results show that the phase codebook model outperforms both direct waveform quantization systems by a significant margin. In the comparisons where extra bits are assigned to direct waveform quantization to compensate for waveform alignment, and where alignment is added to direct waveform quantization, the phase codebook model still has a performance advantage. These experiments confirm that the phase model is valid, and it is possible to quantize phase information at low bit rates.

Chapter 5

Applying the Model to Interpolative Coding

Recently, interpolative coding has emerged as a new class of coders that combines the dominant CELP coder with the sinusoidal model. Interpolative coding has a good potential to reduce the coding bit rates. This chapter applies the phase codebook model developed in chapter 4 to interpolative speech coding. In chapter 3, we have introduced the basic concepts of prototype waveform representation, quantization, and interpolation. These concepts will be directly used in this chapter.

This chapter is organized as follows. In the first section, we develop a prototype waveform interpolation scheme based on the phase codebook model. The phase codebook model can be used in either single or multiple prototype waveform interpolation. In the second section, a speech coding system using the multiple prototype waveform interpolation concept and the phase codebook model is developed. Finally, subjective test results of the coder are reported.

5.1 Waveform Interpolation Based on the Phase Model

In section 3.1.4, we have discussed the general form of prototype waveform interpolation which is derived in the continuous time domain. In practice, the general form is difficult to implement in the discrete time domain because both the pitch and the shape of prototype waveforms change continuously with time while the sampling rate for the discrete time is fixed. However, if a prototype waveform is quantized by the phase codebook model, the length of the prototype waveform is normalized. Thus only the shape of the prototype waveform has to be interpolated, and this can be performed as shown below.

Recall that in section 3.1.4, at each time instant t_i along time axis t, there is an instantaneous prototype waveform $u(t_i, \phi)$. Together with time axis t, the abstract time axis ϕ is introduced to describe the instantaneous prototype waveform shape. All these definitions are in the continuous time domain. In the discrete time domain, let the time axis m correspond to the continuous time axis t, and assume that both instantaneous prototype waveforms are represented by the phase codebook model:

$$\begin{aligned} &v(0,n), \quad 0 \le n < N \quad \text{at time instant } m = 0. \\ &v(M,n). \quad 0 \le n < N \quad \text{at time instant } m = M. \end{aligned} \tag{5.1}$$

where variable n corresponds to the abstract time axis φ. Note that N is the fixed length of the phase codebook model (the size of FFT). The phase codebook model interpolates prototype waveforms to a fixed length N.

Figure 5.1 shows the two instantaneous prototype waveforms, $v(0,n)$ and $v(M,n)$, as two-dimensional signals along time axes m and n. The instantaneous waveforms have distance M on the time axis m. If linear interpolation is used, then the wave-

form shape at each time instant m, $0 \leq m < M$, can be interpolated as:

$$v(m,n) = \frac{(M-m)}{M} v(0,n) + \frac{m}{M} v(M,n) \quad 0 \leq n < N \tag{5.2}$$

In order to recover an interpolated signal $e(m)$, $0 \leq m < M$, along time axis m, first we have to un-normalize the waveform shape $v(m,n)$, where $0 \leq n < N$, from length N to the instant pitch size at each time instant m. Then the ideal interpolated signal $e(m)$, $0 \leq m < M$, can be obtained by concatenating infinitesimal segments of the un-normalized instantaneous waveform shapes. To maintain the smoothness of the ideal interpolated signal $e(m)$, the phase continuity between the boundaries of these infinitesimal intervals has to be preserved.

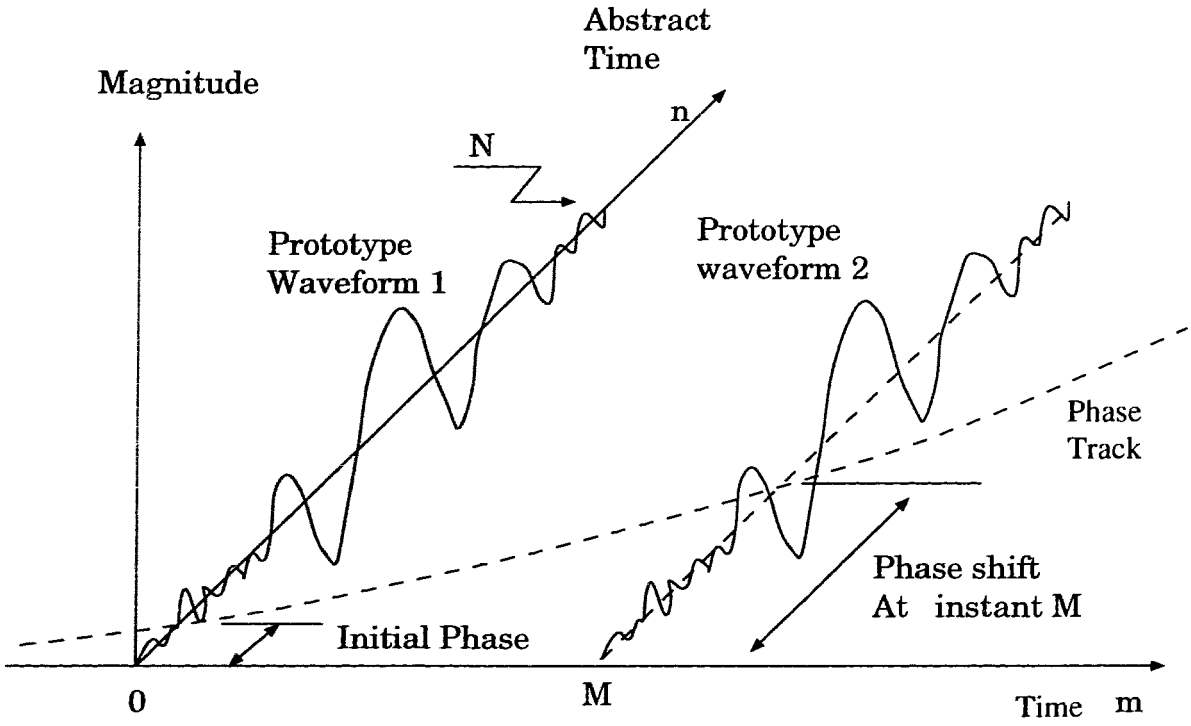

Figure 5.1: Interpolation Between Two Prototype Waveforms

Given the pitch values at time instants $m = 0$ and $m = M$, $P(0)$ and $P(M)$, the harmonic frequencies are $\omega(0) = 2\pi/P(0)$ and $\omega(M) = 2\pi/P(M)$ respectively.

When linear interpolation is used, the fundamental frequency at time instant m is

$$\omega(m) = (\omega(M) - \omega(0))\frac{m}{M} + \omega(0) \quad 0 \leq m < M \tag{5.3}$$

Taking the integral along time axis m, the phase track at time instant m is

$$\theta(m) = \theta(0) + \omega(0)m + (\omega(M) - \omega(0))\frac{m^2}{2M} \quad 0 \leq m < M \tag{5.4}$$

where $\theta(0)$ is the initial phase offset at time instant $m = 0$. The above phase increases monotonically, and the phase can be wrapped to its principle value :

$$\Theta(m) = ((\theta(m)))_{(2\pi)} \tag{5.5}$$

The un-normalization of each interpolated instantaneous waveform shape from size N to its corresponding pitch size $P(m) = 2\pi/\omega(m)$ can be performed by finding the desired phase track in terms of the number of samples along axis n

$$\begin{aligned}
\Phi(m) &= \frac{N}{2\pi}\Theta(m) \\
&= \left(\left(\frac{N}{2\pi}\{\theta(0) + \omega(0)m + (\omega(M) - \omega(0))\frac{m^2}{2M}\}\right)\right)_N \\
&= \left(\left(\Phi(0) + \frac{N}{2\pi}\{\omega(0)m + (\omega(M) - \omega(0))\frac{m^2}{2M}\}\right)\right)_N \quad 0 \leq m < M
\end{aligned} \tag{5.6}$$

where $\Phi(0) = \frac{N}{2\pi}\theta(0)$. $\Phi(m)$ is not usually an integer. Since the interpolated waveform shapes $v(m,n)$ are in the discrete form, only the values at integer samples are known. However, when $N \gg P(m)$, the waveform shape values of non-integer sample locations can be approximated by its closest integer sample values. Based on this approximation, the desired interpolated waveform $e(m)$ is

$$e(m) = v(m, round(\Phi(m))) \quad 0 \leq m < M \tag{5.7}$$

where *round* means taking the closest integer value. When N is not large enough, the oversampling techniques described in section 4.3 can be used to increase the value of N.

With respect to computational complexity, the phase track $\Phi(m)$ can be calculated before evaluating eqn. (5.2). Therefore, only a specific value $n = \Phi(m)$ needs to be evaluated for each time instant m. Combining eqns. (5.7) and (5.2), we have

$$\begin{aligned} e(m) &= v(m,n)|_{n=round(\Phi(m))} \\ &= \frac{(M-m)}{M} v(0, round(\Phi(m))) + \frac{m}{M} v(M, round(\Phi(m))) \end{aligned} \tag{5.8}$$

The gain of the interpolated signal $e(m)$ can be interpolated on a sample-by-sample basis. Suppose $g(0)$ and $g(M)$ are the gains of the instantaneous prototype waveforms at time instants $m = 0$ and $m = M$. If linear interpolation in the logarithm domain is applied, the interpolated gain of prototype waveform $v(m,n)$ at time instant m, $0 \leq m < M$, is:

$$\hat{g}(m) = \frac{M-m}{M} g(0) + \frac{m}{M} g(M) \tag{5.9}$$

From the above equation, a gain factor is obtained for each prototype waveform $v(m,n)$. By applying the gain factor, eqn. (5.8) can be rewritten as:

$$\begin{aligned} e(m) &= \hat{g}(m) v(m,n)|_{n=round(\Phi(m))} \\ &= \frac{(M-m)}{M} \hat{g}(m) v(0, round(\Phi(m))) + \frac{m}{M} \hat{g}(m) v(M, round(\Phi(m))) \end{aligned} \tag{5.10}$$

5.2 A Multiple PWI Speech Coder

In this section, a speech coder is presented based on the concept of multiple prototype waveform interpolation and the phase codebook model.

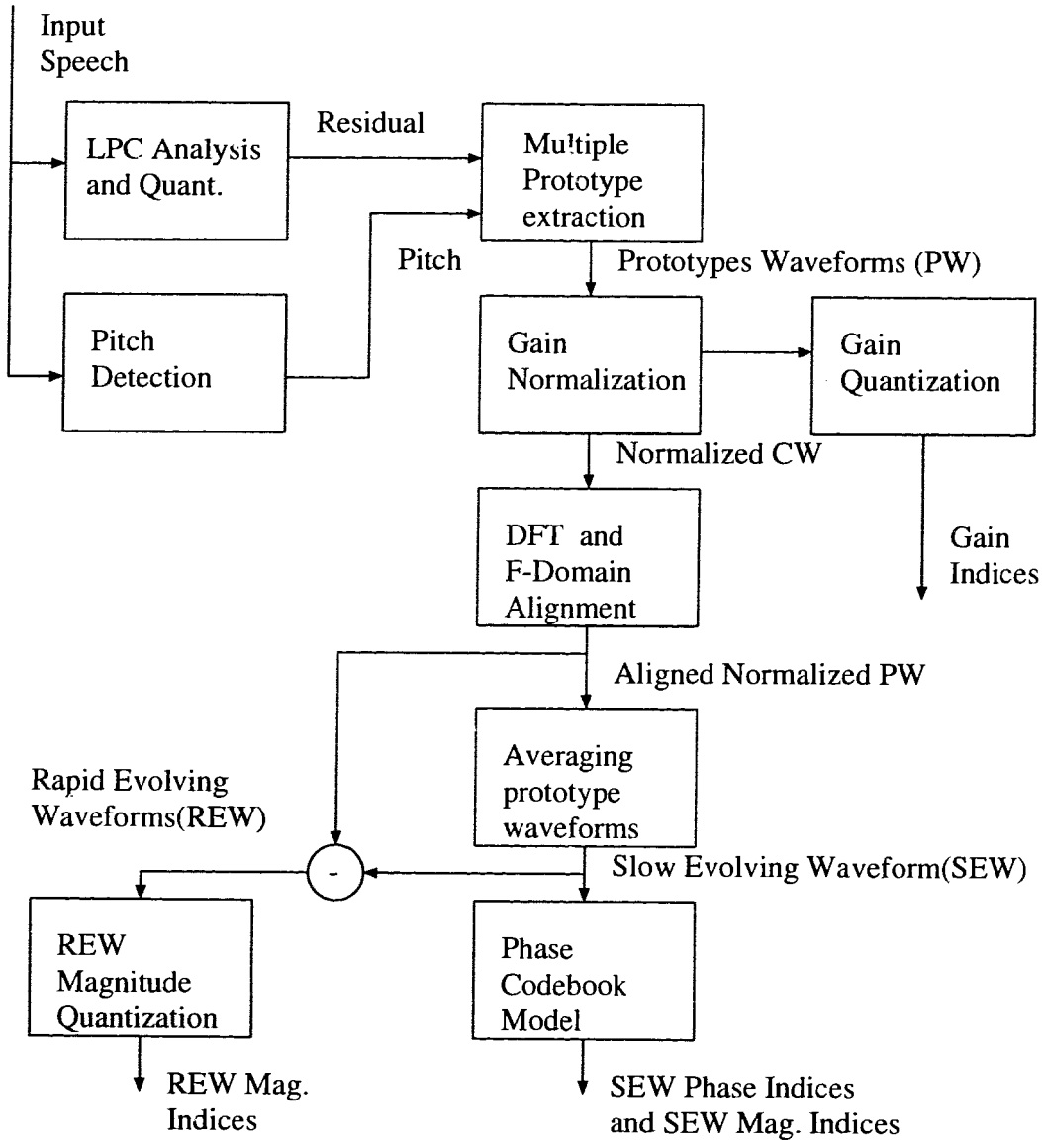

Figure 5.2: Encoder of the Interpolative Coding System

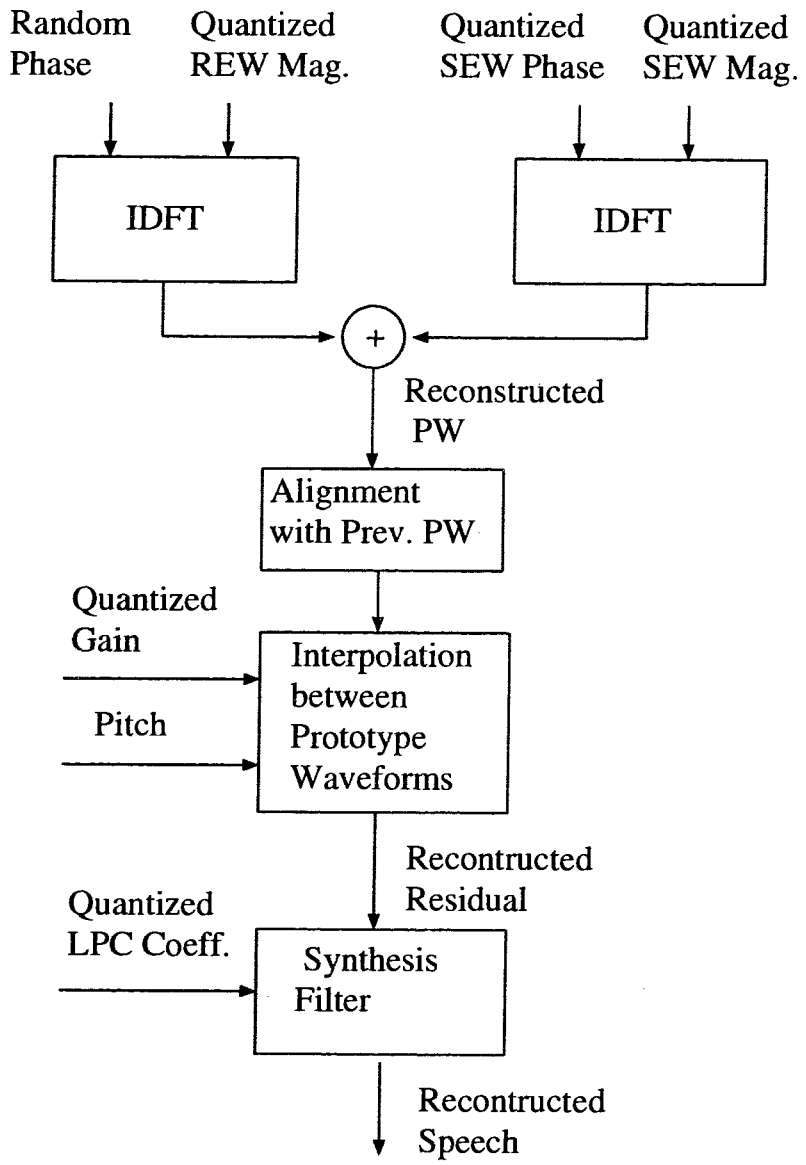

Figure 5.3: Decoder of the Interpolative Coding System

5.2.1 System Overview

This subsection describes briefly the proposed speech coding system. Details of the different parts of the system are discussed in the following subsections.

Figure 5.2 shows the block diagram of the encoder. At the encoder, the input speech is divided into frames with frame size of 25 ms. LPC analysis is performed on each frame, and the residual signal is obtained by passing the input speech through the LPC filter. Pitch detection is performed on the speech signal to obtain pitch values. Each frame of the residual signal is further divided into 10 subframes. For each subframe, a prototype waveform is extracted from the residual signal. Usually the subframe size is smaller than those of prototype waveforms, and a prototype waveform can expand over a few subframes. Therefore, it is common that prototype waveforms overlap with each other (See figure 3.3).

The extracted prototype waveforms are gain-normalized (See section 5.2.4). Then pitch-size DFT is applied to each prototype waveform, and the prototype waveform is oversampled to a fixed length of N by padding zeros in the frequency domain (See section 4.3). Each prototype waveform is aligned with previous ones in the frequency domain (See section 5.2.5).

The DFT coefficients of the resulting normalized and aligned prototype waveforms in each frame are averaged to obtain the *slowly evolving waveform* (SEW) for each frame. For each subframe, one *rapid evolving waveform* (REW) is obtained by subtracting the current frame's SEW from the DFT coefficients of the current subframe's prototype waveform. Both of the magnitudes and phases of the SEW are quantized using the phase codebook model, whereas only the magnitudes of the REW are quantized using NSTVQ [36].

The purpose of decomposing prototype waveforms into SEWs and REWs is to exploit the human auditory system. It is noted that unvoiced speech can be encoded in a perceptually accurate manner at a very low bit rate [33]. This is the case despite the fact that unvoiced speech has a very high information rate from an information-theoretic viewpoint. On the contrary, the bit rate required for voiced speech is usually relatively high, despite the slow evolution of the pitch cycle waveform. These observations show that the human ear is more sensitive to slowly evolving feature of the waveform. The decomposition of prototype waveforms into SEWs and REWs facilitates the use of different quantization schemes for voiced-like SEW and noise-like REW [31].

At the decoder shown in figure 5.3, a prototype waveform is reconstructed by combining a SEW and REW in the time domain. The SEW is obtained by performing IFFT on the quantized SEW magnitudes and phases. The REW is obtained by performing IFFT on the quantized REW using random phases. The random phase is used to reduce the bit rate of the system assuming that the REW has a noise-like characteristic. The SEW and REW are added up in the time domain to obtain reconstructed prototype waveforms.

The prototype waveform interpolation scheme for the phase codebook model (See section 5.1) is employed to recover the excitation (residual) signal using the reconstructed prototype waveforms. Each prototype waveform has to be aligned with its previous prototype waveform before interpolation. The first prototype waveform in a frame is aligned with the last prototype waveform of the previous frame. Finally, the reconstructed speech is obtained by passing the excitation signal through the synthesis filter.

Starting from the next subsection, the details of the coding system are presented.

5.2.2 LP Analysis

Once in each speech frame, the speech spectral envelope is estimated using 10th order LPC analysis. In the system, frame size is set to be 25 ms (200 samples). Analysis is performed on speech segments obtained by multiplying the input speech signal with Hamming window. The coefficients are converted to *line spectral pairs* (LSP) and quantized once per frame using the *tree-searched multi-stage vector quantization* (MSVQ) scheme presented in [6]. The 10 LSP coefficients of each frame are encoded using a 8-stage, 3 bits/stage, MSVQ (24bits). The quantized coefficients are then transformed back into LPCs and used in the short-term filter (See eqn. (2.8)) which computes the excitation signal according to eqn. (2.2). The filter coefficients are updated using LSP interpolation and the update interval is 5 ms.

5.2.3 Prototype Waveform Extraction

For each frame, a pitch value is determined using a modified version of the pitch algorithm developed by Bhattacharya [5]. This algorithm combines a peak picking procedure with forward and backward correlation checking. In the algorithm, the peak picking procedure first selects a set of candidate pitch pulses based on the pitch value of the previous frame and on the pitch track length. Second, from the candidate pitch pulses, a set of candidate pitch values is obtained. If the candidate pitch values are similar to the pitch values of the previous frame, and the deviation of these candidate values is below a given threshold, The pitch value of the current frame is set to be the average of these candidate values. If the candidate pitch values has large deviation, forward and/or backward correlation is employed to check if any pitch pulse is missed by the pitch marker or the current frame is an unvoiced frame. For voiced speech, the algorithm perform pitch tracking to make sure that the pitch

value changes smoothly with time. For unvoiced speech, large pitch values (greater than 100 samples in 8 kHz sampling rate) are selected to avoid unwanted periodicity in reconstructed unvoiced speech.

In the proposed system, each frame is divided into 10 subframes, with subframe size 2.5 ms. A prototype waveform is extracted for each subframe. The starting point of the prototype waveform is restricted to an interval of 1.25 ms. The exact starting point is decided on the basis of minimizing the square error at the boundaries of prototype waveforms. Similar to [9], this interval restriction will maximize the tracking of the speech dynamic and avoid double occurrences of transitory events in the input speech appearing in the synthesized output.

5.2.4 Prototype Waveform Gain-Normalization and Oversampling

Suppose we have L subframes in a frame, and each subframe is of length M. Starting from each subframe, we extract one prototype waveform $r(lM, n)$, where $0 \leq l < L$, and $0 \leq n < P_l$. P_l is the length of the prototype waveform (lM, n), i. e. the pitch size in the current subframe. P_l can be simply obtained by interpolation between quantized pitch values between frames.

In the encoder, the prototype waveform $r(lM, n)$ is first gain-normalized. the gain of each prototype waveform is defined as

$$g_l = \left(\frac{1}{P_l} \sum_{n=0}^{P_l - 1} |r(lM, n)|^2 \right)^{\frac{1}{2}} \quad 0 \leq l < L \tag{5.11}$$

Thus the normalized prototype waveform in the time domain is

$$r_n(lM, n) = \frac{r(lM, n)}{g_l} \quad 0 \leq l < L \quad 0 \leq n < P_l \tag{5.12}$$

Taking pitch size DFT of the $r_n(lM,n)$, we obtain the frequency domain representation of the normalized prototype waveform

$$R_n(lM,k) = \sum_{n=0}^{P_l-1} r_n(lM,n)e^{\frac{-j2\pi nk}{P_l}} \tag{5.13}$$

Applying the *zero-padding* oversampling technique (See section 4.3), prototype waveform $R_n(lM,n)$ of size P_l can be oversampled to a prototype waveform of a fixed length N. Denoted by $V_n(lM,n)$, the oversampled prototype waveform in the frequency domain is

$$V_n(lM,k) = \begin{cases} N/P_l\ R_n(lM,k) & 0 \le k \le \lfloor (P_l-1)/2 \rfloor \\ 0 & \lfloor (P_l-1)/2 \rfloor < k < \lceil (P_l+1)/2 \rceil + N - P_l \\ N/P_l\ R_n(lM,k-N+P_l) & \lceil (P_l+1)/2 \rceil + N - P_l \le k < N \end{cases} \tag{5.14}$$

for $0 \le l < L$. The oversampled prototype waveform in the time domain is

$$v_n(lM,n) = \frac{1}{N}\sum_{k=0}^{N-1} V_n(lM,k)e^{\frac{j2\pi nk}{N}} \tag{5.15}$$

for $0 \le l < L$.

To quantize the gain factor g_l, g_l is first transformed to the logarithm domain. Each frame has L gains, and can be vector quantized using an ordinary VQ. L is 10 in our proposed system. To reduce the vector dimension, in the coding system to be evaluated in section 5.3, only 2 gains are extracted by grouping gains g_l, $0 \le l < L$, into 2 groups and finding the average of the gains in each group. The 2 gains are then quantized using a two-dimensional 10-bit VQ.

5.2.5 Computation of SEW and REW

It has been noted that the human ear is more sensitive to details of the speech waveform when it evolves slowly than when it evolves rapidly. In other words, the human ear is more sensitive to slowly evolving feature of the waveform [33].

In order to achieve high quantization efficiency, the prototype waveforms are decomposed into SEW and REW. In [32], SEW are obtained by linear-phase low-pass filtering the Fourier coefficients along time axis t (section 3.2.5). In [9], the low-pass filtering operation is simplified as averaging Fourier coefficients along time axis t. In the proposed coding system, prototype waveforms are decomposed into SEW and REW using the averaging method. In each frame, a SEW is extracted by averaging of all oversampled prototype waveforms in the frame.

Before averaging prototype waveforms, alignment is required. Each prototype waveform is always aligned with the previous one. The first prototype waveform is aligned with the last prototype waveform of the previous frame. Suppose the last prototype waveform of the previous frame is $v_{na}(-M, n)$, $0 \leq n < N$. The recursive alignment procedure can be described as

$$\begin{aligned} n_l &= \arg\max_i \left(\sum_{n=0}^{N-1} v_{na}((l-1)M, n) v_n(lM, ((n-i))_N) \right) \\ &= \arg\max_i \left(\frac{1}{N} \sum_{k=0}^{N-1} V_{na}((l-1)M, k) V_n^*(lM, k) e^{\frac{j2\pi ik}{N}} \right) \end{aligned} \tag{5.16}$$

$$v_{na}(lM, n) = v_n(lM, ((n-n_l))_N) \quad 0 \leq n < N \tag{5.17}$$

$$V_{na}(lM, k) = V_n(lM, k) e^{\frac{-j2\pi n_l k}{N}} \quad 0 \leq k < N \tag{5.18}$$

for $0 \leq l < L$. Figure 5.4 shows typical normalized and aligned prototype waveforms $v_{na}(lM, n)$ extracted in a frame. In the figure, each frame is divided into 10 subframes, and the FFT size, N, is 512.

The SEW of the current frame is defined as

$$\mathcal{S}(k) = \frac{1}{L} \sum_{l=0}^{L-1} V_{na}(lM, k) \quad 0 \leq k < N \tag{5.19}$$

The l^{th} REW of the current frame is

$$\mathcal{R}_l(k) = V_{na}(lM, k) - \mathcal{S}(k) \quad 0 \leq k < N \tag{5.20}$$

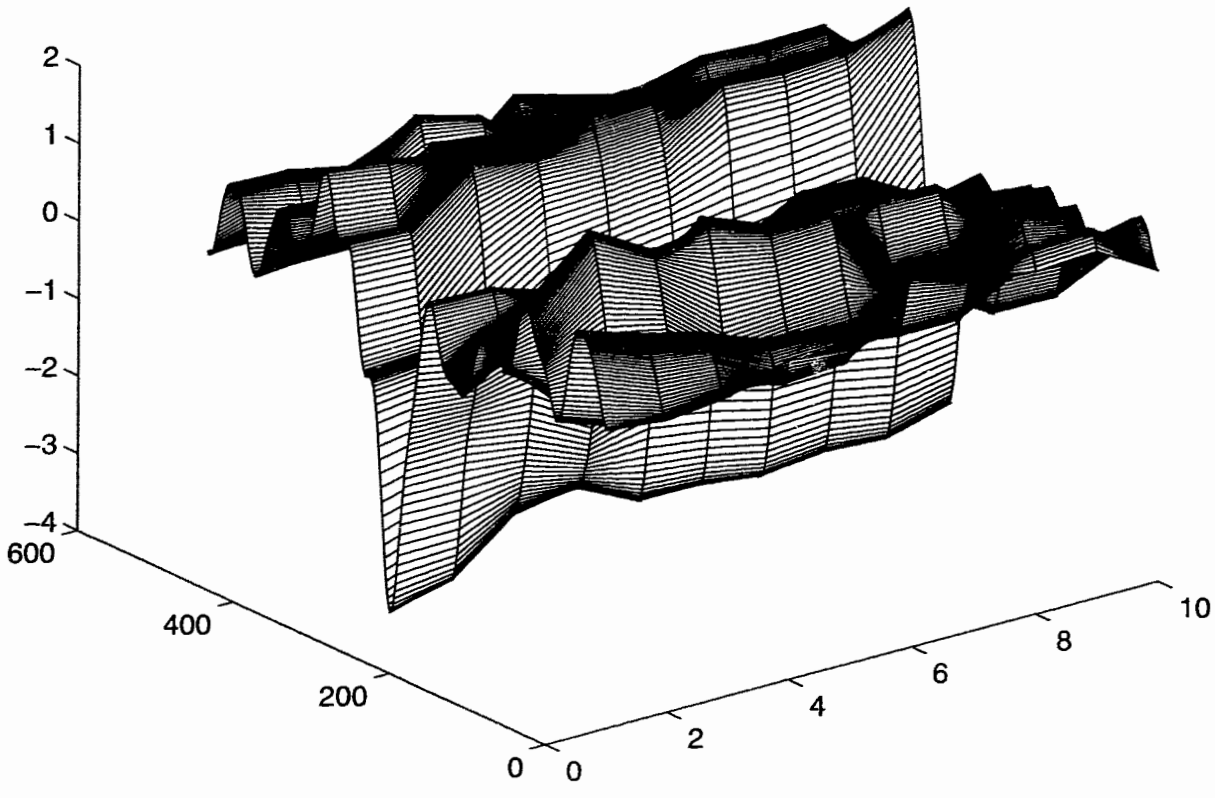

Figure 5.4: Normalized and Aligned Prototype Waveform In a Frame

5.2.6 Quantization SEW and REW

The phase codebook model (See section 4.4) is used to quantize the complex SEW. The SEW is decomposed into spectral magnitudes and spectral phases. The magnitudes are quantized using NSTVQ [36] while the spectral phases are quantized using the phase codebook model. In the coding system to be evaluated in section 5.3, a 7-bit magnitude codebook and a 6-bit phase codebook are used.

The REW has a noise-like characteristic. To reduce the bit rate of the coder, only the magnitudes of REW are coarsely quantized. The low frequency component of the REW is suppressed to reduce the interference with the low frequency, periodic SEW structure of the prototype waveform. In the proposed coding system, the low frequency component of the REW under 1 khz is suppressed, and only the REW magnitude shapes of the 2nd, 5th, and 9th subframes in a frame are quantized using a 3-bit NSTVQ while each waveform shapes of the missing 7 subframes of total 10 is set to be equal to the quantized waveform shape of either previous or next subframe. 1 bit is required for each of these subframes to decide whether previous or next quantized shape is used based on the minimum magnitude distortion criterion. This REW quantization scheme is similar to the one in [32].

5.2.7 Reconstruction of the Excitation Signal

As shown in figure 5.3, the decoder reconstructs the prototype waveforms from the SEW and REW. A reconstructed prototype waveform is the sum of the current reconstructed SEW and REW. To reconstruct the SEW, the magnitudes and phases are recovered from the SEW magnitudes and phase codebooks, and inverse FFT is performed to reconstruct the SEW in the time domain. Similarly, inverse FFT is performed on the combination of the reconstructed REW magnitudes (from the

REW magnitudes codebook) and random phases to obtain the time-domain REW. Then the SEW and REW are added up in the time domain to form the reconstructed prototype waveform $\hat{v}(lM, n)$, where $0 \leq l < L$.

To obtain the one-dimensional reconstructed excitation signal of the current frame $\hat{e}(m)$, where $0 \leq m < ML$, eqn. (5.8) is used in the decoder on a subframe-by-subframe basis. The phase track required in eqn. (5.8) can be obtained by linearly interpolating the pitch of the current frame and that of the previous frame and by using eqn. (5.6).

The gain factor is added to the excitation signal on a sample-by-sample basis as discussed in section 5.1. The gain for each sample is recovered by linear interpolation in the logarithm domain between quantized gains.

When the SEW and REW are not quantized, a small set of speech sentences without background noise has been tested, and the experimental results show that the system generates synthesized speech that has almost no difference to the original speech signal.

5.2.8 Bit Allocation

The detailed bit allocation for the proposed system is given in table 5.1. The frame size is 25 ms, and subframe size is 2.5 ms. As we can see on the table, 24 bits/frame are used to quantize the short-term prediction coefficients, while 7 bits/frame is used for the pitch value. There are 10 prototype waveforms in each frame. From these 10 prototype waveform, one SEW is generated, and quantized using a 7-bit magnitude codebook and a 6-bit phase codebook. For REW magnitude quantization, only the magnitude vectors of 3 subframes (2nd, 5th and 9th) are quantized using a 3-bit

PARAMETER	Bits/Frame	Rates (bps)
LSP Coefficients	24	960
Pitch Period	7	280
Gain	10	400
SEW Mag.	7	280
SEW Phase	6	240
REW Mag.	3x3+7x1	640
Total	70	2800

Table 5.1: Bit Allocations for the 2.8 kbps PWI Codec Using a Frame length of 200 Samples with 10 Prototype Waveforms per Frame

codebook while those of the remaining subframes are set to be equal to the vector of their corresponding left or right subframes. The left or right decision is based on the spectral error, and 1 bit is required for each subframe.

5.3 Performance Evaluation

The performance of the PWI system was evaluated using an informal Mean Opinion Score (MOS) test in which several speech codecs were used to encode 10 sentences, 5 from male speakers and 5 from female speakers. For each sentence, an uncompressed (16-bit PCM) version was played followed by the output of each codec being tested. The order of the codecs was randomized between sentences, and stereo headphones were used in which the same signal was sent to both channels.

Twenty participants took part in the informal MOS test and were asked to rate the quality of each speech sample using a scale from 1 to 5 representing a subjective quality of bad, poor, fair, good, and excellent. This provides a total of 200 ratings for each system. The uncompressed samples were always played first as a reference and were given a score of 5 in advance. Included in the test was the existing 2.4 kbps LPC-10e standard[55], the 4.15 kbps IMBE standard[25] (See section 2.5.3),

		Mean Opinion Score			Variance		
System	Rate (bps)	All	Male	Female	All	Male	Female
IMBE	4150	3.30	3.13	3.48	0.50	0.48	0.47
DoD CELP	4600	3.29	3.21	3.37	0.54	0.60	0.47
PWI	2800	2.77	2.68	2.87	0.47	0.44	0.48
LPC-10e	2400	1.98	1.88	2.09	0.51	0.48	0.51

Table 5.2: Mean opinion scores (MOS) results

and the FS 1016 4.6 kbps CELP standard[29]. All the standard coders are official releases from their corresponding organizations/parties.

The results of the test are shown in table 5.2. The 2.8 kbps PWI system scored about 0.8 MOS points higher than that of the existing LPC-10e standard at 2.4 kbps. However, it scored about 0.5 points less than the FS 1016 CELP system operating at 4.6 kbps, and the 4.15 kbps IMBE standard.

For a small tested set of speech sentences without background noise, experiments show that the speech quality of the system with no SEW and REW quantization is almost equivalent to the original speech. The SEW-only excitation generates mechanical quality speech because of the excessive periodicity. As we add on more and more REW components, the mechanical sounds disappear, however, with increasing noise similar to background noise. Compared with the IMBE standard, the main deficiency of the current system seems to be its noise. To improve the speech quality of the coder, efficient representation of the REW components is required.

Chapter 6

Conclusions

Recent speech coders, using either the sinusoidal model or interpolative coding, bring new challenges to the field of speech coding. One of the challenges common to these coders is the quantization of the phase information of sinusoids at low bit rates. The objective of this thesis is to find an efficient way of quantizing spectral phase information. We have developed a phase codebook model to quantize the phase information of speech prototype waveforms at low bit rates.

In the proposed model, the minimum mean square error (MSE) criterion is applied to prototype waveforms in the time domain. The spectral phase of prototype waveforms is separated completely from the spectral magnitude, and quantized using a phase codebook. The model performs closed-loop waveform alignment together with the phase codebook search procedure. To reduce the computational complexity of the basic model, oversampling techniques are used to obtain a modified phase codebook model.

The proposed phase codebook model is compared with two direct waveform quantization systems, and experimental results shows that the phase codebook model

outperforms both direct waveform quantization systems by a significant margin. In the comparisons where extra bits are assigned to direct waveform quantization to compensate for waveform alignment, and where alignment is added to direct waveform quantization, the phase codebook model still has a performance advantage. These experiments confirm that the phase model is valid, and the phase codebook model provides a way of quantizing phase information at low bit rates.

The phase codebook model is applied to prototype waveform interpolation coding. An efficient interpolation scheme of prototype waveforms based on the phase codebook model is developed. A speech coder based on interpolative coding is developed, and good results are obtained.

6.1 Suggestions for Future Work

This section provides suggestions for further research into several areas covered in this thesis.

- Regarding the phase model, it will be interesting to see if the idea is applicable to the quantization of phase information in sinusoidal coding instead of the phase dispersion factor, which is common in current sinusoidal coders.
- Perceptually efficient REW representation. The REW quantization in the current coder requires high bit rates. To improve the coder's quality and to further reduce the bit rate of the coder, perceptually efficient quantization schemes need to be developed.
- Regarding the representation of prototype waveforms, it is unclear if the SEW and REW decomposition is optimal. It is possible that other decompositions provide more efficient quantization.

Appendix A

Derivation of Interpolator functions $H_l(k)$

This appendix derives in details why interpolators $H_l(k)$, as defined by eqn. (4.11), performs ideal $1-to-L$ interpolation on $x(n)$ as defined in eqn. 4.9.

Suppose we have already obtained the ideal oversampled version of $x(n)$ — $y(m)$— using $H_l(k)$. By definition the DFT of $y(m)$ is

$$Y(k) = \sum_{m=0}^{LP-1} y(m) e^{\frac{-j2\pi mk}{LP}} \tag{A.1}$$

Substituting m by eqn. (4.15), eqn.(A.1) can be written as

$$\begin{aligned} Y(k) &= \sum_{l=0}^{L-1}\sum_{n=0}^{P-1} y(l+nL) e^{\frac{-j2\pi(l+nL)k}{LP}} \\ &= \sum_{l=0}^{L-1}\sum_{n=0}^{P-1} x_l(n) e^{\frac{-j2\pi nk}{P}} e^{\frac{-j2\pi lk}{LP}} \\ &= \sum_{l=0}^{L-1} X_l(((k))_P) e^{\frac{-j2\pi lk}{LP}} \end{aligned} \tag{A.2}$$

Combining eqn. (A.2) and eqn. (4.12),

$$\begin{aligned} Y(k) &= \sum_{l=0}^{L-1} X(((k))_P) H_l(((k))_P) e^{\frac{-j2\pi lk}{LP}} \\ &= X(((k))_P) \sum_{l=0}^{L-1} H_l(((k))_P) e^{\frac{-j2\pi lk}{LP}} \end{aligned} \tag{A.3}$$

Using eqn. (4.11) , the summation term in eqn. (A.3) is

$$\sum_{l=0}^{L-1} H_l(((k))_P) e^{\frac{-j2\pi lk}{LP}} = \begin{cases} \sum_{l=0}^{L-1} e^{\frac{j2\pi((k))_P}{P}} \frac{l}{L} e^{\frac{-j2\pi lk}{LP}} & 0 \le ((k))_P \le \lfloor (P-1)/2 \rfloor \\ 0 & P/2 \text{ if } P \text{ is even} \\ \sum_{l=0}^{L-1} e^{\frac{-j2\pi((P-k))_P}{P}} \frac{l}{L} e^{\frac{-j2\pi lk}{LP}} & \lceil (P+1)/2 \rceil \le ((k))_P < P \end{cases} \tag{A.4}$$

Note that $0 < k < LP - 1$. Now define a new variable k' such that

$$k = l'P + k',\ 0 \le k' < P,\ 0 \le l' < L \tag{A.5}$$

then the first branch at the right side of eqn. (A.4) is

$$\begin{aligned} \textit{First branch} &= \sum_{l=0}^{L-1} e^{\frac{j2\pi k'}{P} \frac{l}{L} e^{\frac{-j2\pi(l'P+k')}{LP}}} \\ &= \sum_{l=0}^{L-1} e^{\frac{-j2\pi ll'}{L}} \\ &= \begin{cases} L & l' = 0 \\ 0 & otherwise \end{cases} \end{aligned} \tag{A.6}$$

The third branch is

$$\begin{aligned} \textit{Third branch} &= \sum_{l=0}^{L-1} e^{\frac{-j2\pi(P-k')}{P} \frac{l}{L} e^{\frac{-j2\pi(l'P+k')}{LP}}} \\ &= \sum_{l=0}^{L-1} e^{\frac{-j2\pi l(l'+1)}{L}} \\ &= \begin{cases} L & l' = L-1 \\ 0 & otherwise \end{cases} \end{aligned} \tag{A.7}$$

From eqn. (A.3), eqn. (A.6) and eqn. (A.7), $Y(k)$ can be written as

$$Y(k) = \begin{cases} L\,X(k) & 0 \le k \le \lfloor (P-1)/2 \rfloor \\ L\,X(((k))_p) & P(L-1) + \lceil (P+1)/2 \rceil \le k < PL \\ 0 & otherwise \end{cases} \tag{A.8}$$

From the above equation, $Y(k)$ equals to $X(k)$ at low frequency part with an scaling factor, and $Y(k)$ equals to zeros at high frequency part, From the Nyquist sampling theorem, it seems that $y(n)$ is a oversampled version of $x(n)$. However, if we look into in more detail the spectrum $Y(k)$, there is a constraint for $y(m)$ to be an ideal oversampled version of $x(n)$— When P is an even number, $X(P/2)$ have to equal to zero.

Because $Y(p/2)$ is always zero using the definition of $H_l(k)$ (the second branch in eqn. (A.4)),

$$Y(p/2) = \sum_{l=0}^{L-1} H_l(p/2)X(p/2) = 0 \tag{A.9}$$

Therefore, the spectrum of $Y(p/2)$ is not exactly $X(p/2)$ with a scaling factor at low frequency part. And $y(m)$ will not be an ideal interpolation of $x(n)$ if p is even and $X(P/2) \neq 0$.

However, $X(P/2)$ corresponds to the half the Nyquist sampling rate of $x(n)$, and it's reasonable to assume $X(P/2) = 0$ if $x(n)$ a speech or residual signal.

References

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

[2] L. B. Almeida and F. M. Silva. Variable-frequency synthesis: An improved harmonic coding scheme. In *IEEE Int'l Conf. on Acoustic, Speech and Signal Processing*, 1984.

[3] L. B. Almeida and J. M. Tribolet. Harmonic coding: A low bit-rate good-quality speech coding technique. In *IEEE Int'l Conf. on Acoustics, Speech and Signal Processing*, pages 1664–1667, 1982.

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

[5] B. Bhattacharya. A pitch determination algorithm. Unpublished, 1994.

[6] B. Bhattacharya, W. LeBlanc, S. Mahmoud, and V. Cuperman. Tree searched multi-stage vector quantization of LPC parameters for 4 kb/s speech coding. In *IEEE Int'l Conf. on Acoustic, Speech and Signal Processing*, pages 105–108, 1992.

[7] M. S. Brandstein. A 1.5 kbps multi-band excitation speech coder. Master's thesis, EECS Dept., MIT, September 1990.

[8] I. S. Burnett and G. J. Bradley. New techniques for multi-prototype waveform coding at 2.84kb/s. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, pages 261–264, Detroit, 1995.

[9] J. S. Burnett and G. J. Bradley. Low complexity decomposition and coding of prototype waveforms. In *IEEE Speech Coding Workshop*, pages 23–24, Annapolis, MD, Sept. 1995.

[10] C. S. Burrus. Efficient Fourier transform and convolution algorithms. In J. S. Lim and A. V. Oppenheim, editors, *Advanced Topics in Signal Processing*, chapter 4, pages 199–245. Prentice Hall, Englewood Cliffs, NJ, 1988.

[11] F. Charpentier and M . G. Stella. A diophone synthesis system using an overlap-add technique for speech waveforms concatenation. In *IEEE Int'l Conf. on Acoustic, Speech and Signal Processing*, pages 2015–2018, 1986.

[12] J. H. Chen. A robust low-delay speech coder at 16 kb/s. In *IEEE Int'l Conf. on Acoustic, Speech and Signal Processing*, 1989.

[13] J. H. Chen, N. Jayant, and R. V. Cox. Improving the performance of the 16 kb/s ld-celp speech coder. In *IEEE Int'l Conf. on Acoustic, Speech and Signal Processing*, 1992.

[14] R. E. Crochiere and L. R. Rabiner. *Multirate Digital Signal Processing.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983.

[15] R. E. Crochiere and L. R. Rabiner. *Multirate Processing of Digital Signal*, chapter 3, pages 123–198. Prentice-Hall, Englewood Cliffs, NJ, 1988.

[16] G. Dadidson and A. Gersho. Complexity reduction methods for vector excitation coding. In *IEEE Int'l Conf. on Acoustic, Speech and Signal Processing*, April 1986.

[17] A. Das, A. V. Rao, and A. Gersho. Variable-dimension vector quantization of speech spectra for low-rate vocoders. In *Proc. Data Compression Conference*, pages 421–429, 1994.

[18] W. A. Gardner and B. D. Rao. Mixed-phase ar models for voiced speech and perceptual cost function. In *IEEE Int'l Conf. on Acoustic, Speech and Signal Processing*, pages I205–I208, 1994.

[19] W. R. Gardner and B. D. Rao. Non-causal linear prediction of voiced speech. In *IEEE Asilomar Conf. on Signals, Systems, and Computers*, 1992.

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

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

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

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

[24] P. Hedelin. A tone-oriented voice-excited vocoder. In *IEEE Intl. Conf. Acoustic, Speech, and Signal Processing*, 1981.

[25] INMARSAT. Inmarsat-M voice coding system description (draft version). Technical report, INMARSAT, February 1991.

[26] Texas Instruments. *TMS320C30 User's Guide.* Texas Instruments Incorporated.

[27] Texas Instruments. *TMS320C50 User's Guide.* Texas Instruments Incorporated.

[28] Y. Jiang and V. Cuperman. An improved 2.4kbps class-dependent CELP speech coder. In *IEEE Int'l Conf. on Communications*, 1995.

[29] 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, Cuperman V, and A. Gersho, editors, *Advances in Speech Coding.* Kluwer Academic, 1991.

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

[31] W. B. Kleijn and J. Haagen. Transformation and decomposition of the speech signal for coding. *IEEE Signal Processing Letters*, 1(9):136–138, September 1994.

[32] W. B. Kleijn and J. Haagen. A speech coder based on decomposition of characteristic waveforms. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, Detroit, 1995.

[33] G. Kubin, B. S. Atal, and W. B. Kleijn. Performance of noise excitation for unvoiced speech. In *IEEE Workshop on Speech Coding for Telecommunications*, pages 35–36, 1993.

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

[35] P. Lupini. *Harmonic Coding of Speech at Low Bit Rates.* PhD thesis, Simon Fraser University, September 1995.

[36] P. Lupini and V. Cuperman. Spectral excitation coding of speech. In *IEEE Globalcomm*, 1994.

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

[38] J. S. Marques, L. B. Almeida, and J. M. Triboletll. Harmonic coding at 4.8 kb/s. In *IEEE Int'l Conf. on Acoustics, Speech and Signal Processing*, pages 17–20, 1990.

[39] J. C. De Martin and Allen Gersho. Mixed-domain coding and interpolation of voiced speech. In *IEEE Speech Coding Workshop*, pages 25–26, Annapolis, MD, Sept. 1995.

[40] R. McAulay and T. Quatieri. Low-bit rate speech coding based on the sinusoidal model. In S. Furui and M. Sondhi, editors, *Advances in speech signal processing*, chapter 6, pages 165–208. 1991.

[41] R. J. McAulay and T. F. Quatieri. Phase modeling and its application to sinusoidal transform coding. In *IEEE Int'l Conf. on Acoustic, Speech and Signal Processing*, 1986.

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

[43] R. J. McAulay and T. F. Quatieri. Sine-wave coding at low data rates. In *IEEE Int'l Conf. on Acoustic, Speech, and Signal Processing*, pages 577–580, 1991.

[44] 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. Acoustic, Speech and Signal Processing*, April 1994.

[45] A. V. Oppenheim and J. S. Lim. The importance of phase in signals. *Proceedings of IEEE*, 69(5):529–541, May 1981.

[46] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing.* Prentice Hall, Englewood Cliffs, NJ, 1989.

[47] W. A. Pearlman and R. M. Gray. Source coding of the discrete Fourier transform. *IEEE Trans. on Information Theory*, IT-24(6):683–692, 1978.

[48] T. F. Quartieri and R. J. McAulay. Mixed-phase deconvolution based on a sinusoidal representation. In *IEEE Int'l Conf. on Acoustic, Speech and Signal Processing*, 1987.

[49] T. F. Quatieri and R. J. McAulay. Phase coherence in speech reconstruction for enhancement and coding applications. In *IEEE Int'l Conf. Acoustic, Speech and Signal Processing*, pages 207–210, 1989.

[50] R. J. Sluijter R. Taori and E. Kathmann. Speech compression using pitch synchronous interpolation. In *IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 512–515, Detroit, 1995.

[51] S. Roucos and A. Wilgus. High quality time-scale modification for speech. In *IEEE Int'l Conf. on Acoustic, Speech and Signal Processing*, pages II145–II148, 1985.

[52] 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.

[53] D. Sen and W. B. Kleijn. Synthesis methods in sinusoidal and waveform-interpolation coders. In *IEEE Speech Coding Workshop*, pages 79–80, Annapolis, MD, Sept. 1995.

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

[55] T. E. Tremain. The government standard linear predictive coding algorithm: Lpc-10. *Speech Technology*, pages 40–49, April 1982.

[56] P. P. Vaidyanathan. Multirate digital filters, filter banks, polyphase networks, and applications : A tutorial. *Proceedings of IEEE*, 78(1):56–92, January 1990.

[57] P. P. Vaidyanathan. *Multirate System and Filter Banks.* Prentice-Hall, Englewood Cliffs, N. J., 1992.

[58] G. Zanellato Y. Gao and H. Leich. Band-widened harmonic vocoder at 2 to 4 kbps. *IEEE Int'l Conf. on Acoustic, Sp ch and Signal Processing*, 1995.

[59] R. Zopf, Y. Jiang, and V. Cuperman. Fixed-point strategies for a variable rate CELP implementation. In *Proc. DSPx*, 1995.