A Pitch Pulse Evolution Model for Linear Predictive Coding of Speech

Jacek Stachurski

Department of Electrical Engineering
McGill University
Montreal, Canada

February 1998

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

Speech coding is important in the effort to make more efficient use of digital telecommunication networks, particularly wireless systems, and to reduce the memory requirements in speech storage systems. The desire for a low-rate digital representation of speech is often contrary to the demand for a high quality speech reconstruction. In this thesis we present a new speech compression technique designed for near toll quality speech coding at bit rates as low as 4 kb/s.

In low-rate speech coding based on linear prediction (LP), poor modelling of the LP excitation for voiced, quasi-periodic segments contributes to the degradation of the quality of the reconstructed speech. In this dissertation, we present a new speech coding method designed for improved modelling of the LP excitation.

Conceptually, the LP excitation is decomposed into a series of underlying pitch pulses and a simultaneous unvoiced noise-like signal. The underlying pitch pulses are estimated from noisy observations, i.e., the pitch pulses extracted from the LP residual. Since the pulses change little from one time instant to another, we call our representation the Pitch Pulse Evolution (PPE) model. The PPE model provides a framework to analyze and effectively control the periodicity of voiced speech.

We have developed a robust algorithm for extracting noisy pitch pulses from the LP residual based on error minimization with respect to a set of model pulses, and we have examined a number of methods for calculating the underlying pulses. The evolving pitch pulse waveshapes, the pulse positions, and the unvoiced signal are encoded separately. The positions and the shapes of the underlying pulses need only be coded infrequently, and the characteristics of intermediate pulses are obtained by interpolation.

The software implementation of a 4 kb/s PPE coder is described. The main features of the implemented PPE coder are: a novel approach to pitch analysis; estimation of evolving pitch pulses which enables control over the pulse characteristics; and a unique coding scheme which avoids the time dilation and contraction of individual pitch pulses found in other waveform interpolation coders.

Sommaire

Le codage de la parole est essentiel dans les efforts pour obtenir un usage plus efficace des réseaux de télécommunication numériques, en particulier les réseaux cellulaires, et pour réduire la mémoire nécessaire dans les systèmes de stockage de la parole. La volonté d'avoir une représentation numérique de la parole à faible débit n'est pas souvent compatible avec la demande d'une reconstruction de la parole de haute qualité. Dans cette thèse, nous présentons une nouvelle méthode de compression de la parole permettant d'obtenir une reconstruction fidèle à des débits aussi faible que 4 kb/s.

Dans le codage de la parole à faible débit utilisant une prédiction linéaire (LP), la pauvre modélisation de l'excitation LP pour les segments voisés quasi périodiques contribue à la dégradation de la qualité de la parole reconstruite. Dans cette thèse, nous présentons une nouvelle méthode de codage de la parole conçue pour un meilleur modèle de l'excitation LP.

Conceptuellement, l'excitation LP est décomposée en une série d'impulsions de pitch sous-jacentes et en un signal non-voisé simultané qui peut être considéré comme du bruit. Les impulsions de pitch sous-jacentes sont estimées à partir d'observations contaminées par du bruit, i.e. les impulsions de pitch extraites du residu LP. Comme les impulsions changent peu d'un instant à l'autre, nous appelons notre représentation le modèle de l'évolution d'impulsions de pitch (PPE). Le modèle PPE fournit un cadre pour analyser et contrôler de façon efficace la périodicité de la parole voisée.

Nous avons développé un algorithme afin d'extraire les impulsions de pitch bruitées des résidus LP basé sur la minimisation de l'erreur par rapport à un ensemble d'impulsions modèles et nous avons examiné plusieurs méthodes pour calculer les impulsions sous-jacentes. Les formes des signaux d'impulsions de pitch qui évoluent, les positions des impulsions et le signal non-voisé sont codés séparément. Les positions et les formes des impulsions sous-jacentes ont seulement besoin d'être codées de façon sporadique et les caractéristiques des impulsions intermédiaires sont obtenues par interpolation.

Le programme informatique d'un codeur PPE à 4 kb/s est décrit. Les principales particularités du codeur PPE sont: une nouvelle approche de l'analyse de pitch; l'estimation des impulsions de pitch qui évoluent, ce qui permet de contrôler les caractéristiques de l'impulsion; et une méthode de codage unique qui élimine les dilatations et contractions de temps des impulsions individuelles de pitch présents dans les autres codeurs d'interpolation de forme de signal.

Acknowledgments

First of all I want to express my gratitude to my supervisor Prof. Peter Kabal for his invaluable guidance throughout the course of this work. He fostered and helped to shape the ideas and concepts presented here. His assistance shows not only in the contents of this thesis but also in the style of this presentation.

I would like to thank the Canadian Institute for Communication Research (CIRT) who financially supported this project. The research was conducted in the Telecommunications and Signal Processing (TSP) laboratory at McGill University and I would like to acknowledge the use of their very good facilities.

I am very thankful to my friends and colleagues in the TSP laboratory. I am thinking of those who are here now as well as those who have already left. The friendly and supportive atmosphere that they provided was just as important as their technical help when I needed it. I am obliged to Hossein for proofreading parts of my thesis, to Khaled who contributed to that as well, to Florence and Marc who helped me with the French abstract.

Some of my friends outside the university are particularly special to me. Glenn Seviss, being my longest roommate, proved to be also my best travel companion. Fran Yadao who has even read my thesis knowing little about engineering, and still less about speech compression, and despite being presently in Winnipeg. Rob Swick was usually there for me to talk about life in general (a great topic after a full day in front of the screen). I owe much to many other friends not mentioned here who have been part of my life in Montreal.

My very special thanks go to Agnieszka Rogińska who became particularly dear to me. She is responsible for some of the best times of my life, and I'm looking forward to more to come.

Finally, I thank my parents for their great love, support and many words of encouragement. I do appreciate that they could write me more letters in a month than I would in a year. They may not know it but they have a big part in this thesis being completed.

Contents

List of Figures

List of Tables

List of Acronyms

First appears on page

Chapter 1

Introduction

1.1 Motivation for Speech Coding

Speech communication is arguably the single most important interface between humans, and it is now becoming an increasingly important interface between human and machine. As such, speech represents a central component of digital communication and constitutes a major driver of telecommunications technology.

With the increasing demand for telecommunication services (e.g., long distance, digital cellular, mobile satellite, aeronautical services), speech coding has become a fundamental element of digital communication. Emerging applications in rapidly developing digital telecommunication networks require low bit, reliable, high quality speech coders. The need to save bandwidth in both wireless and wireline networks, and the need to conserve memory in voice storage systems are two of the many reasons for the very high activity in speech coding research and development. New commercial applications of low-rate speech coders include wireless personal communication systems (PCS) and voice-related computer applications (e.g., message storage, speech and audio over internet, interactive multimedia terminals).

In recent years, speech coding has been facilitated by rapid advancement in digital signal processing and in the capabilities of digital signal processors. A strong incentive for research in speech coding is provided by a shift of the relative costs involved in handling voice communication in telecommunication systems. On the one hand, there is an increased demand for larger capacity of the telecommunication networks. On the other, the rapid advancement in the efficiency of digital signal processors

and digital signal processing techniques have stimulated the development of speech coding algorithms. These trends are likely to continue, and speech compression most certainly will remain an area of central importance as a key element in reducing the cost of operation of voice communication systems.

1.2 The Basics of Speech Coding

1.2.1 Speech Production and Perception

In speech coding, the bit-rate reduction is achieved by removing the inherent information redundancies present in the speech waveform. The understanding of the basic properties of the speech signal and its perception is crucial to the design of a speech coder which would, ideally, parameterize only perceptually relevant information and thus compactly represent the signal.

When speech is produced, an airflow forced from the lungs passes through the larynx into the vocal tract. In the larynx, the elastic vocal folds can partially or completely obstruct the airflow creating a vocal tract excitation of turbulent noise or puffs of air. The opening between the vocal folds is called the glottis, and the air emanating from the vocal folds is often called the glottal excitation.

The speech signal can be roughly divided into voiced and unvoiced segments. During voiced speech the glottis periodically opens and closes and the glottal excitation has a periodic character. The excitation waveform corresponding to one cycle of glottal opening and closure is referred to as a glottal pulse, or pitch pulse. Consecutive pitch pulses may vary in their lengths and waveform shapes and the resulting glottal excitation is quasi-periodic.

For unvoiced speech, the glottal excitation is formed as the air forced through the constriction of the glottis creates a turbulence. The glottis does not open and close periodically but only contracts causing perturbations in the airflow. The unvoiced excitation does not display any apparent periodicity and has a noisy character.

The time properties of the speech production are reflected in the spectral features of the speech signal. The spectrum of the voiced excitation has a harmonic structure (i.e., sharp amplitude peaks at regular frequency intervals) with the fundamental frequency corresponding to the rate of the glottis closures. The spectrum of the unvoiced excitation has no prominent harmonics and it resembles the spectrum of a

white noise signal. The glottal excitation has no distinctive spectral envelope except for a spectral tilt during voiced speech. The spectral envelope, the broad peaks and valleys of the spectrum, is imposed on the glottal excitation by the vocal tract. For both, voiced and unvoiced excitation, the vocal tract acts as a filter shaping the frequency response of the speech signal.

The non-flat frequency response of the vocal tract introduces correlation between adjacent samples of the speech signal (short-term correlations). During voiced speech the periodic character of the excitation results in the correlation between the corresponding samples of adjacent pitch pulses (long-term correlation). In the spectral domain, the short-term correlation corresponds to the spectral envelope and the long-term correlation is reflected in the spectral fine structure. Both correlations introduce information redundancies in the speech signal and can be exploited in speech coding.

It is not known exactly what analysis is performed by the human hearing system. One of the often used properties of the auditory system is the spectral masking phenomenon. The spectral masking makes the inaccuracy of the signal representation which occurs in and near high-energy frequency bands less audible than the inaccuracy which occurs in other frequency regions. In the time domain, the human ear has a larger tolerance to the errors resulting from an inaccurate representation of high-energy samples than to the representation errors which coincide with low-energy samples. It is clear that both temporal and spectral characteristics of the speech signal are important and this is increasingly reflected in modern coders. In fact, coders which combine time domain and frequency domain analysis are strong contenders in the area of very low-rate speech coding.

1.2.2 Quantization

Quantization is an integral part of every speech coder. Most parameters and every waveform used to represent the speech signal must be quantized before they are encoded. A quantized value and the corresponding coded quantity may be equivalent. More often however, the coded value is an index to a quantized parameter or waveform selected from a set of permissible quantization outcomes. In the process of quantization a numerical value, or a vector of values, is represented with reduced precision. The difference between the original value (vector) and its quantized version is the quantization noise.

If a single value is quantized at a time we deal with scalar quantization. The value is represented by one of several fixed discrete values called quantization levels. In uniform scalar quantization the quantization levels are equally spaced. In logarithmic quantization the spacing is uniform on a logarithmic scale.

If a vector is represented with a fixed number of possible outcomes we perform vector quantization (VQ). The collection of the possible representations of a vector is referred to as a codebook. More than one codebook to represent a vector are often used. A large number of procedures have been proposed to create, organize, and search the codebooks. Such methods include tree-structured VQ, transform VQ, product code VQ, split VQ, gain-shape VQ, multistage VQ, hierarchical VQ (Gersho and Gray 1992). The method employed depends on the properties of the vector to be quantized and the desired criteria the quantized representation should satisfy. For example, in quantizing the linear prediction parameters used to represent the vocal tract characteristics in linear prediction coding, split VQ is often used. In modelling the linear prediction filter excitation, gain-shape VQ and multistage VQ are employed.

Although scalar quantization is still used for quantizing some of the parameters, it is the application of vector quantization which enables significant reduction of the number of bits required to efficiently represent the speech signal.

1.2.3 Pulse Code Modulation

In pulse code modulation (PCM) coding the speech signal is represented as a series of quantized values which correspond to the amplitudes of the speech samples. In uniform 128 kb/s PCM, for example, narrow-band speech (200–3400 Hz) is sampled at 8 kHz and represented with 16 bits per sample by the means of uniform scalar quantization. In μ-law and A-law log-PCM, the samples are logarithmically quantized with 8 bits which results in the bit rate of 64 kb/s. Eight-bit log-PCM coders are widely used in network telephony.

Uniform PCM does not exploit any specific properties of speech and is valid for any band-limited signal. Log-PCM takes advantage of the nonuniform distribution of speech amplitudes and the fact that the louder the sound the less sensitive the human ear becomes to small changes in the intensity of the sound. The latter property allows the increase of the quantization noise in the regions of high energy without significant loss of the reconstructed speech quality. PCM coding does not take advantage of the

existing correlations between speech samples.

The correlation between adjacent samples is exploited in differential PCM (DPCM). In DPCM the difference between the current sample and its predicted value is quantized and transmitted. In a simple linear prediction the current sample is predicted from a number of past, reconstructed samples. In adaptive DPCM (ADPCM), the linear predictor or/and the quantization levels are varied based on the characteristics of the past reconstructed speech signal. The same predictor/quantizer modifications are performed by the encoder and the decoder. If the modifications are based on the reconstructed speech samples, the information about the modifications need not be encoded.

The speech quality achievable with 64 kb/s log-PCM coding and 32 kb/s ADPCM is referred to as "toll" quality. The toll quality rating constitutes the reference point with respect to which the performance of lower bit rate coders is often compared.

1.2.4 Attributes of Speech Coders

The main attributes of a speech coder include: (i) the bandwidth of the speech signal for which the coder is intended, (ii) bit rate of the compressed signal, (iii) reconstructed speech quality, (iv) complexity and delay of the coder, (v) sensitivity of the coder to background acoustical noise, (vi) sensitivity of the encoded bits to transmission channel errors. Different applications require coders optimized for different features. In message transmission systems, for example, low-delay of the coder may not be an issue, and central storage systems may not require a low-complexity implementation of the coder. While in a large number of applications the primary goal is to ensure the perceived similarity between the original and the reconstructed signal, in some cases (i.e., in the systems in which security is the main concern) it is sufficient that the reconstructed speech sounds intelligible and natural. In general, the central trade-off in speech coding is between the bit rate of the compressed signal and the perceptual quality of the reconstructed speech. In most commercial applications real-time implementation of the coder is required. A real-time implementation imposes constraints on both the complexity and the delay of the coder.

1.2.5 Evaluating the Performance

One of the major difficulties in designing and testing various speech coders is the lack of an objective quality measure to represent the perception-based goals in the form of an error function between the original and the reconstructed signal. The most commonly used objective criteria (signal-to-noise ratio, segmental signal-to-noise ratio, log spectral distance) are sensitive to gain variations and delays between the original and coded speech. They also usually do not fully account for perceptual properties of the hearing system. A number of objective methods based on human auditory perception models have been proposed (Schroeder *et al.* 1979, Wang *et al.* 1992, Paillard *et al.* 1992, Jayant *et al.* 1993, De 1993), but none has yet eliminated the necessity of subjective testing.

The most commonly performed subjective tests are absolute category rating (ACR) tests of which one example is the Mean Opinion Score (MOS) test (described for example by Kroon 1995). In the MOS test a number of listeners are asked to evaluate the quality of recorded speech according to a five-level scale. For narrow-band speech, a score of 4–4.5 implies toll quality and a score between 3.5 and 4 indicates communications quality. Scores below 3.5 mean that the reconstructed speech is of poor quality; synthetic speech often scores in the range 2.5–3.5. The MOS scores can differ from one test to another significantly, often due to cultural and/or linguistic biases, and therefore are not an absolute comparison between coders.

Subjective testing in general is time consuming and therefore expensive. Many proposed coders have not been subjected to rigid testing and the reported results are difficult to calibrate.

1.3 State-of-the-Art Coders

In the current state-of-the-art coders a noticeable coding noise appears at bit rates below 8 kb/s. The coded speech is natural, intelligible, the speaker is easily identified and his/her intonation is preserved, but the distortion is noticeable even though not annoying. This corresponds to MOS values above 3.5 and below 4.0. The naturalness is slightly lost at rates 2–4 kb/s. The coded speech has also increasing noisy "hoarse" quality with a varying degree of buzziness. Such coders usually obtain MOS values of 3.0–3.5. At rates below 1 kb/s the speaker identity and naturalness are mostly lost.

Techniques that have been especially successful in achieving high quality speech at low bit rates include linear prediction (LP) coding and sinusoidal coding. Linear prediction coders operate mainly in the time domain while sinusoidal coders perform most of their analysis in the frequency domain. Some of the recent work (for example Waveform Interpolation) can be viewed as an attempt to combine time domain and frequency domain analysis.

1.3.1 Linear Prediction Analysis-by-Synthesis Coders

A particularly successful group of the LP coders is comprised of coders which use analysis-by-synthesis techniques (Kroon and Deprettere 1988, Kroon and Kleijn 1995, Cucchi *et al.* 1996). In linear prediction analysis-by-synthesis (LPAS) coding, the reproduced speech is synthesized by filtering an excitation signal with a time-varying linear filter. The coefficients of the synthesis filter are determined by linear prediction analysis of the speech signal. The excitation is determined by filtering excitation candidates with the synthesis filter and selecting the one which minimizes a perceptually weighted distortion measure between the reconstructed and the original signal. LPAS coders include multi-pulse LP (introduced by Atal and Remde 1982), Regular-Pulse Excitation (RPE) (introduced by Kroon *et al.* 1986) and, most studied to date, Code-Excited Linear Prediction (CELP) (introduced by Atal and Schroeder 1984, Schroeder and Atal 1985). The initially large computational complexity of CELP was significantly reduced through the subsequent improvements (Davidson and Gersho 1986, Trancoso and Atal 1990, Kleijn and Krasinski 1990, Gerson and Jasiuk 1991a, Elshafei-Ahmed and Al-Suwaiyel 1993, Moreau and Dymarski 1994), and over the years CELP became the most widely used speech coding technique.

The success of the LPAS technique is reflected in the fact that many low-rate speech coding standards adopted in the last few years are LPAS coders. In Europe the Regular-Pulse Excitation with Long-Term Prediction (RPE-LTP) coder at 13 kb/s (MOS $\sim$ 3.6) was chosen as a standard for the GSM (Global System for Mobile Telecommunications) digital cellular telephony. The Vector Sum Excited LP (VSELP) coder (Gerson and Jasiuk 1991b) operating at 5.6 kb/s (MOS $\sim$ 3.5) was selected as the corresponding half-rate standard. In the North American digital cellular telephony VSELP operating at 7.95 kb/s (MOS $\sim$ 3.5) and Qualcomm CELP (QCELP) (DeJaco *et al.* 1993) at 8.5 kb/s (MOS $\sim$ 3.4) were chosen as interim

standards. The U.S. government has adopted a 4.8 kb/s CELP coder (MOS $\sim$ 3.2) as the secure voice communication standard (Campbell *et al.* 1989). The International Telecommunication Union (ITU) has very recently adopted a new standard for 8 kb/s toll quality coding — G.729; the standard is a CELP-based coder with MOS $\sim$ 4.0 (Salami *et al.* 1994, 1995).

1.3.2 Frequency Domain Coders

Although the LPAS-based coders produce very high quality speech in the range of 4–16 kb/s, their performance degrades rapidly around 4 kb/s (Atal and Caspers 1991, Tzeng 1991), at which point the performance of time domain waveform matching (even with a carefully chosen perceptually-weighting error criterion) deteriorates. A viable alternative to LPAS coders, particularly in the range 2–4 kb/s, is comprised of coders which directly use frequency representations in their analysis. The most prominent frequency domain techniques for low-rate coding are: harmonic coding (Almeida and Tribolet 1982, Marques *et al.* 1990), Sinusoidal Transform Coding (STC) (McAulay and Quatieri 1986, McAulay *et al.* 1991, McAulay and Quatieri 1995), and coding based on Multi-Band Excitation (MBE) (Hardwick and Lim 1988, 1989, Brandstein *et al.* 1990). The three coding methods are sometimes grouped under a common name as sinusoidal coders.

In the sinusoidal coding, the spectral peaks of a short time Fourier transform are identified and the speech signal is reconstructed by interpolation of the amplitudes, the phases and the frequencies of a set of sine waves. Although the amount of work on sinusoidal coders has been small compared to CELP, there are many indications that this is a promising approach for the future (Gersho 1994). For example, an Improved Multi-Band Excitation (IMBE) coder (Brandstein *et al.* 1990) operating at 4.15 kb/s (MOS $\sim$ 3.3) was selected by Inmarsat as a standard for satellite voice communications.

An insightful comparison between CELP and the sinusoidal coding is offered by Trancoso *et al.* (1990). The authors argue that the two techniques are complementary and might well be merged in future systems.

1.3.3 Waveform Interpolation Coders

Waveform Interpolation (WI) is an attempt to combine aspects of the time domain and the frequency domain analysis. Prototype Waveform Interpolation (PWI) (Kleijn 1991, Kleijn and Granzow 1991) and Time-Frequency Interpolation (TFI) (Shoham 1992, 1993b) techniques are precursors to the recent WI (Kleijn and Haagen 1994b, 1995b). A WI coder implemented at 2.4 kb/s demonstrated very high quality of synthesized speech with MOS $\sim$ 3.5 (Kleijn and Haagen 1995a, Kleijn *et al.* 1996). Over the last couple of years many techniques have been suggested for use within the WI framework (Tanaka and Kimura 1994, Burnett and Bradley 1995, Jiang and Cuperman 1995, Festa and Sereno 1995, Tang and Cheetham 1995). Although interpolation of the prototype waveforms is usually performed in the frequency domain, interpolation in the time domain has also been implemented with good results (Yang *et al.* 1995).

Similarities and differences between WI and STC are examined in (Sen and Kleijn 1995) and (Kleijn and Haagen 1995b).

1.3.4 Other Coders

A number of other coders have been implemented at bit rates below 4 kb/s with good results. The five coders evaluated in the second stage of the competition for the U.S. government standard for 2.4 kb/s secure voice communication were: an IMBE coder, a STC coder, a WI coder, a Pitch Synchronous Excited Linear Prediction (PSELP) coder (Fette *et al.* 1993), and a Mixed Excitation Linear Prediction (MELP) coder (McCree and Barnwell III 1993, 1995). The last two coders, the PSELP coder and the MELP coder, use linear prediction but they do not choose the LP excitation based on analysis-by-synthesis; both coders perform part of their analysis in the frequency domain. The MELP coder was selected as the winning candidate for the aforementioned U.S. federal standard (McCree *et al.* 1996).

1.4 Objectives and Scope of Our Research

In this thesis we are concerned with telephone quality speech band-limited from 200 Hz to 3.4 kHz. The analog signal is sampled at 8 kHz and represented with 16 bits uniform PCM resulting in a digital signal with the bit rate of 128 kb/s. Our goal is to

represent this digital signal with a bit stream of about 4 kb/s with the reconstructed speech signal very close or equivalent to toll quality.

We propose a new speech coding method based on our Pitch Pulse Evolution (PPE) model. We introduce and describe the PPE model in the context of existing coding systems and we present an implementation of a 4 kb/s PPE coder.

Our coding system is constrained to have moderate complexity and algorithmic delay[†]. A moderate complexity coder is a coder which is implementable on a single fixed-point 16-bit DSP chip which can perform about 40 million instructions per second (MIPS). Moderate algorithmic delay is understood to be about 50–60 ms, which includes the processed speech block and the look-ahead. Such moderate complexity and delay is a requirement for real-time operation in the context of applications used for conversational speech.

In this work we have concentrated on achieving high quality reconstructed speech. We have not been directly concerned with the sensitivity of the coder to background acoustic noise or with the sensitivity of the encoded bits to transmission errors. However, the final configuration has elements of similarity to existing coders and as such we do not expect the PPE coder to be unduly sensitive to these factors.

1.5 Organization of the Thesis

The organization of this thesis is as follows. In Chapter 2 we review the principles of linear prediction analysis-by-synthesis (LPAS) coding in more detail and discuss ways of representing the LP excitation. In LP coding poor representation of the excitation for voiced segments is to a large extent responsible for the degradation of speech quality with decreased bit rate. Various techniques for improving the quality of voiced speech are examined.

In Chapter 3 a general formulation of the pitch pulse evolution PPE model is presented and demanding requirements are imposed on the pitch pulse extraction algorithm. The problem of estimating the evolving pitch pulses is discussed and several methods of the estimation are investigated.

The focus of Chapter 4 is on pitch interpolation. We compare the pitch interpolation used in Waveform Interpolation (WI), Sinusoidal Transform Coding (STC),

[†]Algorithmic delay is the sum of (i) the length of currently processed block of speech, (ii) the length of the look-ahead which is needed to process the samples of the current block.

and Relaxed-CELP (RCELP). The pitch pulse length and the pitch pulse waveshape interpolations used in the PPE model are described.

Chapter 5 presents the implementation of a PPE coder with emphasis on the components which are unique to our coder. Among others, a practical and robust method for extracting individual pitch pulses from the LP residual is developed and the strategy for encoding the pitch information is specified. We discuss the results of informal comparison tests of quality of the PPE coded speech with respect to the original signal and with respect to the speech coded with G.729.

Our work is summarized and the future research directions are outlined in Chapter 6. This chapter also states the contributions of this thesis and lists the claims of originality in our work.

Chapter 2

Modelling the Excitation in Linear Predictive Coding

In a general speech production model air flows from the lungs to the larynx where it is forced through a variable opening between vocal folds (vocal cords). The opening between the vocal folds is called the glottis and the airflow which emanates from the folds is often referred to as the glottal excitation. The excitation passes through the vocal tract which can be modelled as an acoustic tube. The speech signal is created as the air exits the vocal tract causing a waveform of air pressure variations.

The characteristics of the vocal tract are determined by the shape of the passage though which the glottal excitation flows. The air-flow passage is shaped by: the oral and nasal cavities, the tongue, the teeth, the lips, and a number of other articulators (see for example O'Shaughnessy 1987). The shape of the air passage influences the transfer function of the vocal tract. The effects of the glottal excitation and the vocal tract are considered to be independent (Rabiner and Schafer 1978, O'Shaughnessy 1987, Deller Jr. *et al.* 1993), which is justified by the fact that the interaction between the vocal tract shape and the lung pressure is negligible with respect to other simplifications of the model.

2.1 Voiced and Unvoiced Speech

The speech signal can be roughly divided into voiced and unvoiced segments. For voiced speech the excitation is generated by a periodic opening and closing of the

glottis resulting in a series of similar pitch pulses. During unvoiced speech the glottal excitation is a flat power spectrum noise, which is often modelled by a random noise generator. A general, simplified speech production model is presented in Fig. 2.1. The relative ratio of the unvoiced and voiced components is controlled in the model by adjusting the corresponding gains. For a "purely voiced" signal the noise source gain is zero, and for a "purely unvoiced" signal the voice source gain is set to zero.

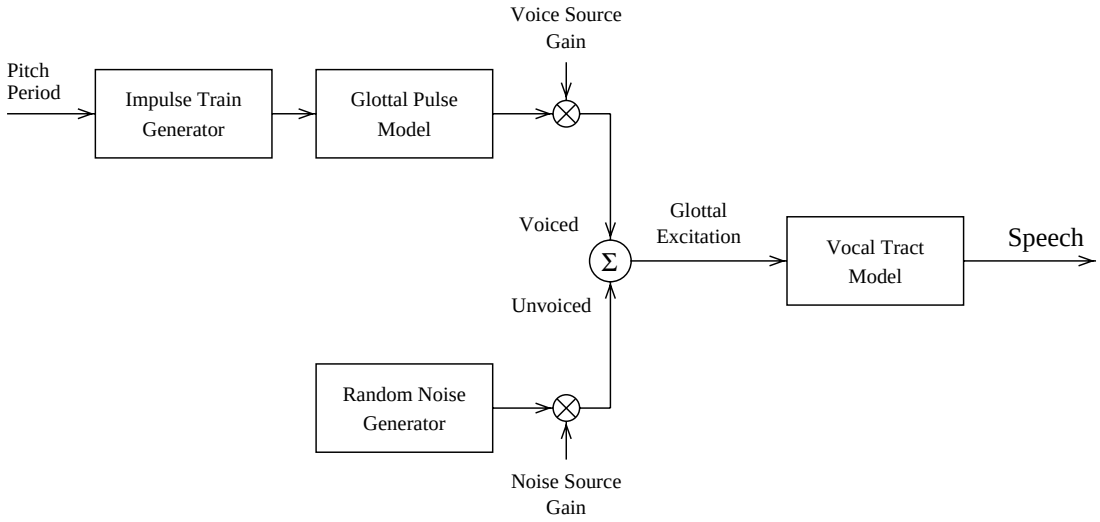


Fig. 2.1 General speech production model.

A short-time power spectrum of speech calculated with a smooth time window of 30 ms displays the basic characteristics of the speech signal (Fig. 2.2). We can identify the fine spectral structure due to the glottal excitation and the spectrum envelope imposed by the vocal tract. During voiced speech, one can observe in the fine structure regularly spaced harmonics which are the result of periodic oscillations of the vocal folds. During unvoiced speech, the fine structure does not display any apparent harmonic makeup. The unvoiced excitation is noise-like (with no periodicity evident). The broad peaks of the spectral envelope correspond to resonances of the acoustic tube of the vocal tract. The resonances are called formants and the vocal tract is said to impose a formant structure on the glottal excitation.

The fine structure of the spectrum is related to long-term correlation of the samples of the signal in the time domain. During voiced speech, the harmonic spectral structure implies a similarity of sequential cycles of the pitch period. For unvoiced speech, the long-term sample correlation is very small or nonexistent. The spectral

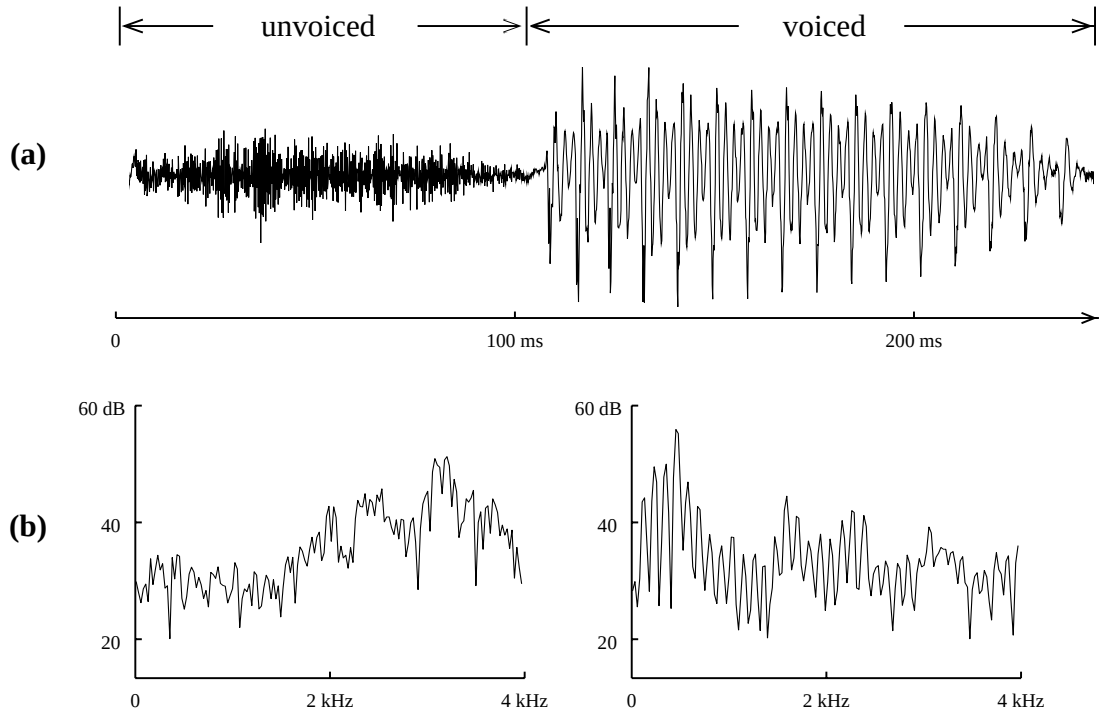


Fig. 2.2 (a) Unvoiced and voiced segments of speech signal. (b) Power spectra calculated in the unvoiced and the voiced regions respectively. The power spectra were calculated over segments 30 ms long smoothed with a Hamming window.

envelope corresponds to short-term correlations between nearby samples. Both the long-term and the short-term correlations are important and they are exploited in speech coders.

In linear predictive (LP) coding a linear filter of the form

$$\frac{1}{A(z)} = \frac{1}{1 - \sum_{i=1}^{N} a_i \, z^{-i}} \tag{2.1}$$

models the short term correlation in the speech signal (spectral envelope) introduced by the vocal tract. The LP filter coefficients $a_1, \ldots, a_N$ are estimated and transformed into a set of parameters judged to have better coding properties and error robustness. The LP parameters are then coded and transmitted.

The glottal excitation is modelled based on the LP residual and/or the error

between the original and the reconstructed speech. The LP residual is formed by filtering the speech signal with the time-varying LP analysis filter

$$A(z) = 1 - \sum_{i=1}^{N} a_i \, z^{-i} \, . \tag{2.2}$$

Voiced and unvoiced speech and the corresponding LP residual are presented in Fig. 2.3. The LP residual roughly corresponds to the glottal excitation which emanates from the larynx. One can observe randomness of the LP residual within the unvoiced region and well defined energy peaks within the voiced region. The peaks correspond to the pitch pulses present in the voiced excitation.

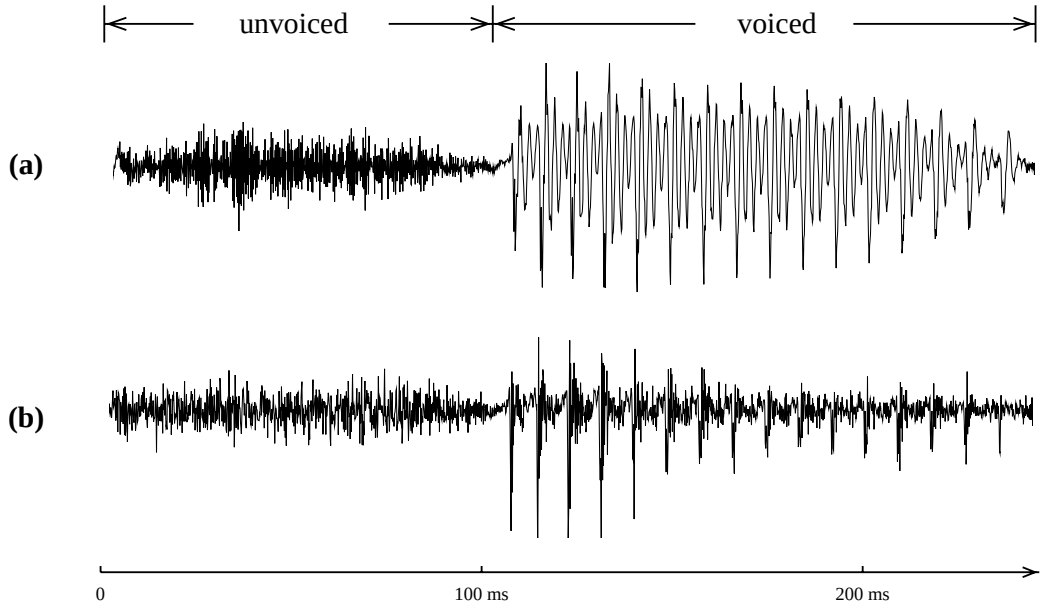


Fig. 2.3 Unvoiced and voiced segments of (a) speech signal (b) LP residual (scaled by a factor of 2).

It has been shown (Kubin *et al.* 1993) that the LP model is capable of producing very high quality speech for the unvoiced regions even if no bits are assigned to coding the excitation vector (the excitation is generated as a series of independent Gaussian random numbers). This is possible because the noise-like excitation of the unvoiced speech contains little perceptually important information. The difficulty of coding voiced speech at low bit rates stems from the fact that the human ear is particularly sensitive to small changes in the speech periodicity. At low bit rates, the small changes

between consecutive pitch pulses are very hard to model with the small number of bits available per coding update.

2.2 Linear Prediction Analysis

The LP filter coefficients are determined from the speech signal using linear prediction techniques (Makhoul 1975, Markel and Gray 1976, Rabiner and Schafer 1978, Deller Jr. *et al.* 1993). The traditional auto-correlation and covariance methods of calculating the LP coefficients have new alternatives such as discrete all-pole modelling (El-Jaroudi and Makhoul 1991). Many improvements to the basic estimation methods of the LP coefficients are summarized by Paliwal and Kleijn (1995).

The update rate for the LP coefficients is related to the characteristics of the vocal tract. Most of the time, the shape of the vocal tract changes relatively slowly in time. The vocal tract articulators move usually less than 1 cm at a time at speeds up to 30 cm/s O'Shaughnessy (1987). This translates into a change period of about 30 ms although during slow speech the shape of the vocal tract may not change for up to 200 ms. The vocal tract characteristics can also change rapidly, e.g., when the air-flow passage of the vocal tract closes or opens at the lips. The LP coefficients are calculated with update rates varying from 30 to 100 times per second (every 30 to 10 ms).

The number of calculated LP coefficients is related to the number of formants present in the spectrum of the speech signal. The vocal tract imposes formant structure on the glottal excitation with an average of one formant per 1 kHz. A few coefficients are used to better approximate the spectral valleys and general shape of the spectrum. The number of calculated coefficients is often equal to 10–12 per update.

The LP coefficients are usually not coded directly but first transformed into a set of parameters which has desirable coding properties. Various representations of the LP coefficients have been proposed. Currently the most popular are Line Spectral Frequencies (LSF) also known as Line Spectral Pairs (LSP) (Soong and Juang 1984). Other representations include reflection coefficients, log-area ratio, cepstral coefficients, and the LP filter impulse response (Rabiner and Schafer 1978).

Considerable work has been done in developing efficient quantization methods for

the LP parameters. Scalar quantizers achieve transparent coding[†] at rates of 32 bits per update with 50 updates per second, which results in coding rate of 1.6 kb/s (Soong and Juang 1993). Vector quantization techniques (Gray 1984, Makhoul *et al.* 1985, Gersho and Gray 1992) have provided means to transparent coding at rates as low as 24 bits per update which, with 50 updates per second, results in a rate of 1.2 kb/s (Paliwal and Atal 1993). Developments in LP analysis and coding are reviewed for example by Paliwal and Kleijn (1995).

The LP model assumes independence between the excitation and the parameters of the linear filter so that separate analysis and interpolation of the LP parameters and the LP excitation can be performed. The LP parameters are often up-sampled (interpolated) to the rate of 200–400 parameter sets per second. The interpolation of the LP parameters in different domains has been studied recently by Paliwal (1995) with indication that the interpolation in the LSF domain has desirable properties.

Although major progress has been made in reducing the bit rate for encoding the LP parameters, the bit rate for the transparent encoding of the LP excitation still remains very high. A multitude of methods for representing the excitation signal have been proposed but the lack of an efficient representation of the excitation still remains a major obstacle in synthesizing high quality speech at low bit rates (Atal and Caspers 1991). We review a number of techniques which aim at improved coding of the LP excitation.

2.3 Modelling the LP Excitation with Fixed-Length Analysis

In this section we describe linear prediction analysis-by-synthesis (LPAS) coding. Although analysis-by-synthesis does not have to be performed with fixed analysis block lengths, it was first developed in such a context (Kroon and Deprettere 1988). Some coders which perform their analysis with pitch-synchronous block lengths, for example the WI coder discussed later, also employ elements of analysis-by-synthesis coding, i.e., error measurement with respect to the perceptually weighted speech. First, the LPAS coding is presented. Then, some of the proposed improvements in the representation of the LP excitation signal for Code-Excited Linear Prediction (CELP) coders are examined. Finally, the generalized analysis-by-synthesis paradigm

[†] Transparent coding of a parameter generally means that the coding of the parameter does not introduce any perceptual distortion in the reconstructed signal.

is introduced.

2.3.1 Analysis-by-Synthesis

In the analysis-by-synthesis procedure, the parameters describing the LP excitation are determined by minimizing the perceptually weighted mean square error between the original and the reconstructed speech (Kroon and Deprettere 1988) (Fig. 2.4).

The perceptual weighting exploits the masking properties of the human hearing system. The masking makes the noise in and near frequency bands of high energy less audible than the noise at the frequencies corresponding to the energy valleys. The perceptual weighting filter emphasizes the error in the spectral valleys of the input speech and deemphasizes the error in the regions of spectral peaks. As the effect, the quantization noise in the valleys is reduced and the noise near the peaks is increased. This increased noise on the spectral peaks is masked by the human auditory system.

The perceptual weighting is often specified as a filter

$$W(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)}, \quad 0 < \gamma_1 < \gamma_2 \leq 1 \tag{2.3}$$

where $A(z)$ is the LP analysis filter as given in (2.2). The values γ_1 are γ_2 are fixed or adaptive.

The analysis-by-synthesis approach is often call "closed loop" analysis as opposed to "open loop" analysis in which the parameters are determined without reconstruction of the speech signal. The "closed loop" analysis is usually computationally more expensive than the "open loop" approach. In practice those two are often combined. "Open loop" analysis provides a set of initial candidates for parameter representation and the "closed loop" analysis serves as a final criterion for selecting the best set of parameters. Many techniques for reducing the computational complexity of the LPAS coders have been reviewed by Kroon and Deprettere (1988), Gersho (1994), Kroon and Kleijn (1995).

In the improvements of the coded speech quality, the emphasis is on perceptually accurate representation of the periodicity of the coded LP excitation. Poor representation of speech periodicity in the voiced regions is the main shortfall of LPAS coders operating at low bit rates.

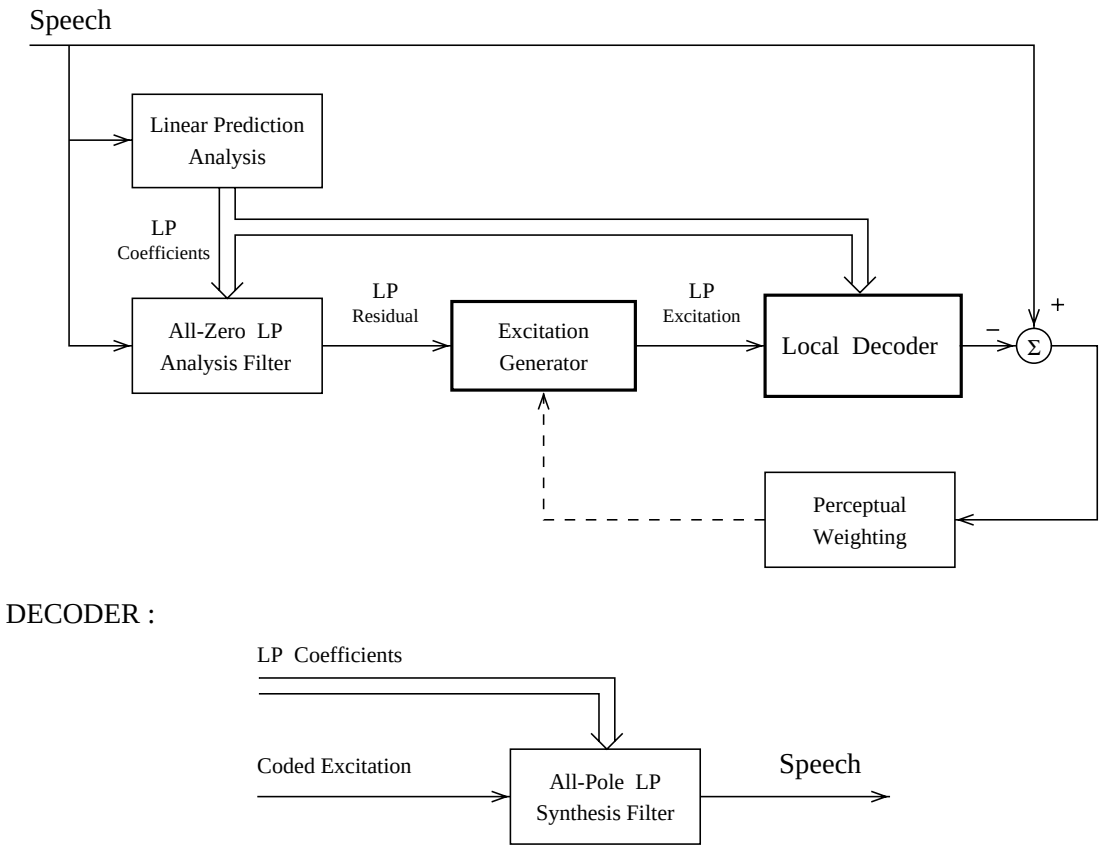


Fig. 2.4 Linear prediction analysis-by-synthesis coding.

2.3.2 Code-Excited Linear Prediction (CELP)

In a CELP coder the LP excitation is modelled using vector quantization (VQ). The vectors which represent the LP excitation are selected with the analysis-by-synthesis procedure.

Stochastic Codebook

The unvoiced part of the excitation is modelled in CELP by the so-called stochastic codebook. The stochastic codebook is also used to model the start and changes of the voiced excitation. The same fixed codebook is used at the transmitter and the receiver. The index to the selected codebook entry is transmitted.

In early CELP the entries of the stochastic codebook were populated with Gaus-

sian independent random numbers. The search of such an unstructured codebook necessitates a very high computational complexity. To reduce this complexity and to reduce the required storage space, a variety of structural constraints have been imposed on the codebook. The proposed structures of the stochastic codebook include overlapped codebooks, sparse codebooks and algebraic codebooks (see for example Kleijn and Krasinski 1990, Gersho 1994).

More than one codebook may be used to represent the unvoiced contribution; the configuration of multiple codebooks is often called multistage VQ (Gersho and Gray 1992). In multistage VQ, the excitation vector is generated as a sum of scaled entries from several codebooks which are sequentially searched. The sequential search of multiple codebooks is suboptimal and a joint search usually introduces excessive complexity. To approach the optimal selection, the orthogonalization of the multiple codebooks is used (Gerson and Jasiuk 1991b, Moreau and Dymarski 1994).

Pitch Filter

A simple model of the periodicity present in the LP residual can incorporate a pitch filter. The pitch synthesis filter is specified as

$$\frac{1}{P(z)} = \frac{1}{1 - \beta\, z^{-M}}\,, \tag{2.4}$$

where β and M are respectively the gain coefficient and the pitch lag. The lag M approximates the periodicity (or the pitch period) of the signal. The gain β can be interpreted as an indicator of the "level of periodicity" with β approaching the value of 1 for "very periodic" signals. Although the parameters of the pitch filter are determined via analysis-by-synthesis ("closed-loop"), the initial estimate of the parameters is often performed with "open-loop" methods. The properties of the pitch filters have been studied for example by Ramachandran and Kabal (1987, 1989).

For voiced speech, the pitch period varies typically from 2 to 20 ms. For the 8 kHz sampling rate, to facilitate 7-bit encoding, the range of the delays is often limited from 20 to 147 samples (128 possible delays). The update rate of the pitch predictor parameters is higher than that of the LP parameters, typically 200 times per second (every 5 ms). The gain coefficient can be encoded with 3 to 4 bits per update, which means that the bit rate for coding only the pitch information is from 2 to 2.2 kb/s

(higher than the coding rate of the LP parameters).

Multi-tap pitch filters have been also suggested with reported better performance than the single-tap filter. Their disadvantage, however, is an even higher bit rate needed to encode the filter coefficients. Using vector quantization with 5–7 bits to code the coefficients of a three-tap filter results in the pitch-information coding rate of 2.4–2.8 kb/s.

Fractional Pitch

In an important contribution to modelling the LP residual for voiced speech, the search of the pitch period is refined to a fraction of a sample (the analysis is performed with sub-sample resolution) (Kroon and Atal 1990, Marques *et al.* 1990). The fractional pitch is used as an alternative to multi-tap pitch filters. Although the fractional pitch increases the bit rate of the coded pitch information, the technique is now used in many CELP implementations. Most coders use a nonuniform spacing with higher resolution for shorter delays. No more than one or two additional bits per update are often used and the total bit rate is increased only by 200–400 b/s.

Adaptive Codebook

In most of the modern implementations of the CELP coder, the pitch filter is represented as a codebook called the adaptive codebook (Kleijn *et al.* 1988b,a). In contrast with the adaptive codebook, the stochastic codebook is often called the fixed codebook. A CELP coder with two codebooks, the adaptive and the fixed codebook, is shown in Fig. 2.5.

In a sequential search, first every entry of the adaptive codebook is tried and the vector which minimizes the perceptually weighted error is selected. The optimal gain for the selected vector is calculated. The adaptive codebook contribution, multiplied by the optimal gain, is used during the search for the fixed codebook entry. Again, the vector which minimizes the perceptually weighted error is selected and the optimal gain for the fixed codebook contribution is calculated.

The adaptive codebook can be interpreted as a generalization of a pitch filter. As a special case, the codebook entries can be formed from the output of a pitch filter applied to the past LP excitation. In a case of a single-tap pitch filter, the entries of

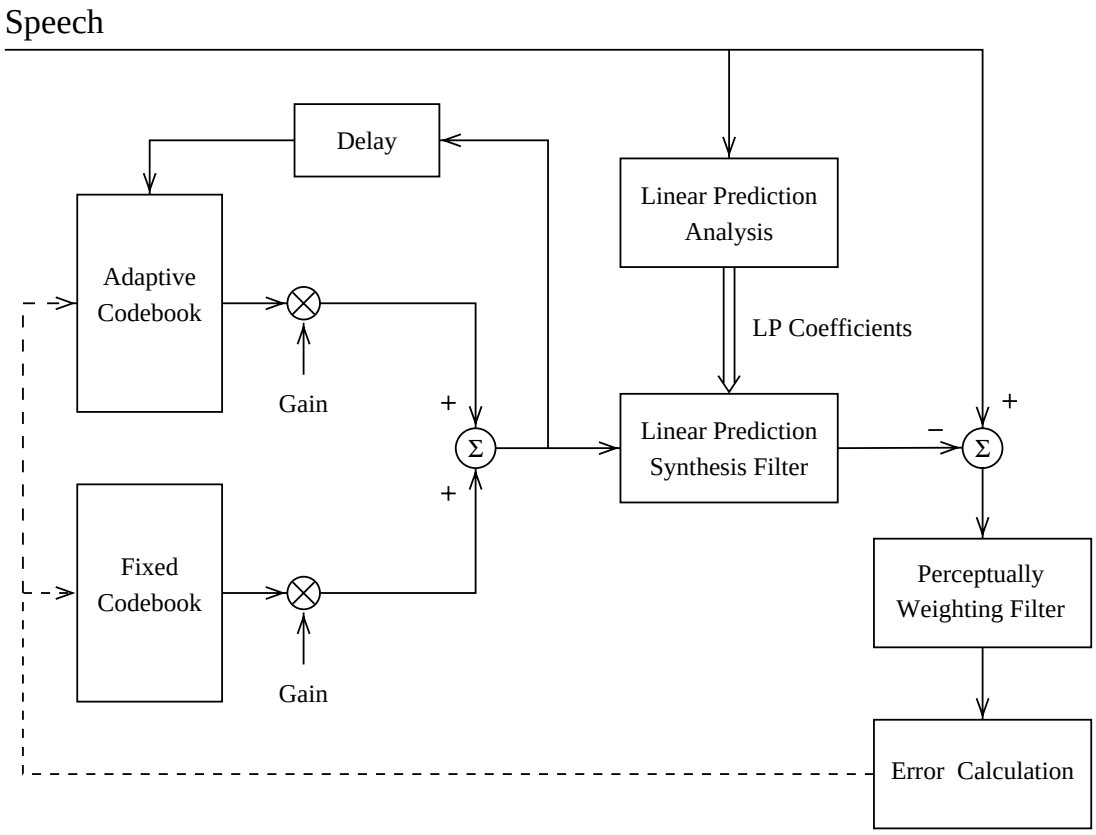


Fig. 2.5 CELP coder implementation using an adaptive and a fixed codebook.

the adaptive codebook would be formed from the samples of the past excitation and the gain of the codebook would be the coefficient of the pitch filter.

In a two-codebook structure† CELP of Fig. 2.5, controlling the periodicity of the LP excitation can be achieved by (i) modifying the error weighting and thus influencing the selection of the adaptive and fixed codebook entries, (ii) controlling the relative gain of the two codebooks when forming the LP excitation, (iii) a specific way of forming the entries of the adaptive codebook and/or the fixed codebook.

To better describe the various techniques used to improve the representation of the LP excitation, the operations performed in a CELP coder are classified into three stages (Fig. 2.6):

†As mentioned earlier, several codebooks are sometimes used to represent the stochastic contribution in which case the fixed codebook is implemented as more than one codebook.

(i) The entries of the codebooks are selected and gains of the codebooks are calculated.

(ii) The selected entries are combined to form the LP excitation and the coded speech is synthesized. The codebook gains can be used as calculated in stage (i) or they can be updated.

(iii) The adaptive codebook is updated.

The operations of stages (i) and (iii) are performed at the encoder. The task of the encoder is to code the parameters which are to be used in stage (ii). The decoder performs the operations of stages (ii) and (iii).

Stage (ii) includes post-filtering for which parameters are not directly coded. The post-filter parameters are determined from the LP filter parameters, the gains of the codebooks, and the index of the adaptive codebook. Post-filtering is included in the same functional block as the LP synthesis filter because a pitch post-filter often precedes the LP synthesis filter and a formant post-filter usually follows the LP synthesis filter. We do not consider post-filtering as a method which attempts to refine the representation of the LP excitation. Post-filtering in many cases improves the quality of the reconstructed speech but it does not contribute to the *modelling* of the LP excitation.

We now describe a number of methods which, by modifying the operations in stages (i), (ii) and (iii), aim to improve the modelling of the LP excitation.

Harmonic Noise Weighting

In harmonic noise weighting (Gerson and Jasiuk 1991a), a multi-tap pitch filter is used to further weight the perceptually weighted error. By changing the error weighting, the selection of the codebook entries is influenced. The calculated gains are also, in general, different. Harmonic noise weighting deemphasizes the error at pitch harmonics and the fixed codebook contribution is steered to better match the signal spectrum between the harmonics. The method takes advantage of the property of auditory masking which suggests that the noise between the harmonics is more audible than the noise on the harmonics. The gains calculated in stage (i) are used in stages (ii) and (iii). The method modifies stage (i) only.

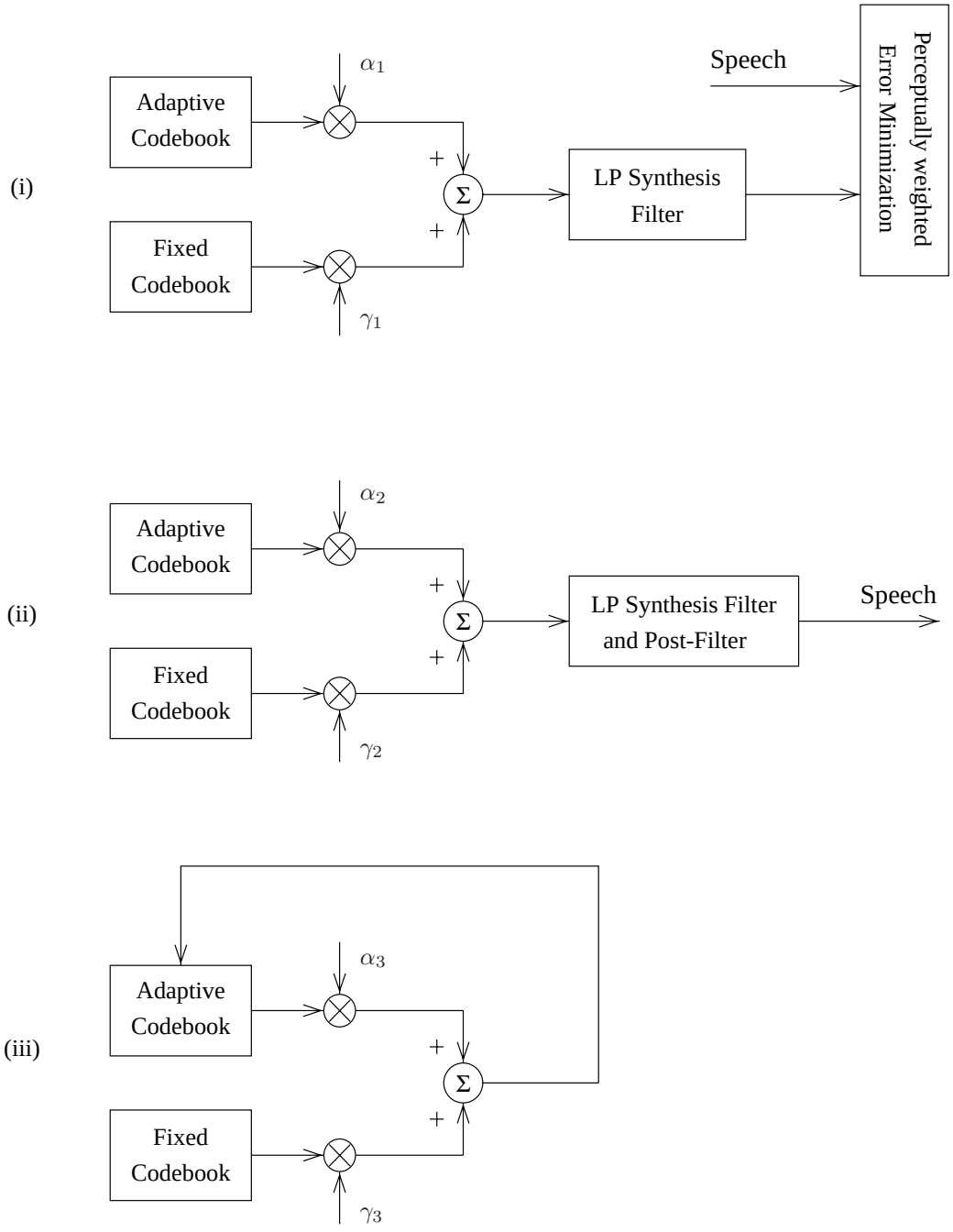


Fig. 2.6 The three stages of coding the LP excitation in a CELP coder: (i) selection of the codebook entries and calculation of the gains, (ii) synthesis of the coded signal, (iii) update of the adaptive codebook.

Constrained Excitation

In the constrained excitation technique (Shoham 1991), the fixed codebook contribution gain in stages (ii) and (iii) is different from the gain calculated in stage (i). The gain of the fixed codebook contribution used in (ii) and (iii) is reduced if the contribution of the adaptive codebook is determined to be large. Decreasing the gain of the fixed codebook component enhances the periodicity of the synthesized signal. Also, the adaptive codebook is updated with a signal which contains a smaller noise component. The method modifies only the fixed codebook gain in stages (ii) and (iii).

Pitch Synchronous Innovation

In the pitch synchronous innovation technique (Mano *et al.* 1995), the fixed codebook contribution is made more periodic based on the estimated pitch period. If the pitch period is shorter than the vectors in the fixed codebook, the entries of the codebook are modified. The new entries are formed by repeating pitch-period-sized blocks which are part of the old entries. The method increases the periodicity of the synthesized speech for pitch values shorter than the length of the fixed codebook vectors. The same fixed codebook change is applied in all three stages. The gains of the codebook contributions in stages (ii) and (iii) are not changed – they are as determined in stage (i).

Comb Filtering

In comb filtering (Wang and Gersho 1990), an extra filter is inserted after the summation of the adaptive and the fixed codebook contributions. The extra filter is used in all three stages. The filter proposed has the form

$$H(z) = (1 - \eta)\frac{1 + \gamma z^{-p}}{1 + \lambda z^{-p}} \tag{2.5}$$

with $\eta = 0.2$, $\gamma = 0.6$, and $\lambda = 0.001F_0$ where F_0 is the estimated fundamental frequency and p is the determined pitch period. The filter is designed to suppresses the noise between pitch harmonics. As in the harmonic noise weighting and the pitch synchronous innovation, the gains of the codebook contributions used in stages (ii) and (iii) are as calculated in stage (i).

Pitch Sharpening

In *pitch sharpening* (Taniguchi *et al.* 1991), the update procedure of the adaptive codebook is modified. The suggested modifications include reducing the fixed codebook gain and center-clipping the adaptive codebook vectors. The control over the signal periodicity is exercised only via changing the update of the adaptive codebook. The signal fed back to the adaptive codebook is different from the LP excitation used for synthesizing the reconstructed speech. The method modifies stage (iii) only.

2.3.3 Generalized Analysis-by-Synthesis

The generalized analysis-by-synthesis coding leads to reduction of the number of bits required for encoding the pitch information (Kleijn *et al.* 1992, 1994). In the generalized LPAS the original speech signal is time-scale modified to facilitate infrequent pitch updates. The modifications should be such that no perceptual distortion is introduced. The update rate of the pitch information is typically reduced from 200 to 50 times per second and the intermediate pitch values are obtained by interpolation. The amount of bits needed for coding the pitch is cut by the factor of four. A number of Relaxed-CELP (RCELP) coders have been implemented based on the generalized LPAS paradigm (Kleijn *et al.* 1993, 1994, Nahumi and Kleijn 1995).

2.4 Modelling the LP Excitation with Pitch-Synchronous Analysis

Traditionally (e.g., in CELP) the LP excitation analysis has been carried out on a fixed-rate, fixed-analysis-block-length basis. In this approach the analysis-block boundaries are asynchronously imposed on the signal and some of the important features (pitch pulses) could be split into two separate blocks.

In the case of speech produced by an idealized model, the voiced excitation is formed by a series of pitch (glottal) pulses which can be seen as separate entities. The analysis in this case should be pitch synchronous preferably in both the rate and the analysis block length (Fig. 2.7). One of the advantages of the pitch-synchronous analysis is that the periodicity of the signal can be controlled by moving individual pitch pulses.

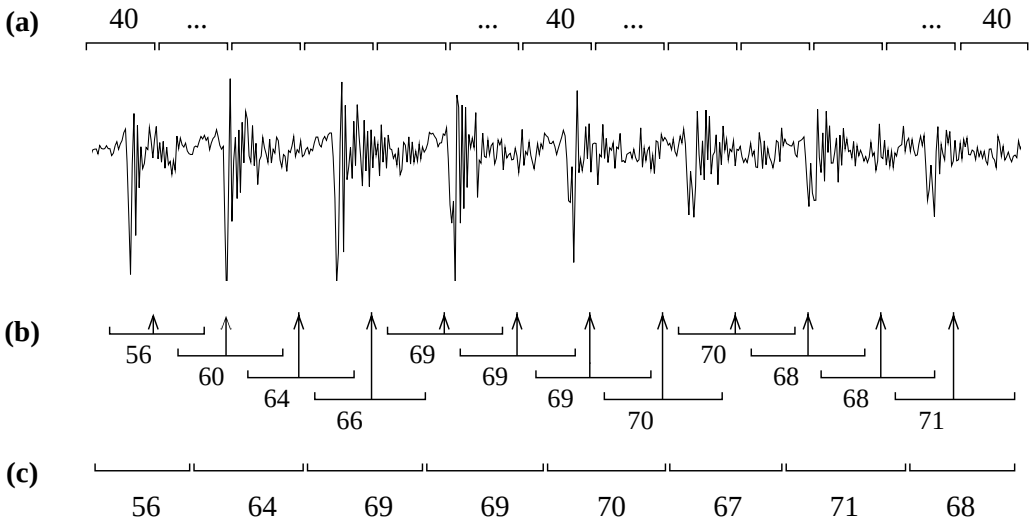


Fig. 2.7 Different types of the LP residual analysis. The numbers indicate the length of analysis blocks. (a) Fixed-rate, fixed-block-length analysis. (b) Fixed-rate, pitch-synchronous block-length analysis. (c) Pitch-synchronous-rate and pitch-synchronous block-length analysis.

2.4.1 Glottal Coding

In a group of LP based coders called the glottal coders the pitch (glottal) pulses are modelled directly by a fixed number of parameters (Hedelin 1986, Fujisaki and Ljungqvist 1986, Krishnamurthy 1992, Childers and Hu 1994). The pulses are identified and the parameters are estimated in an "open loop" fashion from the identified pulses. The coding is based on individual pitch pulses and the analysis-by-synthesis procedure is not used.

The glottal coders require a reliable detection of the boundaries of the glottal pulses. A number of algorithms estimating the instance of the glottal closure have been proposed (Cheng and O'Shaughnessy 1989, Ma *et al.* 1994, Smits and Yegnanarayana 1995) but they are often not very reliable, particularly for noisy speech.

The glottal coders, in general, lack an adequate mechanism to represent the pitch pulse parameters in a way which maintains the perceptually "correct" periodicity of the coded speech. The noise-like component of the LP residual, for example, is modelled only for the unvoiced regions.

2.4.2 Waveform Interpolation Coding

Waveform Interpolation (WI) coding also models individual pitch pulse waveforms. The analysis is pitch synchronous in block-length and the analysis-rate is fixed (see Fig. 2.7b). In the originally proposed Prototype Waveform Interpolation PWI (Kleijn 1991, 1993), waveforms of relatively distant pitch pulses are extracted and intermediate pulses are interpolated from these prototypes. This approach did not fully utilize the actual intermediate pulses to identify appropriate prototypes. Special measures were taken to control the periodicity level of the coded speech and approximate time synchrony between the original and the reconstructed signal was maintained. In the Time-Frequency Interpolation (TFI) method (Shoham 1993b,a) the analysis is performed more often to improve tracking of the inter-pitch variations. The higher-rate analysis results in a better overall quality of the reconstructed speech. In the proposed improvements to the original PWI technique the prototype waveforms, called in WI the characteristic waveforms, are also extracted with a higher rate. They are additionally filtered to separate their periodic and noise components called the slowly evolving waveform (SEW) and the rapidly evolving waveform (REW). The SEW and the REW are then coded separately (Kleijn and Haagen 1994b, 1995b). The pitch interpolation employed in WI coding is such that the time synchrony between the original and the reconstructed speech is not maintained. We write more about pitch interpolation and time synchrony in Chapter 4.

The WI coder is sometimes classified as LPAS (Gersho 1994). In fact the weighted error measurement used in coding the parameters of the WI model is usually performed with respect to the original speech signal. But the LPAS coders are waveform coders in the sense that with decreasing quantization error the reconstructed signal converges to the original. In a WI coder it is not in general true and hence WI is often put into the class of parametric coders (for which the reconstructed signal does not converge to the original even with decreasing quantization error).

2.4.3 The Pitch Pulse Evolution Model

Observing successive pitch waveshapes one can see an evolution though the waveforms are often obscured by noise components that tend to be different for every pitch pulse. We have developed a Pitch Pulse Evolution (PPE) model (Stachurski and Kabal 1994) to efficiently track the changes of the pitch waveforms. In the model, a canonical

waveshape based on a number of noisy pitch pulses is identified, and the canonical waveform is called the underlying pitch pulse. The observed pulses are coded with respect to the estimated underlying pulse.

The model consists of two parts. The voiced LP excitation is composed of a series of pitch pulses. The pitch pulse waveshapes evolve slowly and they may overlap if the lengths of the pulses are small enough. Superimposed on the pitch waveform is an unpredictable component — the unvoiced, noise-like part of the signal. We do not presuppose any particular shape for the pitch pulses, only that the waveshapes of the pitch pulses have some form of continuity from one instance to another. The periodicity of the reconstructed waveform is controlled by (i) adjusting the level of similarity between the consecutive pitch pulses, (ii) changing the amount of the superimposed noise, (iii) placing pitch pulses at encoded (and calculated) positions.

The pitch pulse analysis in the PPE coder is pitch synchronous in analysis-block-length and analysis-rate (Fig. 2.7c). The analysis of the noise contribution is based on LPAS coding with fixed-block-length, fixed-rate analysis (Fig. 2.7a). The PPE coder maintains a relaxed time synchrony with the original signal. The selection of the optimal noise contribution is performed with analysis-by-synthesis with respect to the time-modified speech signal. In that sense the PPE coding is related to the generalized LPAS. More detailed formulation of the PPE method is given in Chapter 3.

2.5 Summary

In this chapter we have presented the basics of linear prediction analysis and the principles of the LP analysis-by-synthesis coding. A number of analysis methods and representations of the LP excitation have been discussed (including glottal coding which does not use analysis-by-synthesis).

It was pointed out that the LP excitation is often modelled as consisting of two parts: the voice component and the unvoiced, noise-like component. The two components are modelled differently:

1. The voiced component is represented in various techniques as:
   - the output of a pitch filter (single-tap filter, multi-tap filter, interpolation filter to accommodate fractional pitch),

- an entry in an adaptive codebook (which can be created as the output of a pitch filter with some additional filtering, for example as in pitch-sharpening),
- a glottal pulse described by a set of parameters,
- a slowly evolving prototype waveform (SEW).

2. The unvoiced component is represented as an entry in a stochastic codebook (or a sum of entries of several codebooks).

Different coding techniques use various analysis rates and analysis-block lengths. CELP coders perform fixed-rate, fixed-block-length analysis. The WI uses fixed-rate analysis but the analysis-block lengths are pitch synchronous. The glottal coding analysis is pitch synchronous in the rate and the analysis-block length.

In the context of the two-component representation, with fixed-rate analysis, the periodicity level of the reconstructed signal is controlled by (i) adjusting the similarity between the corresponding segments of the voiced component (e.g., pitch sharpening), (ii) changing the waveshapes of the vectors which are used to represent the unvoiced component (e.g., pitch synchronous innovation), (iii) varying the ratio between the voiced component and the unvoiced component (e.g., constraint excitation). When the analysis is pitch synchronous, the periodicity level of the reconstructed signal can also be controlled by adjusting the relative positions of the consecutive pitch pulses which make up the voiced excitation.

We have introduced a general idea behind the new proposed coding model, the Pitch Pulse Evolution (PPE) model. In the PPE model the unvoiced component is superimposed on the voiced part of the excitation. The periodicity of the reconstructed signal is dependent on (i) the similarity level between the consecutive pitch pulses, (ii) the amount of unvoiced component superimposed on the pulses, (iii) the positions of the pitch pulses. The LP residual analysis in the proposed PPE model is pitch synchronous in the rate and the analysis-block length.

We would like to emphasize that the ability of a coder to control the periodicity of the reconstructed speech is not sufficient for good perceptual quality of the coded signal. In the same way as there is no objective measurement for perceptual equivalence between audio signals, there is no measure of the "correct" periodicity of the reconstructed signal. The emphasis is, therefore, shifted from *controlling* the period-

icity to the appropriate *modelling* of the original signal. The PPE method constructs the LP excitation directly using the speech production model (the vocal tract excited with a series of similar pitch pulses) as a guide. In this sense the PPE coding is close to glottal coding. The way the LP residual is analyzed and the LP excitation is synthesized, however, makes the PPE method even more closely related to WI coding.

Chapter 3

The Pitch Pulse Evolution Model

3.1 The PPE Concept

In the LP coding, the voiced LP excitation represents the glottal excitation. The voiced LP excitation signal is composed of glottal pulses which are formed as the air is forced through the vocal folds into the speaker's vocal tract. The glottal pulses are the result of the vibrations of the vocal folds and they are similar from one instance to another.

The characteristics of the glottal excitation are reflected in the LP residual. One can identify individual glottal (pitch) pulses which are alike to each other. Since the pitch pulses are not identical, the LP residual is often described as quasi-periodic.

We have no access to the "clean" pitch pulses of the voiced speech which correspond to the glottal pulses as they emanate from glottis. We recover the pulses from the LP residual, assuming that the LP coefficients model all the remaining components of the speech production system. This assumption, although proven to be adequate in the context of speech coding, is nonetheless inaccurate. Moreover, the LP residual even less resembles the true glottal excitation in the presence of an acoustic background noise. As a result, the observed pitch pulses, denoted as $\boldsymbol{u}$, are contaminated with noise and they may significantly differ from the glottal pulses. We try to estimate the "clean" pitch pulses, written as $\boldsymbol{v}$, from the noisy pitch pulses $\boldsymbol{u}$ obtained from the LP residual. We call the estimated pulses $\boldsymbol{v}$ the underlying pulses because, conceptually, they correspond to the glottal pulses which are at the basis of voiced speech production.

In a vector representation of a pitch pulse, if a pulse lasts for 40 samples, the corresponding pitch pulse vector is forty-dimensional (40-D). To compare pitch pulse vectors of different lengths, we pad the shorter vectors with zeros so that all the vectors have the same dimensionality. The dimensionality of the vectors is then equal to the dimension of the vector corresponding to the longest pitch pulse. The consecutive pitch pulses are alike and the similarity between the pulses translates into a relatively small error between the pitch pulse vectors. This fact has been used in practically every low-rate speech coder. The coding schemes employed for the voiced segments usually take advantage of the small difference between the consecutive pulses. In Fig. 3.1 we show a schematic representation of consecutive pitch pulses, which are portrayed as 2-D vectors.

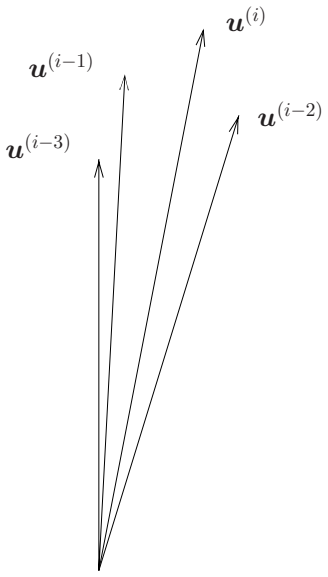
For the unvoiced speech the excitation of the vocal tract is generally random, and the error between consecutive blocks of the unvoiced LP residual is relatively large. A 2-D vector representation of consecutive blocks of unvoiced LP residual is depicted in Fig. 3.2.

The traditional CELP approach to coding the error between consecutive pitch pulses is to code the orthogonal error between them (Fig. 3.3a). In the PPE coder, we first try to estimate the underlying pitch pulse and then code the difference between this calculated pulse and the observed, noisy pulses (Fig. 3.3b).

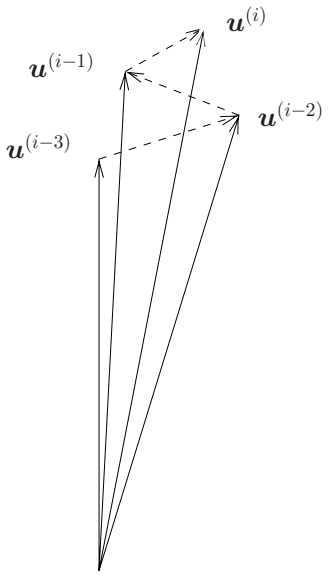
As the vocal folds change their vibration characteristics, the underlying pitch pulse waveshape is not constant. The pitch pulse waveshapes change but they remain similar in their structure. In our methodology we call this change an evolution which led us to the name of our model, the pitch pulse evolution (PPE) model (Fig. 3.4).

In the PPE model we decompose a noisy pitch pulse into the underlying pitch pulse and the superimposed noise (Fig. 3.5a). We estimate the underlying pulse from a noisy observation, which is the LP residual. With a series of the underlying pitch pulses we could determine how the pulses evolve (the drift of the pulses) and then predict the next underlying pulse. The pitch pulses of the LP excitation are created by adding the estimated noise to the estimated underlying pulses (Fig. 3.5b).

We will now develop a general formulation of the PPE method. We adopt a notation in which the vectors obtained in the process of estimation or prediction are marked with the hat $\hat{}$. The tilde $\tilde{}$ marks coded vectors available at both the transmitter and the receiver.

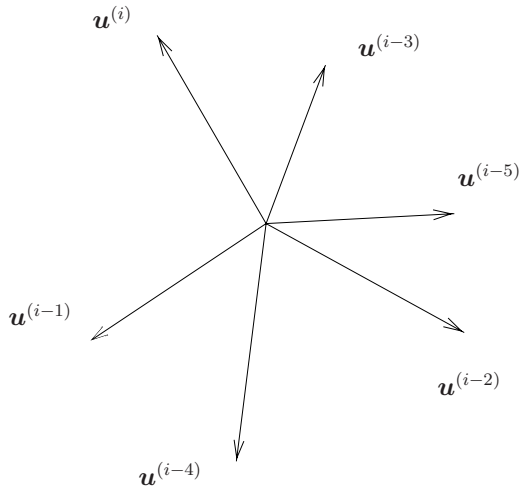


(a) Vector representation of a series of pitch pulses.

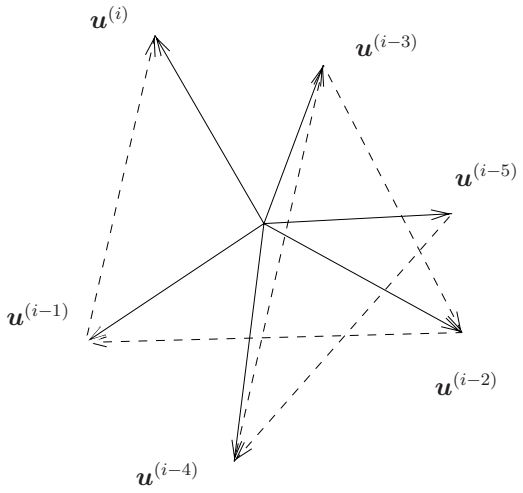


(b) Error between consecutive pitch pulses.

Fig. 3.1 Vector representation of pitch pulses for voiced LP residual.

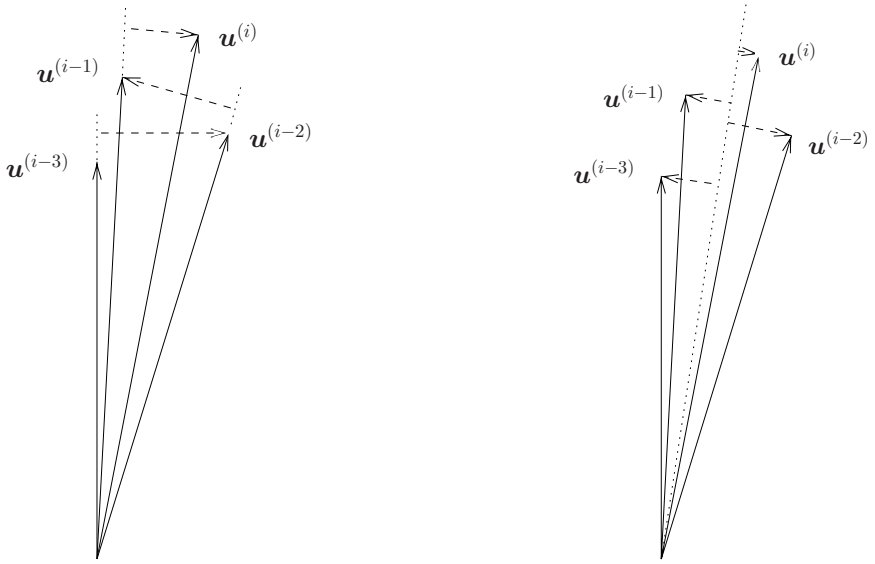


(a) Vector representation of a series of unvoiced blocks of LP residual.



(b) Error between consecutive unvoiced blocks of LP residual.

Fig. 3.2 Vector representation of unvoiced LP residual.



(a) The orthogonal error between consecutive pitch pulses.

(b) The error between the underlying pitch pulse and the observed noisy pulses.

Fig. 3.3 The error between pitch pulses.

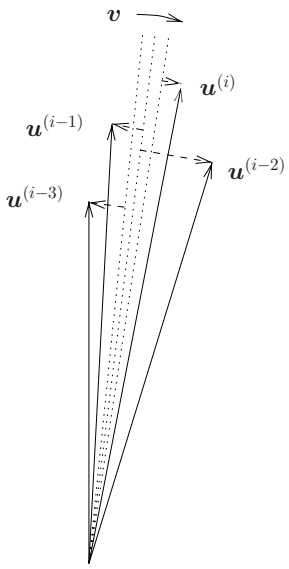
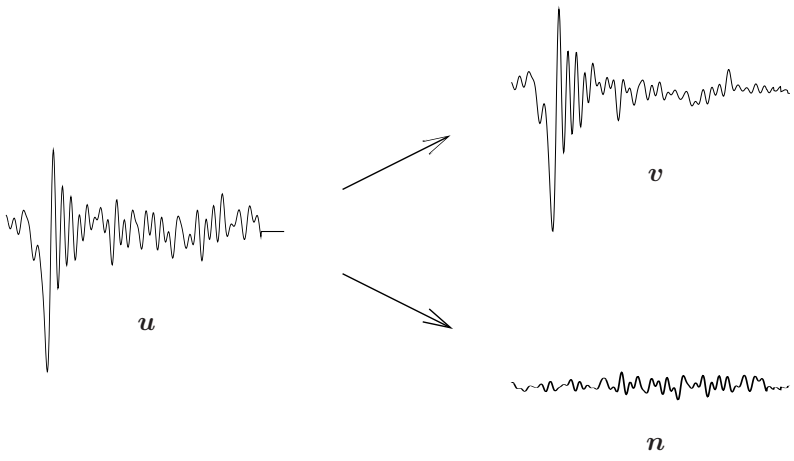
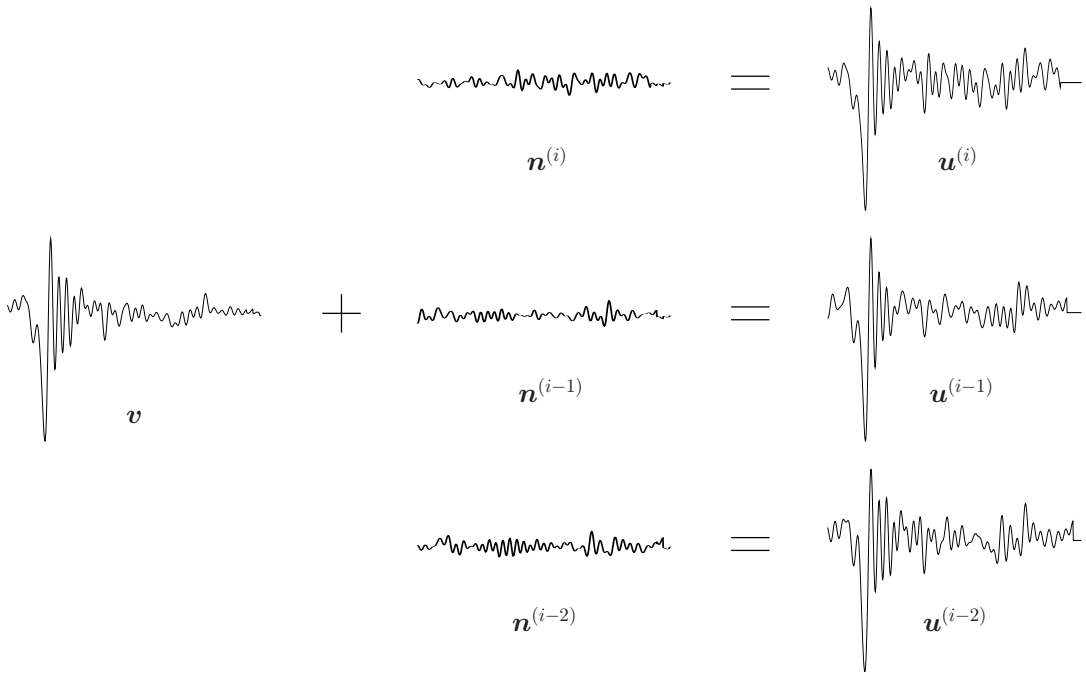


Fig. 3.4 The underlying evolving pitch pulse $\boldsymbol{v}$ and the observed pulses $\boldsymbol{u}$.



(a) Decomposition of a noisy pitch pulse.



(b) Adding noise to the underlying pitch pulse.

Fig. 3.5 The underlying pitch pulse and the noisy pulses.

A vector of the LP residual corresponding to the pitch pulse found at the time instant i is denoted as $\boldsymbol{u}^{(i)}$. The coded equivalent of the vector $\boldsymbol{u}^{(i)}$ is marked as $\tilde{\boldsymbol{u}}^{(i)}$. The past coded pitch vectors corresponding to the slowly evolving pitch pulse shape are written as $\tilde{\boldsymbol{v}}^{(i-1)}$, $\tilde{\boldsymbol{v}}^{(i-2)}$, ...

The new pitch vector can be predicted from the past values of $\tilde{\boldsymbol{v}}$ and $\tilde{\boldsymbol{u}}$ according to some prediction procedure $\mathcal{P}_p$:

$$\hat{\tilde{\boldsymbol{v}}}^{(i)} = \mathcal{P}_p\,(\,\tilde{\boldsymbol{u}}^{(i-1)}, \tilde{\boldsymbol{u}}^{(i-2)}, \ldots,\ \tilde{\boldsymbol{v}}^{(i-1)}, \tilde{\boldsymbol{v}}^{(i-2)}, \ldots\,)\,. \tag{3.1}$$

The same prediction can be performed in both the transmitter and the receiver with the procedure $\mathcal{P}_p$ fixed or adaptive.

The transmitter has access to more information, namely the uncoded versions of vectors $\boldsymbol{u}$ (including the past, present and possibly the future ones to the extent that delay is permissible) and the vectors $\hat{\boldsymbol{v}}$ (unquantized past estimates). It can therefore form a better estimate of the present value of $\boldsymbol{v}$ according to some estimation procedure $\mathcal{P}_e$:

$$\hat{\boldsymbol{v}}^{(i)} = \mathcal{P}_e\,(\,\ldots, \boldsymbol{u}^{(i+1)}, \boldsymbol{u}^{(i)}, \boldsymbol{u}^{(i-1)}, \ldots,\ \hat{\tilde{\boldsymbol{v}}}^{(i)}, \hat{\boldsymbol{v}}^{(i-1)}, \ldots\,)\,. \tag{3.2}$$

Procedure $\mathcal{P}_e$ can also use vectors $\tilde{\boldsymbol{u}}$ and $\tilde{\boldsymbol{v}}$ directly.

We define a vector $\boldsymbol{d}^{(i)}$ which represents the unpredicted drift of the pitch vector $\boldsymbol{v}^{(i)}$, and a vector $\boldsymbol{n}^{(i)}$ which represents the unvoiced part of the vector $\boldsymbol{u}^{(i)}$, so that

$$\boldsymbol{d}^{(i)} = \hat{\boldsymbol{v}}^{(i)} - \hat{\tilde{\boldsymbol{v}}}^{(i)}\,, \tag{3.3}$$

$$\boldsymbol{n}^{(i)} = \boldsymbol{u}^{(i)} - \tilde{\boldsymbol{v}}^{(i)}\,. \tag{3.4}$$

The quantized vectors $\tilde{\boldsymbol{d}}$ and $\tilde{\boldsymbol{n}}$ are used to form the coded underlying pulses $\tilde{\boldsymbol{v}}$ and the coded pulses $\tilde{\boldsymbol{u}}$ which are assembled at the decoder into the LP excitation. We have

$$\tilde{\boldsymbol{v}}^{(i)} = \hat{\tilde{\boldsymbol{v}}}^{(i)} + \tilde{\boldsymbol{d}}^{(i)}\,, \tag{3.5}$$

$$\tilde{\boldsymbol{u}}^{(i)} = \tilde{\boldsymbol{v}}^{(i)} + \tilde{\boldsymbol{n}}^{(i)}\,. \tag{3.6}$$

Note that with this formulation $\boldsymbol{n}^{(i)}$ also accounts for the quantization noise of $\tilde{\boldsymbol{d}}^{(i)}$.

In general, the transmitter performs both operations: $\mathcal{P}_p$ and $\mathcal{P}_e$. The receiver performs the prediction $\mathcal{P}_p$ and based on the transmitted information reconstructs an approximation to the waveform.

Fig. 3.6 summarizes the notation. The diagram schematically displays the predicted, the estimated and the observed vectors. The diagram does not show the "true" underlying pulse $\boldsymbol{v}$ which is unknowable.

$\tilde{}$	coded vectors
$\hat{}$	predicted / estimated vectors
$\boldsymbol{u}$	aligned, pitch length residual vectors
$\boldsymbol{v}$	evolving pitch pulse
$\hat{\hat{\boldsymbol{v}}}$	predicted pitch pulse
$\hat{\boldsymbol{v}}$	estimated pitch pulse
$\boldsymbol{d}$	drift of the pitch vector
$\boldsymbol{n}$	unvoiced part of the excitation

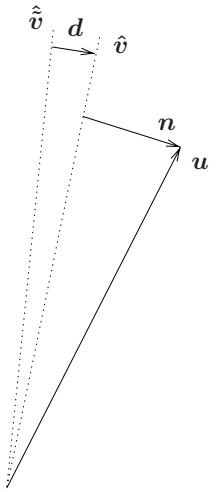


Fig. 3.6 Summary of the notation used in the PPE model.

Our PPE model includes the following features:

1. The approach does not sharply categorize speech into whether it is voiced or unvoiced. The proportion of the two components changes with time as shown schematically in Fig. 3.7. In fact, human speech production does not require voicing to turn off before the unvoiced part of an utterance, and some sounds (e.g. 'z','v') require both voiced and unvoiced forms of excitation. The pitch pulse waveform can be frozen, or adapted very slowly during unvoiced segments. This means that a pitch pulse waveform is available for coding the next voiced region and does not have to be built "from scratch" as in, for example, the CELP coder.

2. We decompose the overall residual waveform into predictable and unpredictable components for separate coding. The predictable part is formed by the underlying pitch pulses and the unpredictable component is formed by the superimposed noise.

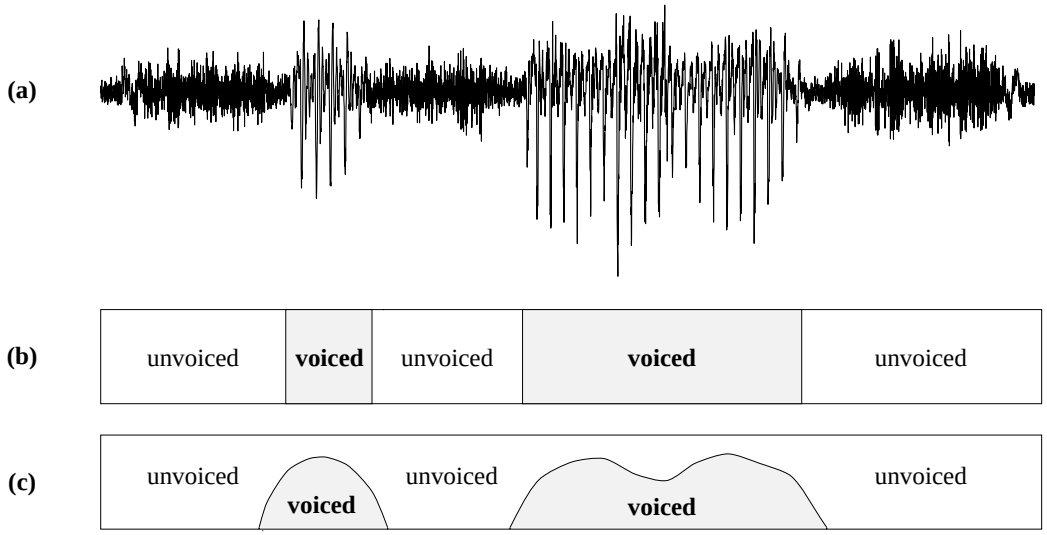


Fig. 3.7 (a) A noisy speech signal. (b) Traditional voiced/unvoiced division. (c) Voiced/unvoiced decomposition in the PPE model.

The transmitter predicts the present underlying pitch pulse waveform based on the past coded LP excitation. It also estimates the current underlying pitch pulse based on the LP residual with possible look ahead. It then transmits (i) the difference between the predicted and the estimated pulse, (ii) the information about the pitch pulse positions, (iii) the unvoiced component of the excitation.

The receiver predicts the present pitch pulse based on the past coded pulses in the same way as the transmitter does. It then forms the current underlying pulse based on the transmitted information about the pulse evolution. The receiver forms the LP excitation combining the coded underlying pulses and the unvoiced component of the excitation.

3. The LP residual is regarded as a series of consecutive pitch pulses. The pitch pulse vectors are extracted from the LP residual in such a way that:

   (i) the combined pitch pulse vectors form the original LP residual,

   (ii) the error between the underlying pitch pulse vectors and the extracted vectors is minimized.

The second condition implies that, with a relatively small drift of the evolving underlying pitch pulse, the extracted pitch pulse vectors are aligned for maximum correlation between each other. The extraction is performed with sub-sample resolution.

For the purpose of estimating the underlying pitch pulses and coding the pulses, the vectors of the extracted pulses are padded with zeros to have the same dimensionality.

4. The estimation of the voiced component of the LP excitation is realized in the process of the estimation of the underlying pitch pulses. We describe the following estimation methods:

   (i) linear filtering of the pulses extracted from the LP residual (filtering with fixed coefficients),

   (ii) maximum ratio combining of the extracted pulses (linear filtering with adaptive coefficients),

   (iii) error minimization between an underlying pitch pulse and a number of the extracted pulses (also linear filtering with adaptive coefficients),

   (iv) an algorithm which minimizes a weighted sum of the errors between
      - a series of the underlying pitch pulses,
      - the underlying and the extracted pulses.

   The underlying pitch pulse estimation can be performed either in the time or in the frequency domain.

5. Pitch interpolation based on separate interpolation of the pitch pulse length and the pitch pulse waveshape is employed to effectively control the periodicity of the LP excitation. The interpolation of the pitch pulse waveshapes is performed on the underlying pitch pulse vectors avoiding interpolation of the noise component.

6. The pulse shape can be decoupled from its final gain-scaled contribution to the LP excitation. Our later formulations are based on the prediction and estimation of normalized signals. Separate quantization of the gain and the pulse shape is used.

The PPE model also allows for a number of new approaches which may further improve performance of a PPE coder. For example, the smooth evolution of the pitch pulses depends on smooth changes in the LP analysis parameters (if different pitch waveforms are processed by different LP filters, unnecessary pulse-to-pulse variations may occur). The LP analysis can be modified to minimize the error between the LP residual and the target pitch pulse waveform. This approach has been investigated already by Zad-Issa and Kabal (1997).

3.2 Extraction of the Pitch Pulses

In the PPE model we view the voiced LP residual as a series of pitch pulses. In general, the pulses in the series overlap so that each pulse is superimposed on the ringing tail of the previous pulses. Every pulse has an initial high energy which falls on top of a small energy signal of the tails of the past pulses. The tails of the previous pulses are buried in a relatively high energy of a new pulse. We consider the tails of the past pulses as part of the noise of the current pulse. We thus regard the LP residual as a series of concatenated pulses in which (except the first and the last pulse) the end of one pulse indicates the beginning of the next pulse.

The extraction of the pulses is equivalent to segmenting the LP residual into pitch pulse vectors of varying length such that:

1. The concatenated pitch pulse vectors form the original residual.
2. The error between the underlying pitch pulses and the extracted vectors is minimized.

An optimal solution is easy to formulate:

1. Segment the LP residual into pitch pulse vectors. Use every possible combination of valid pitch pulse lengths.
2. For every segmentation:
    a) Estimate the underlying, evolving pitch pulses
    b) Calculate the error between the estimated underlying pitch pulse and the segmented-out pitch pulse vectors.

3. Choose the segmentation for which the error calculated in (2) is minimized.

Unfortunately, the solution is as easy to formulate as it is difficult to implement. Firstly, verification of every segmentation of a block of LP residual can be computationally extremely expensive. Secondly, every new sample of the residual could change the past, already determined "best" segmentation.

To bring the computational complexity to an implementable level, two problems should be addressed:

1. How to limit the number of allowable segmentations.

2. How to simplify the estimation of the evolving pitch pulse to reduce its dependency on too many pitch pulse vectors.

Those two problems are related in the sense that the more we limit the number of valid segmentations the more complex estimation of the underlying pitch pulse we can afford with a fixed computational complexity. Also the simpler the estimation of the underlying pitch pulse, the more possible segmentations we can verify.

The specifics of the implementation of the pitch pulse extraction algorithm are left for Chapter 5. Here we will just outline the main ideas used in our implementation which directly deal with the problems presented above.

The number of the valid segmentations is limited by means of the following:

1. The block of the LP residual processed at a time is fixed to a reasonable length.

2. The boundaries of the segments must lie in the specified proximity of the expected beginnings/ends of pitch pulses determined based on the energy of the LP residual.

3. The boundaries of the segments are determined sequentially with a limited inter-dependence between non-adjacent segments.

The estimation of the underlying pitch pulse is simplified. To avoid confusion between the simplified estimation and the estimation described in the next section, we call the underlying pitch pulse obtained with the simplified estimation the model pulse. Conceptually, model pitch pulses correspond to the underlying pitch pulses. The current model pitch pulse is one of the following:

1. The last pitch pulse vector extracted from the LP residual.
2. The next pitch pulse vector (one pulse look-ahead).
3. The average of the last and the next pitch pulse vectors.
4. The average of the past pitch pulse vectors.

The current model pitch pulse is the one of the above four which minimizes the prediction error with respect to the current pitch pulse (current candidate pitch pulse vector).

These limitations and simplifications enabled us to implement a pitch pulse extraction algorithm suitable for our model. The algorithm is discussed in detail in Section 5.3.

3.3 Estimation of the Evolving Pitch Pulse

In Section 3.1 we wrote the consecutive, noisy pitch pulse vectors as $\boldsymbol{u}^{(i)}$ and the corresponding underlying pitch pulses as $\boldsymbol{v}^{(i)}$. In Section 3.2 we outlined the requirements imposed on the pitch pulse extraction procedure in which we identify in the LP residual the noisy pitch pulses. The extraction has been presented as a segmentation of the LP residual into a series of pitch pulses. The pitch pulse vectors $\boldsymbol{u}^{(i)}$ are formed from the pitch pulse segments of the LP residual by padding with zeros, so that all the vectors $\boldsymbol{u}^{(i)}$ have the same dimensionality.

We simplify the notation by dropping the time index i corresponding to the current pulse i. The superscript $^{(i-k)}$ is replaced by a subscript $_k$ and the parameter k is assumed to be in the range $1, \ldots, N$ (the number N is the number of pulses used in the estimation method being described). We have

$$\boldsymbol{u}_k = \boldsymbol{u}^{(i-k)} \quad \text{and} \quad \boldsymbol{v}_k = \boldsymbol{v}^{(i-k)} \,. \tag{3.7}$$

We separate the gain and the shape of each vector,

$$\boldsymbol{u}_k = \mu_k \, \underline{\boldsymbol{u}}_k \,, \quad \boldsymbol{v}_k = \nu_k \, \underline{\boldsymbol{v}}_k \,, \quad \boldsymbol{n}_k = \alpha_k \, \underline{\boldsymbol{n}}_k \,, \tag{3.8}$$

where vectors marked with an underscore are normalized with some normalizing function $\mathcal{N}(\cdot)$. The vectors can be normalized so that (i) the energy of the vector is unity,

(ii) the average energy of the vector elements is unity (the energy of the vector is then equal to the length of the vector), (iii) the maximum energy peak of the vector is equal to unity. We have

$$\underline{\boldsymbol{u}}_k = \mathcal{N}(\boldsymbol{u}_k)\,, \quad \underline{\boldsymbol{v}}_k = \mathcal{N}(\boldsymbol{v}_k) \quad \text{and} \quad \underline{\boldsymbol{n}}_k = \mathcal{N}(\boldsymbol{n}_k)\,. \tag{3.9}$$

We describe four estimation methods: linear filtering, maximum ratio combining, noise error minimization and total error minimization.

3.3.1 Linear Filtering

This is the simplest type of the underlying pulse estimation. Given a set of N vectors $\underline{\boldsymbol{u}}_k$, we calculate $\underline{\hat{\boldsymbol{v}}}$ as a normalized average of the vectors $\underline{\boldsymbol{u}}_k$,

$$\underline{\hat{\boldsymbol{v}}} = \mathcal{N}\Big(\sum_{k=1}^{N} a_k\, \underline{\boldsymbol{u}}_k\Big)\,. \tag{3.10}$$

The filter coefficients a_k are fixed. The estimation is simple but it does not mean that choosing a good set of the linear filter coefficients a_k is easy.

In the filtering proposed for the estimation of the slowly evolving waveforms (SEW) in the WI coding, the coefficients a_k specify a 20 Hz low pass filter (Kleijn and Haagen 1994b). To allow a fast update of the pulses at the voiced onsets (characterized by a relatively large change in the signal energy), the filtering is done on the unnormalized vectors $\boldsymbol{u}_k$, so that

$$\underline{\hat{\boldsymbol{v}}} = \mathcal{N}\Big(\sum_{k=1}^{N} a_k\, \boldsymbol{u}_k\Big)\,. \tag{3.11}$$

In the WI method the pitch pulses are extracted with a fixed rate. Depending on the rate of extraction and the length of the pulses, the pulses may overlap or some samples will not be considered a part of any pulse. There is a constant number of pulses in every filtering operation as the pulses are taken from within a fixed-length time span. In this case specifying a_k as coefficients of a fixed length low-pass filter is reasonable. In the context of the PPE model the pulses are extracted pitch synchronously so that within a constant time span we deal with a variable number of pulses. In this case the linear filter has to have a variable number of taps. A different

set of the filter coefficients would have to be specified depending on the number of pulses filtered.

We found that the linear filtering with fixed coefficients is not flexible enough to control the characteristics of the estimated underlying pulses. The coefficients which performed well on one set of pulses were often inadequate for another set and vice versa. One set of fixed coefficients seems too much of a compromise.

3.3.2 Maximum Ratio Combining

In this section we estimate the underlying pitch pulse from a series of noisy pulses so that the signal-to-noise ratio of the underlying pulse vector is maximized. We assume that

(i) the underlying pitch pulse vector $\underline{\boldsymbol{v}}$ is constant in N consecutive noisy pitch pulses vectors $\boldsymbol{u}_k$,

(ii) each vector $\boldsymbol{u}_k$ is a summation of the underlying pitch pulse and the noise component,

$$\boldsymbol{u}_k = \beta_k \, \underline{\hat{\boldsymbol{v}}} + \alpha_k \, \underline{\boldsymbol{n}}_k \,, \tag{3.12}$$

(iii) the vectors $\underline{\boldsymbol{n}}_1, \ldots, \underline{\boldsymbol{n}}_N$ are orthogonal,

$$\underline{\boldsymbol{n}}_i{}^T \, \underline{\boldsymbol{n}}_j = \begin{cases} 0 & \text{for} \quad i \neq j \,, \\ 1 & \text{for} \quad i = j \,. \end{cases} \tag{3.13}$$

We estimate the pitch pulse $\underline{\hat{\boldsymbol{v}}}$ as a linear combination of $\boldsymbol{u}_k$. We write

$$\beta \, \underline{\hat{\boldsymbol{v}}} + \alpha \, \underline{\boldsymbol{n}} = \sum_{k=1}^{N} a_k \, \boldsymbol{u}_k \,, \tag{3.14}$$

and we want to choose a_k so that the signal-to-noise ratio $\dfrac{\beta^2}{\alpha^2}$ is maximized.

From (3.14) and (3.12) we obtain

$$\beta \, \underline{\hat{\boldsymbol{v}}} + \alpha \, \underline{\boldsymbol{n}} = \Big(\sum_{k=1}^{N} a_k \beta_k \Big) \underline{\hat{\boldsymbol{v}}} \; + \sum_{k=1}^{N} (a_k \, \alpha_k \, \underline{\boldsymbol{n}}_k) \,, \tag{3.15}$$

so

$$\beta = \sum_{k=1}^{N} a_k \, \beta_k \tag{3.16}$$

and

$$\alpha \, \underline{\boldsymbol{n}} = \sum_{k=1}^{N} a_k \, \alpha_k \, \underline{\boldsymbol{n}}_k \, . \tag{3.17}$$

With (3.13) applied to (3.17) we have

$$\alpha^2 = \sum_{k=1}^{N} a_k^2 \, \alpha_k^2 \, . \tag{3.18}$$

Now

$$\frac{\beta^2}{\alpha^2} = \frac{\Big(\sum\limits_{k=1}^{N} a_k \, \beta_k \Big)^2}{\sum\limits_{k=1}^{N} a_k^2 \, \alpha_k^2} \, . \tag{3.19}$$

We use the Schwartz inequality:

$$\Big(\sum_k x_k \, y_k \Big)^2 \leq \sum_k x_k^2 \, \sum_k y_k^2 \, . \tag{3.20}$$

Equality occurs only if the vectors formed by the values x_k and y_k are linearly dependent, i.e., x_k / y_k is constant. We identify

$$x_k = a_k \, \alpha_k \quad \text{and} \quad y_k = \frac{\beta_k}{\alpha_k} \, , \tag{3.21}$$

which gives

$$\Big(\sum_{k=1}^{N} a_k \, \beta_k \Big)^2 \leq \sum_{k=1}^{N} a_k^2 \, \alpha_k^2 \, \sum_{k=1}^{N} \frac{\beta_k^2}{\alpha_k^2} \, . \tag{3.22}$$

From (3.19) and (3.22) we obtain

$$\frac{\beta^2}{\alpha^2} \leq \sum_{k=1}^{N} \frac{\beta_k^2}{\alpha_k^2} \, , \tag{3.23}$$

with equality only if

$$\frac{a_k \, \alpha_k}{\beta_k / \alpha_k} = \text{const} \, , \tag{3.24}$$

which is satisfied for

$$a_k = \frac{\beta_k}{\alpha_k^2}\,. \tag{3.25}$$

We form the approximation of $\underline{\hat{\boldsymbol{v}}}$ such that

$$\underline{\hat{\boldsymbol{v}}} = \mathcal{N}\Big(\sum_{k=1}^{N} \frac{\beta_k}{\alpha_k^2}\,\boldsymbol{u}_k\Big)\,. \tag{3.26}$$

The values β_k and α_k can be calculated as

$$\beta_k = \boldsymbol{u}_k{}^T\underline{\boldsymbol{v}} \quad \text{and} \quad \alpha_k = \sqrt{|\boldsymbol{u}_k|^2 - \beta_k^2} \tag{3.27}$$

with respect to the last estimated underlying pitch pulse vector $\underline{\boldsymbol{v}}$. This is adaptive linear filtering with the coefficients a_k customized for every set of noisy pitch pulse vectors $\boldsymbol{u}_k$.

Although this method seemed very promising at the beginning we soon found that inaccurate estimation of the ratio β_k/α_k^2 may lead to a very poor estimate of the underlying pulse $\underline{\hat{\boldsymbol{v}}}$, especially at times when the underlying pulse changes. Occasionally the error between the estimated pitch pulse and the extracted pitch pulse would be larger than the error between consecutive extracted pulses.

3.3.3 Noise Error Minimization

In this section an estimation based on the minimization of the energy of the noise component is examined. We assume that

(i) the underlying pitch pulse $\underline{\boldsymbol{v}}$ is constant for N consecutive noisy pulses $\boldsymbol{u}_k$,

(ii) each vector $\boldsymbol{u}_k$ is a summation of the underlying pitch pulse and the noise component,

$$\boldsymbol{u}_k = \beta_k\,\underline{\hat{\boldsymbol{v}}} + \alpha_k\,\underline{\boldsymbol{n}}_k\,, \tag{3.28}$$

(iii) the noise vectors $\boldsymbol{n}_1, \ldots, \boldsymbol{n}_N$, are orthogonal to the underlying pitch pulse $\underline{\hat{\boldsymbol{v}}}$,

$$\underline{\boldsymbol{n}}_k{}^T\underline{\hat{\boldsymbol{v}}} = 0 \quad \text{for} \quad k = 1, \ldots, N\,. \tag{3.29}$$

We want to find the underlying pitch pulse $\underline{\hat{\boldsymbol{v}}}$ such that the sum of the noise energies $\sum_k \alpha_k^2$ is minimized.

From (3.28) and (3.29) we have

$$\beta_k = \boldsymbol{u}_k{}^T \underline{\hat{\boldsymbol{v}}}\,, \quad \alpha_k = \boldsymbol{u}_k{}^T \underline{\boldsymbol{n}}_k\,, \tag{3.30}$$

and

$$\boldsymbol{u}_k{}^T \boldsymbol{u}_k = \beta_k \boldsymbol{u}_k{}^T \underline{\hat{\boldsymbol{v}}} + \alpha_k \boldsymbol{u}_k{}^T \underline{\boldsymbol{n}}_k \tag{3.31}$$

$$= \beta_k^2 + \alpha_k^2\,. \tag{3.32}$$

The noise energy is given by

$$\alpha_k^2 = |\boldsymbol{u}_k|^2 - \beta_k^2 \tag{3.33}$$

so that minimization of the sum $\sum_k \alpha_k^2$ is equivalent to maximization of the sum $\sum_k \beta_k^2$.

We write

$$\boldsymbol{U} = [\boldsymbol{u}_1 \cdots \boldsymbol{u}_N] \quad \text{and} \quad \boldsymbol{b} = [\beta_1 \cdots \beta_N]^T\,. \tag{3.34}$$

In this matrix notation, equation (3.29) becomes

$$\boldsymbol{b} = \boldsymbol{U}^T \underline{\hat{\boldsymbol{v}}}\,. \tag{3.35}$$

We want to solve

$$\max_{\|\underline{\hat{\boldsymbol{v}}}\|=1} \|\boldsymbol{U}^T \underline{\hat{\boldsymbol{v}}}\|\,, \tag{3.36}$$

which is the L_2 norm or maximum singular value of $\boldsymbol{U}^T$, σ_1. Vector $\underline{\hat{\boldsymbol{v}}}$ is the first right singular vector of $\boldsymbol{U}^T$ corresponding to the singular value σ_1[†].

We introduce normalization and weighting of the vectors $\boldsymbol{u}_k$. The former deemphasizes vectors with larger energy (they may have a strong noise component), while the latter assigns more importance to the vectors closest to the estimation instance (these vectors may be a better approximation of the current vector $\underline{\hat{\boldsymbol{v}}}$). Now

$$\underline{\hat{\boldsymbol{v}}} = \arg\max_{\|\boldsymbol{v}\|=1} \|\,\boldsymbol{W}\,\underline{\boldsymbol{U}}^T\,\boldsymbol{v}\,\|\,, \tag{3.37}$$

[†]Vector $\underline{\hat{\boldsymbol{v}}}$ is also the eigenvector corresponding to the largest eigenvalue of the matrix $\boldsymbol{U}\boldsymbol{U}^T$.

where matrix $\boldsymbol{W}$ has weighting coefficients on its diagonal and zeros elsewhere.

This estimation method can be seen as linear filtering with adaptive coefficients. In fact, from the Singular Value Decomposition (SVD) theory (Golub and Loan 1989) we have

$$\boldsymbol{U}^T \underline{\boldsymbol{v}} = \sigma_1 \underline{\boldsymbol{z}} \quad \text{and} \quad \boldsymbol{U} \underline{\boldsymbol{z}} = \sigma_1 \underline{\boldsymbol{v}} \,. \tag{3.38}$$

The vectors $\underline{\boldsymbol{z}}$ and $\underline{\boldsymbol{v}}$ are, respectively, the left and the right singular vector (corresponding to the first singular value σ_1) of the matrix $\boldsymbol{U}^T$. Writing $\boldsymbol{a} = \underline{\boldsymbol{z}}/\sigma_1$ we have

$$\underline{\hat{\boldsymbol{v}}} = \boldsymbol{U}\,\boldsymbol{a} \tag{3.39}$$

$$= \sum_{k=1}^{N} a_k \,\boldsymbol{u}_k \,. \tag{3.40}$$

The pitch pulse $\underline{\hat{\boldsymbol{v}}}$ is then a linear combination of vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N$ (or a linear combination of the weighted normalized vectors $w_1\underline{\boldsymbol{u}}_1, \ldots, w_N\underline{\boldsymbol{u}}_N$). The coefficients of the adaptive linear filter a_k are generated, as in Section 3.3.2, for every set of vectors $\boldsymbol{u}_k$.

This method guarantees the smallest error between the underlying pitch pulse and a set of the extracted noisy pulses under the assumption that the underlying pulse is constant within the estimation interval. The assumption of a constant pulse results in a lack of control over the error between the consecutive estimated underlying pitch pulses. We use the insights gained in this section to develop a more general estimation procedure, which is presented in the next section.

3.3.4 Total Error Minimization

In the estimation types described so far, we did not consider the error between the consecutive underlying pitch pulses. We estimated one pitch pulse at a time, assuming, in Section 3.3.2 and Section 3.3.3, that the pulse $\underline{\boldsymbol{v}}$ is constant for N vectors $\underline{\boldsymbol{u}}_1, \ldots, \underline{\boldsymbol{u}}_N$. In this section we estimate a series of underlying pitch pulses while simultaneously controlling the error between them.

For a series of N noisy pitch pulse vectors $\underline{\boldsymbol{u}}_k$ and the corresponding underlying

pitch pulses $\underline{\boldsymbol{v}}_k$, we write the evolution of the pulses as

$$\underline{\boldsymbol{v}}_k = \gamma_k \underline{\boldsymbol{v}}_{k-1} + \sigma_k \underline{\boldsymbol{d}}_k \tag{3.41}$$

$$\text{and} \quad \underline{\boldsymbol{u}}_k = \beta_k \underline{\boldsymbol{v}}_k \quad + \alpha_k \underline{\boldsymbol{n}}_k \,. \tag{3.42}$$

We assume that

$$\underline{\boldsymbol{d}}_k{}^T \underline{\boldsymbol{v}}_{k-1} = 0 \quad \text{for} \quad k = 1, \ldots, N \tag{3.43}$$

$$\text{and} \qquad \underline{\boldsymbol{n}}_k{}^T \underline{\boldsymbol{v}}_k = 0 \quad \text{for} \quad k = 1, \ldots, N \,. \tag{3.44}$$

We have

$$\gamma_k = \underline{\boldsymbol{v}}_k{}^T \underline{\boldsymbol{v}}_{k-1} \,, \quad \sigma_k^2 = 1 - \gamma_k^2 \tag{3.45}$$

$$\text{and} \quad \beta_k = \underline{\boldsymbol{u}}_k{}^T \underline{\boldsymbol{v}}_k \,, \qquad \alpha_k^2 = 1 - \beta_k^2 \,. \tag{3.46}$$

We want to find a set of N vectors $\underline{\hat{\boldsymbol{v}}}_k$ which will minimize the total error

$$e_t = \sum_{k=1}^{N} \Big(\omega \sigma_k^2 + (1-\omega) \alpha_k^2 \Big) \,. \tag{3.47}$$

The weight $\omega \in (0, 1)$ determines the relative importance of the errors $\underline{\boldsymbol{d}}_k$ and $\underline{\boldsymbol{n}}_k$.

Starting with a set of N vectors $\underline{\boldsymbol{v}}_k$ written as $\{\underline{\boldsymbol{v}}_k\}$, we refine one vector $\underline{\boldsymbol{v}}$ at a time so that the error e_t is reduced. The influence of a vector $\underline{\boldsymbol{v}}_i$ on the error e_t can be expressed as

$$e(\underline{\boldsymbol{v}}_i) = \omega \sigma_{i-1}^2 + (1-\omega) \alpha_i^2 + \omega \sigma_{i+1}^2 \,, \tag{3.48}$$

with σ_{i-1}^2, α_i^2 and σ_{i+1}^2 calculated as in (3.45) and (3.46).

The vector $\underline{\boldsymbol{v}}_i^{(min)}$ which minimizes $e(\underline{\boldsymbol{v}}_i)$ can be calculated with SVD applied to the weighted vectors $\underline{\boldsymbol{v}}_{i-1}$, $\underline{\boldsymbol{u}}_i$, $\underline{\boldsymbol{v}}_{i+1}$ (compare with Section 3.3.3). Let $\{\underline{\boldsymbol{v}}_k\}_{(i)}$ denote the set $\{\underline{\boldsymbol{v}}_k\}$ in which vector $\underline{\boldsymbol{v}}_i$ is replaced with vector $\underline{\boldsymbol{v}}_i^{(min)}$. Since the error $e(\underline{\boldsymbol{v}}_i^{(min)})$ is smaller than or equal to the error $e(\underline{\boldsymbol{v}}_i)$, the total error of the new set $\{\underline{\boldsymbol{v}}_k\}_{(i)}$ is smaller than or equal to the total error of the set $\{\underline{\boldsymbol{v}}_k\}$.

The iterative method is applied as follows:

1. Start with an initial set of N vectors $\{\underline{\boldsymbol{v}}_k\}^{(0)}$. Calculate the initial error $e_t{}^{(0)}$ and set l, the number of iterations, to 1.

2. If the vector $\underline{\boldsymbol{v}}_0$ is known:

   – Find a new vector $\underline{\boldsymbol{v}}_1{}^{(l)}$ which will minimize the sum of its weighted errors with respect to vectors $\underline{\boldsymbol{v}}_0$, $\underline{\boldsymbol{u}}_1$ and $\underline{\boldsymbol{v}}_2{}^{(l-1)}$.

   Otherwise:

   – Find the vector $\underline{\boldsymbol{v}}_1{}^{(l)}$ which will minimize the sum of its weighted errors with respect to vectors $\underline{\boldsymbol{u}}_1$ and $\underline{\boldsymbol{v}}_2{}^{(l-1)}$.

3. For every $i = 2, \ldots, N-1$, find a new vector $\underline{\boldsymbol{v}}_i{}^{(l)}$ which will minimize the sum of its weighted errors with respect to vectors $\underline{\boldsymbol{v}}_{i-1}{}^{(l)}$, $\underline{\boldsymbol{u}}_i$ and $\underline{\boldsymbol{v}}_{i-1}{}^{(l-1)}$.

4. Find a new vector $\underline{\boldsymbol{v}}_N{}^{(l)}$ which will minimize the sum of its weighted errors with respect to vectors $\underline{\boldsymbol{v}}_{N-1}{}^{(l)}$ and $\underline{\boldsymbol{u}}_N$.

5. Calculate the error $e_t{}^{(l)}$ and compare it with $e_t{}^{(l-1)}$. If the difference is larger than a specified threshold and the number of iterations is smaller than a given maximum, repeat starting with step (2).

6. Set the estimated underlying pitch pulses as:

$$\underline{\hat{\boldsymbol{v}}}_1 = \underline{\boldsymbol{v}}_1{}^{(l)}\,,\ \ldots\,,\ \underline{\hat{\boldsymbol{v}}}_N = \underline{\boldsymbol{v}}_N{}^{(l)}\,. \tag{3.49}$$

When the new vectors $\underline{\boldsymbol{v}}_i{}^{(l)}$ are found using SVD, the above algorithm converges to a set of pulses $\{\underline{\hat{\boldsymbol{v}}}_k\}$ which corresponds to a *fixed point* or *stationary point* with respect to the iteration operation. The underlying pitch pulses $\{\underline{\hat{\boldsymbol{v}}}_k\}$ depend on the initial conditions, i.e., the initial set of pulses $\{\underline{\boldsymbol{v}}_k\}^{(0)}$. We have obtained good results with the vectors $\{\underline{\boldsymbol{v}}_k\}^{(0)}$ set to the noisy pitch pulse vectors $\{\underline{\boldsymbol{u}}_k\}$.

Performance of the Estimation Algorithm

The prediction error between the noisy pulses $\underline{\boldsymbol{u}}_k$ is written as

$$\varepsilon_k^2 = 1 - \left(\underline{\boldsymbol{u}}_{k-1}{}^T \underline{\boldsymbol{u}}_k\right)^2. \tag{3.50}$$

We specify the estimation gain as the log of the ratio between the sum of the error energies ε_k^2 and the sum of the error energies $\sigma_k^2 + \alpha_k^2$,

$$G_E = 10 \log \frac{\sum_{k=2}^{N} \varepsilon_k^2}{\sum_{k=1}^{N} (\sigma_k^2 + \alpha_k^2)} . \tag{3.51}$$

In a coder in which the differences between the consecutive noisy pitch pulses are coded directly (e.g., CELP in its basic configuration), the error with the energy $\sum_{k=2}^{N} \varepsilon_k^2$ is coded. A positive value of G_E indicates an improvement over this basic approach.

An example of the estimation of the underlying pitch pulses is presented in Fig. 3.8 – Fig. 3.11. In the experiment individual pulses of the LP residual are identified (Fig. 3.8), and the estimation algorithm, with different values of the error weight ω, is applied to the noisy pitch pulses (Fig. 3.9 – Fig. 3.11). The total of 22 pulses are identified in the LP residual of the word "figure". The estimation algorithm was performed on the normalized pulses. The initial set of the underlying pulses $\{\underline{\boldsymbol{v}}_k\}^{(0)}$ is set to the extracted normalized pulses of the LP residual.

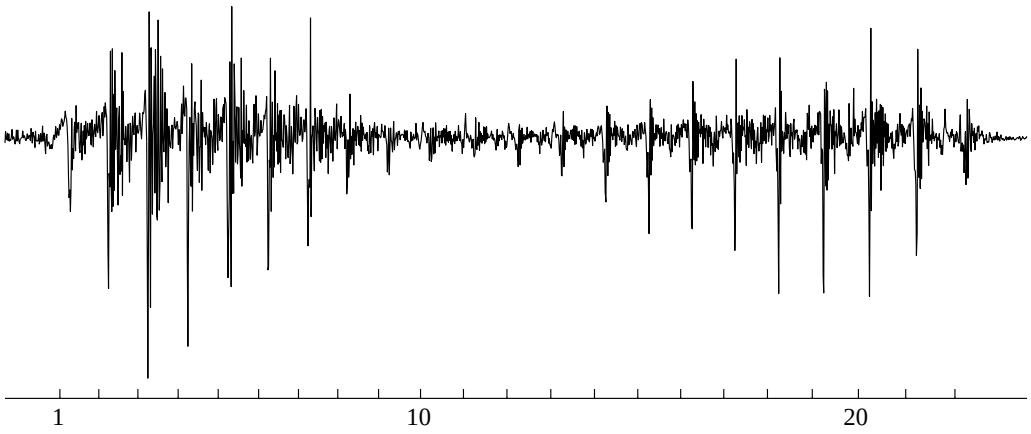


Fig. 3.8 The LP residual of the word "figure" with identified pitch pulses (22 pulses). The normalized, aligned pitch pulses of this residual are used in the example of the underlying pitch pulse estimation.

Fig. 3.9 shows the waveforms and the errors between the waveforms obtained using

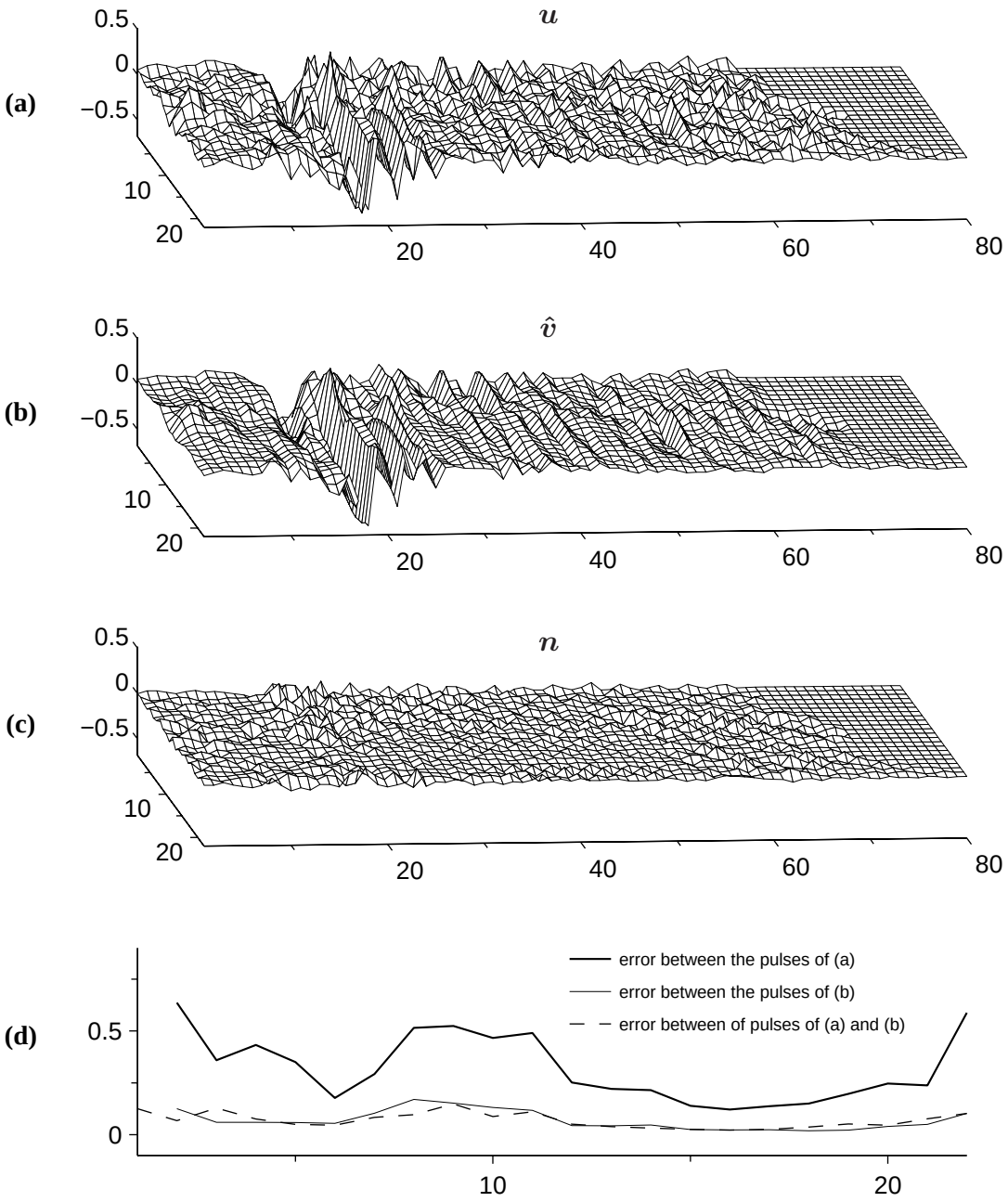


Fig. 3.9 Estimation of the underlying pitch pulses with the error weight $\omega = 0.5$. (a) Pitch pulses extracted from the LP residual (noisy pitch pulses $\boldsymbol{u}$). (b) The estimated underlying pulses $\hat{\boldsymbol{v}}$. (c) The error between the underlying pulses and the noisy pulses $\boldsymbol{n}$. (d) The prediction error between the noisy pulses, the prediction error between the underlying pulses and the orthogonal error between the underlying and the noisy pulses.

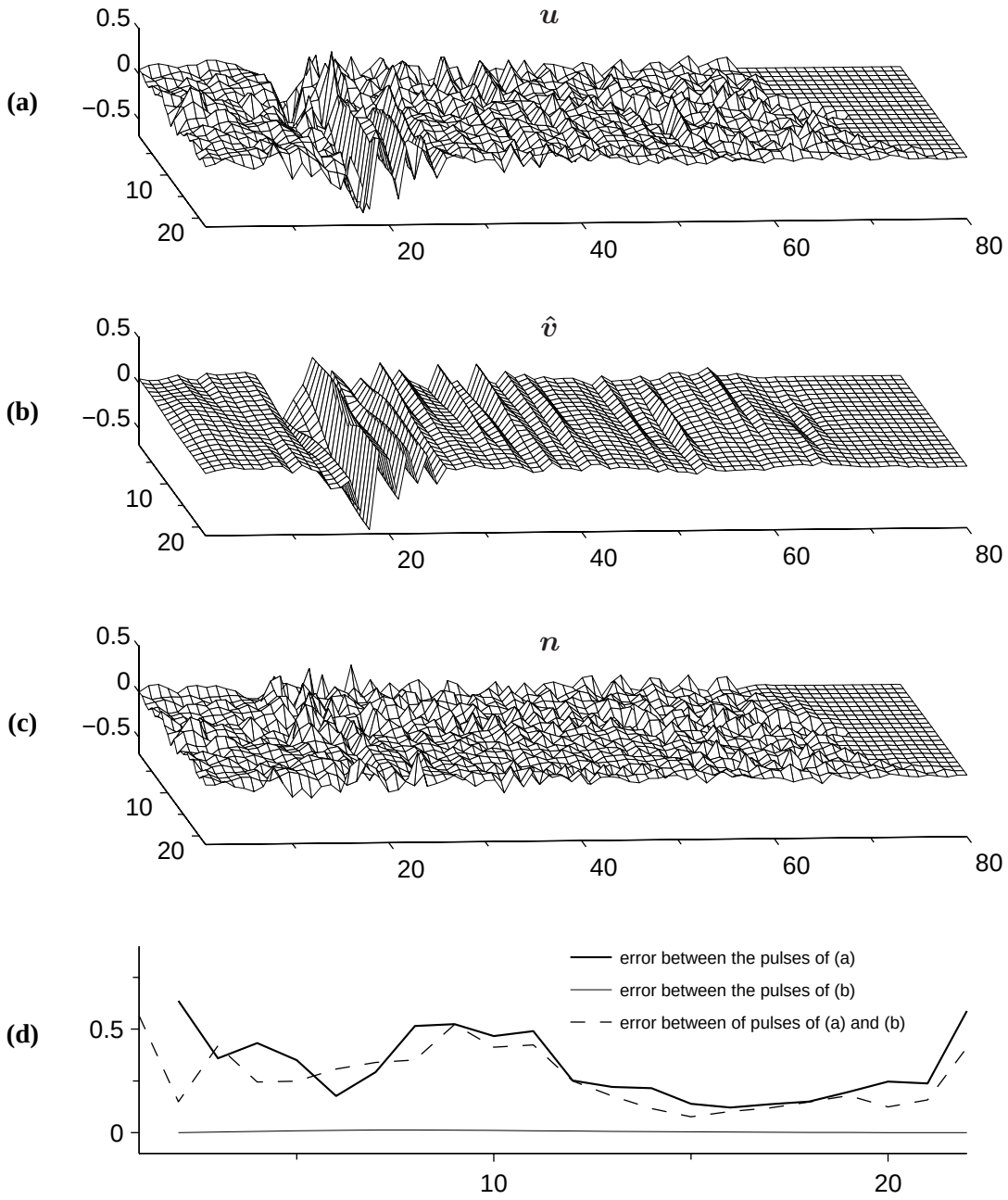


Fig. 3.10 Estimation of the underlying pitch pulses with the error weight $\omega = 0.9$. (a) The noisy pitch pulses of the LP residual. (b) The estimated underlying pulses. (c) The error between the underlying pulses and the noisy pulses. (d) The prediction error between the noisy pulses, the prediction error between the underlying pulses and the orthogonal error between the underlying and the noisy pulses.

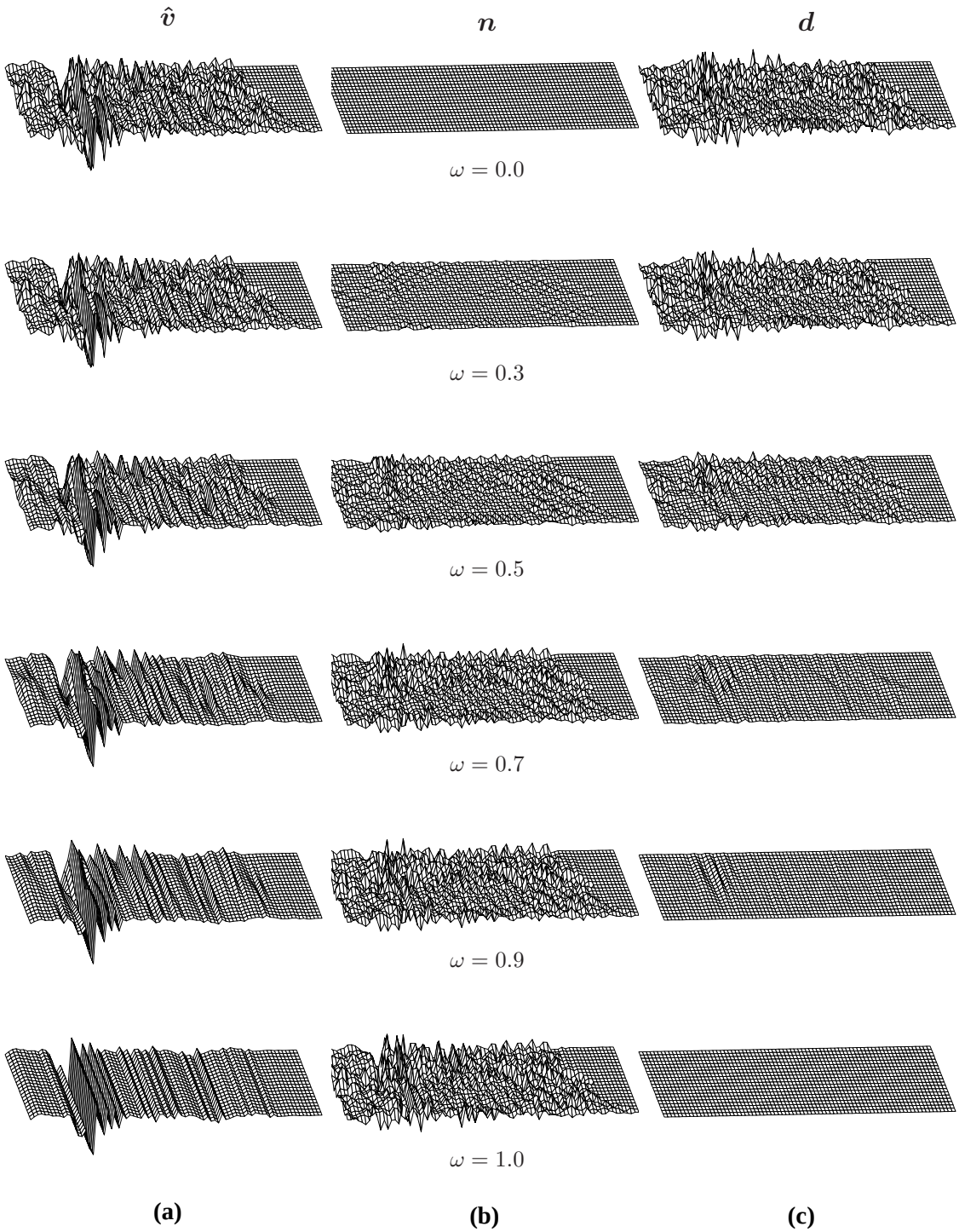


Fig. 3.11 Estimation of the underlying pitch pulses for different values of the error weight ω. (a) The estimated underlying pitch pulses $\hat{\boldsymbol{v}}$. (b) The error between the underlying pulses and the noisy pulses $\boldsymbol{n}$. (c) The error between the consecutive underlying pitch pulses $\boldsymbol{d}$.

the estimation algorithm with the error weight ω equal to 0.5. The decomposition of the noisy pitch pulses into (i) the underlying pitch pulses, (ii) the noise (the error between the underlying and the noisy pulses) is such that (i) the energy of the error between the consecutive underlying pitch pulse vectors, (ii) the energy of the noise vectors, are approximately equal. The estimation gain G_E is equal to 3.51 dB. For $\omega = 0.5$ the estimation gain G_E is maximum because in the formula used for calculating G_E the error energies σ_k^2 and α_k^2 are added with equal weights.

Fig. 3.10 shows the waveforms obtained with the error weight $\omega = 0.9$. The estimated underlying pitch pulses are very similar (the error between the consecutive pulses is almost equal to zero) but the error between the underlying and the noisy pitch pulse vectors is high. In this case the estimation gain G_E is only 0.53 dB.

The shift of the error energy from the error between the noisy pulses onto the error between the underlying pulses is presented in Fig. 3.11. For $\omega = 0.0$ the error between the underlying pulses and the noisy pulses is zero which means that the underlying pulses are equal to the noisy pulses. The error between the consecutive underlying pitch pulses is equal to the error between the consecutive noisy pulses. This is the error which is coded in the basic-configuration CELP. With the increasing ω the consecutive underlying pitch pulses are more and more similar from one to the other, but the error between the underlying pulses and the noisy pulses increases. When $\omega = 1.0$ there is one constant underlying pitch pulse for all the 22 noisy pulses (the single underlying pulse is obtained by applying SVD to the 22 noisy pulses). One can observe a large error between the underlying pulses and the noisy pulses. While for $\omega = 0.9$ the estimation gain G_E is 0.53 dB, for $\omega = 1.0$ the gain G_E is equal to -1.87 dB. It means that the error between the underlying and the noisy pulses is larger than the original error between the noisy pitch pulses.

In general, the smaller the weight ω the smaller is the error between the underlying and the noisy pulses, but the consecutive underlying pulses are less similar from one to the other. The larger the weight ω the more similar are the underlying pulses at the cost of increased error between the underlying and the noisy pulses. For ω in the proximity of 0.5 the estimation algorithm reasonably decomposes the error between the noisy pulses into (i) the error between the underlying pulses (the drift between the pulses), (ii) the error between the underlying and the noisy pulses.

Estimation Using Weighted Average

Using SVD for the error minimization in steps (2)–(4) of the estimation algorithm guarantees a convergence of the estimated underlying pulses to a local minimum. SVD is, however, computationally expensive. With only three vectors involved, given the fact that the vectors are normalized and relatively well correlated with each other, SVD can be approximated with a weighted average of the vectors. Fig. 3.12 shows the underlying pitch pulses obtained with the estimation algorithms which used the SVD and the weighted average error minimization. There is almost no difference between the underlying pulses obtained with the two methods for $\omega = 0.5$. Even for $\omega = 0.1$ and $\omega = 0.9$, the differences between the two sets of the underlying pulses are very small.

The errors, the estimation gains, and the number of iterations required for the convergence of the SVD and the weighted average estimations for different values of the error weight ω are summarized in Table 3.1. In all the cases, the estimation algorithm was stopped when the weighted error e_t specified in (3.47) changed by less than 10^{-6}.

Table 3.1 Comparison between the underlying pitch pulse estimation using the SVD and the weighted average for different values of the error weight ω.

	SVD				Weighted Average			
ω	$\sum_k \sigma_k^2$	$\sum_k \alpha_k^2$	G_E	Iter.	$\sum_k \sigma_k^2$	$\sum_k \alpha_k^2$	G_E	Iter.
0.1	6.54	0.00	0.14	2	4.89	0.11	1.30	3
0.2	5.75	0.03	0.67	2	3.61	0.37	2.30	4
0.3	4.37	0.19	1.70	2	2.69	0.79	2.99	5
0.4	2.77	0.67	2.93	4	2.00	1.10	3.38	5
0.5	1.48	1.53	3.51	6	1.47	1.54	3.51	6
0.6	0.70	2.68	3.01	7	1.05	2.04	3.39	7
0.7	0.36	3.82	2.08	6	0.72	2.64	3.03	10
0.8	0.23	4.78	1.30	7	0.44	3.41	2.44	14
0.9	0.13	5.85	0.53	12	0.21	4.58	1.49	24

When ω is small, the underlying pulses obtained with the weighted average estimation are not as close to the noisy pulses as the underlying pulses obtained with the

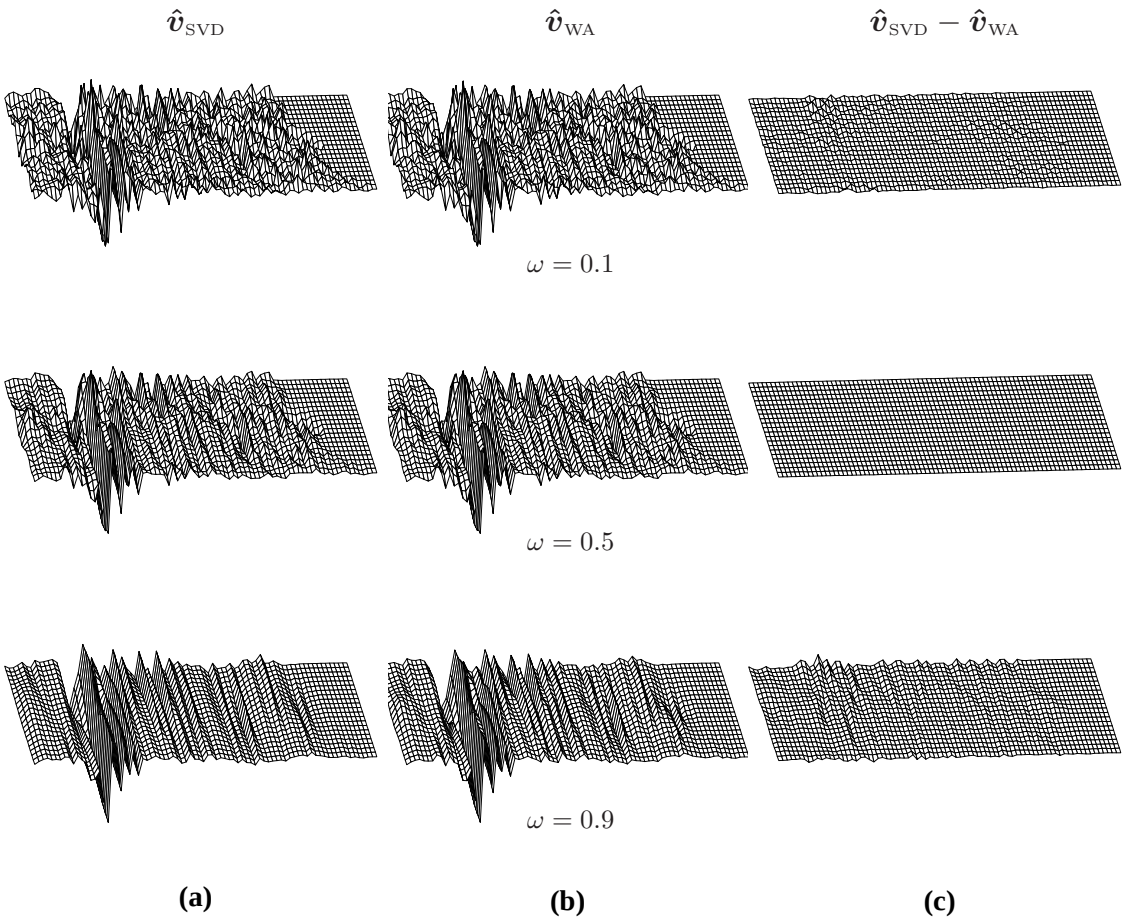


Fig. 3.12 Comparison between the SVD and the weighted average estimation for different values of the error weight ω. (a) The underlying pitch pulses estimated using the SVD method ($\hat{\boldsymbol{v}}_{\mathrm{SVD}}$). (b) The underlying pulses estimated using the weighted average method ($\hat{\boldsymbol{v}}_{\mathrm{WA}}$). (c) The difference between $\hat{\boldsymbol{v}}_{\mathrm{SVD}}$ and $\hat{\boldsymbol{v}}_{\mathrm{WA}}$.

SVD estimation (the error energy $\sum_k \alpha_k^2$ is larger for the weighted average estimation). When ω is large, the weighted average estimation requires more iterations and does not produce the underlying pulses as smoothly evolving as the SVD estimation (the error energy $\sum_k \sigma_k^2$ is smaller for the SVD estimation). The weighted average estimation works, however, effectively when the error weight ω is near 0.5.

3.4 Summary

In this chapter we have formulated the principles of the PPE method. Based on the adopted speech production model, the LP excitation is modelled as a series of underlying pitch pulses buried in noise. The noisy pitch pulses are extracted from the LP residual and the underlying pitch pulses are estimated based on the extracted pulses.

The transmitter predicts the current underlying pulse based on the past coded LP excitation, estimates the current underlying pulse based on the current LP residual, and transmits the difference. The information about the pitch pulse positions and the unvoiced component of the LP excitation is also transmitted. The receiver predicts the current underlying pulse in the same way as the transmitter does, and based on the transmitted information about the pulse evolution, refines this prediction. The LP excitation is formed by placing the created pulses at the coded pulse positions and adding the unvoiced component.

We have outlined the requirements imposed on the pitch pulse extraction algorithm. The LP excitation is regarded as a series of consecutive pitch pulses so that the concatenation of the extracted pitch pulses should form the original LP residual. The straightforward formulation of the problem of the pitch pulse extraction leads to a computationally very expensive system; we have indicated directions for reducing the computational burden to implementable levels.

The pitch pulse extraction is one of the most important parts of the PPE system on which the rest of the coder performance is dependent. The extraction procedure is especially designed to fit the adopted speech production model and it performs a number of operations which are distinct in other coders (i.e., the WI coder). The PPE pitch pulse extraction combines (i) pitch period estimation, (ii) pitch pulse extraction, (iii) alignment of the extracted pitch pulses. The formulation of the pitch pulse extraction further includes simple estimation of the underlying pitch pulse which

corresponds to the SEW/REW separation used in the WI coding. Since the pulse extraction is of central importance, it has been given particular attention in our work. A practical implementation of the pitch pulse extractor is presented in Section 5.3.

Various methods for estimating the underlying pitch pulses have been presented. In particular we have examined linear filtering with fixed coefficients, maximum ratio combining, noise error minimization and total error minimization. The last technique provides the most flexible framework in which the properties of the calculated pitch pulses can be directly influenced by a single parameter, the error weight.

We have developed an iterative algorithm as a practical way of finding a solution to the non-linear optimization problem which results from the total error minimization approach. The performance of the algorithm has been illustrated and the SVD and the weighted average versions of the algorithm have been compared. We have shown that the weighted average estimation, while computationally much less expensive, produces similar results to the SVD estimation.

Chapter 4

Interpolation of the Pitch Pulses

Interpolation of the pitch involves creating intermediate pitch pulses between given points in the waveform. A pitch pulse is characterized by (i) the pitch pulse length, and (ii) the waveshape which forms the pulse. Pitch interpolation techniques include those used in Sinusoidal Transform Coding (STC) (McAulay *et al.* 1991, Brandstein *et al.* 1991, McAulay and Quatieri 1995), Waveform Interpolation (WI) coding (Kleijn and Granzow 1991, Kleijn 1993, Kleijn and Haagen 1995b), and Relaxed-CELP (RCELP) (Kleijn *et al.* 1994). In all three methods (STC, WI and RCELP), the interpolation of the pulse length and the interpolation of the pulse waveshape are part of one interpolation procedure.

In the PPE model we decouple the interpolation of the pitch pulse length and the interpolation of the pulse waveshape to gain a greater control over the evolution of the pulse characteristics. We argue that such an approach is justifiable from the point of view of the speech production model. Our experiments and observations of the LP residual indicate that indeed the waveshape of a pitch pulse is largely independent of the pitch pulse length and, therefore, should be considered separately.

In this chapter we examine how the interpolation techniques used in other coders influence the pitch pulse length and the waveshapes of the interpolated pulses. We describe the PPE interpolation of the pitch pulse lengths in which the waveshapes of the pitch pulses are not changed. Finally, spectral interpolation used in STC and WI are presented. We adopt the spectral interpolation to modify the waveshapes of the pitch pulses without changing the interpolated pitch pulse lengths.

4.1 Pitch Pulse-Length Interpolation

4.1.1 Periodic and Quasi-Periodic Signals

A periodic signal is composed of a number of waveforms which have the same length and identical shapes. The period of the signal $p(t)$ is constant and is equal to the length of one waveform. The phase of the signal $\phi(t)$ increases linearly in time and changes by 2π within one period.

A quasi-periodic signal is composed of a number of waveforms which may not have the same length and which are only alike in shapes. We can describe the periodicity of a quasi-periodic signal using the phase of the signal, $\phi(t)$. We define the following:

(i) the instantaneous period $p(t)$ which reflects the local variations of the phase,

$$\frac{1}{p(t)} = \frac{1}{2\pi}\frac{d\phi(t)}{dt}\,, \tag{4.1}$$

(ii) the interval $P(t)$ which is the time period over which, starting at t, the phase increases by 2π,

$$P(t) = \min_{\phi(\tau)-\phi(t)=2\pi} \tau - t\,. \tag{4.2}$$

Given the initial phase $\phi(t_0)$, the instantaneous period $p(t)$ determines the phase $\phi(t)$ as

$$\phi(t) = \phi(t_0) + 2\pi \int_{t_0}^{t} \frac{1}{p(t)} dt\,. \tag{4.3}$$

The interval $P(t)$ can be calculated from the instantaneous period $p(t)$ by solving the equation:

$$\int_{t}^{t+P(t)} \frac{1}{p(t)} dt = 1\,. \tag{4.4}$$

In a quasi-periodic signal the interval $P(t)$ corresponds to the time which separates two consecutive similar-shape features of the concatenated waveforms. For a periodic signal the interval $P(t)$ and the instantaneous period $p(t)$ are equivalent and they are equal to the constant period of the signal. For a quasi-periodic signal the interval $P(t)$ and the instantaneous period $p(t)$ are, in general, different.

The LP Residual

For voiced speech, the quasi-periodic LP residual is composed of a series of pitch pulses of varying lengths and shapes. We mark the beginnings of two adjacent pitch pulses as t_i and t_{i+1}. We specify the time interval $P(t_i)$ as

$$P(t_i) = t_{i+1} - t_i \tag{4.5}$$

so that the interval $P(t_i)$ is equal to the length of the pitch pulse beginning at time t_i. Pitch interpolation results in the change of the pitch pulse positions t_i and t_{i+1} into $\tilde{t}_i$ and $\tilde{t}_{i+1}$ respectively. The pitch-interpolated pulse begins at time $\tilde{t}_i$ and is of length $P_I(\tilde{t}_i)$,

$$P_I(\tilde{t}_i) = \tilde{t}_{i+1} - \tilde{t}_i \,. \tag{4.6}$$

Various types of pitch interpolations transform the waveshape within the interval $P(t_i)$ into the interval $P_I(\tilde{t}_i)$ in different ways. We examine the pitch interpolations in the light of this transformation.

4.1.2 Pitch Interpolation in Existing Coders

Waveform Interpolation Coder

In the context of WI the pitch period is viewed as the instantaneous period $p(t)$. The period $p(t)$ is estimated at regular time intervals, coded and transmitted. The interpolated instantaneous period $p_I(t)$ is created by linear interpolation of the transmitted values of $p(t)$. The LP excitation is formed based on the interpolated period $p_I(t)$ which determines, given the initial phase $\phi_{I0}(t)$, the evolution of $\phi_I(t)$.

Consider a pitch pulse which starts at time t_i and whose length is $P(t_i)$. In WI the instantaneous pitch period $p(t)$ is estimated without explicit concern with the evolution of $\phi(t)$. In particular, even if the instantaneous period $p(t)$ is estimated for every time t, the phase $\phi(t)$ determined from the estimated $p(t)$ may not change by 2π within the time interval $P(t)$.

The instantaneous period $p_I(t)$ is formed by interpolating the transmitted values of $p(t)$. The interpolated period $p_I(t)$ determines the phase $\phi_I(t)$, but the employed interpolation of $p_I(t)$ (linear interpolation of the coded $p(t_n)$) provides no explicit constraint on the evolution of $\phi_I(t)$ (which determines the pitch-interpolated pulse

length $P_I(\tilde{t}_i)$). The difference between $P_I(\tilde{t}_i)$ and $P(t_i)$ results in the loss of time synchrony between the original and the reconstructed signal and it may accumulate over a number of pulses. Eventually, addition or deletion of pitch pulses may occur.

In WI, pitch pulse waveshapes are length (phase) normalized, coded and transmitted. The signal is reconstructed by applying the interpolated phase $\phi_I(t)$ to the phase-normalized waveshapes. The evolution of the phase $\phi_I(t)$ is continuous and in effect the original pitch pulse of length $P(t_i)$ is time-warped to fit the pitch-interpolated pulse length $P_I(\tilde{t}_i)$. The time warping is determined by the implicit evolution of the phase $\phi_I(t)$.

Sinusoidal Transform Coding

In STC, a set of frequencies $f_k(t)$ and their phases $\phi_k(t)$ are identified. In general, the set $\{f_k(t)\}$ may include any frequencies but, in low bit rate coders, it often consists of the harmonics of the fundamental frequency $f(t)$. The fundamental frequency $f(t)$ is the inverse of the instantaneous period $p(t)$,

$$f(t) = \frac{1}{p(t)}\,. \tag{4.7}$$

The frequencies $f_k(t)$ and the phases $\phi_k(t)$ are interrelated,

$$f_k(t) = \frac{1}{2\pi}\frac{d\phi_k(t)}{dt} \tag{4.8}$$

and

$$\phi_k(t) = \phi_k(t_0) + 2\pi \int_{t_0}^{t} f_k(t)dt\,. \tag{4.9}$$

The frequencies $f_k(t)$ corresponding to each analysis frame are estimated, coded and transmitted. The interpolation of the coded frequencies $f_k(t_n)$ is quadratic. Since there are many possible quadratic paths which the interpolated frequencies $f_{k_I}(t)$ can follow between the coded updates, the initial phase $\phi_{k_I}(t_0)$ and the frequencies corresponding to consecutive frames $f_k(t_n)$ are not sufficient to determine the interpolation. By specifying an extra condition for every coded frequency, the piece-wise quadratic evolutions of $f_{k_I}(t)$ can be chosen such that the cubic evolutions of the phases $\phi_{k_I}(t)$ satisfy the imposed boundary conditions. Given the initial phases $\phi_{k_I}(t_0)$, the phases $\phi_{k_I}(t)$ determine the reconstructed signal. By imposing the boundary conditions

on $\phi_{k_I}(t)$, the time synchrony between the original and the reconstructed signals is maintained. The encoding of the information about the boundary condition, however, increases the coder bit rate.

In STC the signal is reconstructed by the summation of the waveforms obtained by applying the interpolated phases $\phi_{kI}(t)$ to a set of sinusoids. Although the time synchrony between the original and the reconstructed signals is retained, the lengths of individual pitch pulses before the interpolation $P(t_i)$ and after the interpolation $P_I(\tilde{t}_i)$ may differ. During the reconstruction a pitch pulse of length $P(t_i)$ is in effect time-warped to fit the interpolated pulse length $P_I(\tilde{t}_i)$. The warping is determined by the evolution of the phases $\phi_{kI}(t)$.

Relaxed-CELP

In a CELP coder with an adaptive codebook, the LP excitation is formed as a sum of the adaptive codebook entry and a fixed codebook entry. First the adaptive codebook is created from the past LP excitation. Then the adaptive codebook contribution and the fixed codebook contribution are selected based on a weighted error between the original and the reconstructed speech. In a generalized analysis-by-synthesis procedure, the original speech is time-scale modified and the weighted error is calculated with respect to the modified signal.

The Relaxed-CELP (RCELP) coders use the generalized analysis-by-synthesis approach. For every frame, the RCELP coder estimates and encodes the instantaneous period $p(t_n)$. The interpolated period $p_I(t)$ is formed by linear interpolation of the transmitted $p(t_n)$. The adaptive codebook contribution is formed, at the transmitter and at the receiver, by time warping the past LP excitation. The time warping is such that the instantaneous period of the created signal matches, in the current frame, the interpolated period $p_I(t)$. The modified speech signal used in the generalized analysis-by-synthesis procedure is obtained from the time-scale modified LP residual. In the time-scale modifications, blocks of the original LP residual are time shifted.

The adaptive codebook contribution is used as the reference vector with respect to which the time-shifts of the LP residual are optimized (the minimized perceptually-weighted error is calculated in the speech domain). In effect the time warping is used to create the reference vector for the shifting procedure but is not used to actually modify the LP residual. By limiting the maximum allowable accumulated shift, the time synchrony between the original and the modified signals is maintained.

The fixed codebook contribution is chosen based on the weighted error between the reconstructed speech and the speech obtained from the modified LP residual. There is an inconsistency in this procedure: the adaptive codebook contribution is formed by time warping and the fixed-codebook target vector is based on the LP residual modified with time shifting[†]. For best results (i.e., for a best match between the adaptive codebook pulses and the modified LP residual pulses) both signals should be created with the same procedure: either time warping or time shifting.

In the description of RCELP, the use of time shifting is presented as a computational saving over more computationally expensive time warping. It is implied that, computational complexity aside, time warping is the preferable method of forming the modified LP residual. It was reported that a system with the time-warped LP residual had a significant increase in coding efficiency (Kleijn *et al.* 1993). Also time shifting could be applied to both, the LP residual used to form the modified speech and the past LP excitation used to create the adaptive codebook. We do not know of such a system having been tested.

4.1.3 Is Time Warping Justified?

Time warping results in "continuous evolution" of the waveforms but it changes the "internal structure" of a pulse by time stretching or contracting the pulse. Is it the right thing to do?

We have carried out a number of experiments to see if time warping is justifiable based on what happens to the original LP residual waveform in the case of rapidly changing pitch. We recorded a vowel "a" spoken with a quickly rising pitch. We then examined the behaviour of consecutive pitch pulses and tried to determine the relation between waveshapes of pulses with different lengths. Given two pulses, occurring in the same voiced region but one being considerably shorter that the other, we wanted to determine if the longer pulse could be formed by "stretching" the shorter pulse. In this case one pulse would be a time-warped version of the other. Or, if the longer pulse could be formed by letting the shorter pulse "ring" longer. We did not try to predict this "ringing" and we simply padded the shorter pulse with zeros to the length of the longer pulse.

[†]In the context of a time window which contains few pitch pulses time shifting can be viewed as a form of a discretized time warping. We are interested in what is happening within one pitch pulse, within the time interval $P(t_i)$, and in this context time warping and time shifting are distinct.

Our observations show that a longer pulse can, in fact, be much better approximated by the "ringing" version of the shorter pulse and not by its "stretched" version. Fig. 4.1 depicts a series of pitch pulses extracted from the LP residual of the recorded vowel "a". Between the pulses plotted in Fig. 4.1a and the pulses plotted in Fig. 4.1b, the LP residual contained only four extra pitch pulses (not shown on the figure). The pulses of Fig. 4.1a and Fig. 4.1c are identical. The pulses of Fig. 4.1d are created by padding the pulses of Fig. 4.1b with zeros. The pulses of Fig. 4.1e are formed by time warping (which in this case is equivalent to time stretching) the pulses of Fig. 4.1c. We can observe that the pulses padded with zeros (Fig. 4.1d) maintain the alignment with the longer pulses (Fig. 4.1c) and the pulses stretched (Fig. 4.1e) lose this alignment. Based on our experiments, we believe that time warping of the LP residual while changing (interpolating) the pitch is not a proper thing to do. In our pitch interpolation we allow only time shifting of the pitch pulses, effectively changing the length of the pulses but avoiding any time warping within a pulse. Time shifting may introduce signal discontinuities at time-shift boundaries. This, however, is not perceptually noticeable if the discontinuity occurs in a region of low energy, which is the case when the pitch pulses are properly extracted.

4.1.4 Pitch Pulse-Length Interpolation in the PPE Model

In the PPE model we code the average pitch and not the local pitch associated with one time instant. We code the length of a few pitch pulses and our interpolation can be viewed as a segmentation of the encoded length into the correct number of pulses. In this way we preserve the pitch "on average" and maintain the time synchrony of the reconstructed speech with the original.

The pitch information is coded once per frame. For frame k, we encode the position of a pitch pulse τ_k and the information about the number of pulses between τ_k and the pulse position coded in the previous frame τ_{k-1}. We write the number of pulses between τ_{k-1} and τ_k as N_k. In general, there are more than one pitch pulse in a frame. The problem of selecting the pulse whose position will be code is deferred to Section 5.4.1.

The reconstructed signal is time-synchronous with the original at least once per frame, at the time instant τ_k. The average pitch pulse length is preserved although the intermediate pitch pulse lengths may be different from those in the original signal.

Fig. 4.1 (a) and (b) Pitch pulses extracted from the LP residual of a vowel "a" pronounced with a quickly rising pitch. (c) The same pulses as in (a). (d) Pulses of (b) padded with zeros. (e) Pulses of (b) stretched.

We refer to the segmentation of a block of $\tau_{k-1} - \tau_k$ samples into N_k pulses as the pitch pulse-length interpolation. In pulse-length interpolation each segment corresponds to an interpolated pitch pulse length $P_I(t_i)$. The segmentation is determined by an interpolation specified in one of the following domains: the interval $P(t)$, the fundamental frequency $f(t)$, or the instantaneous period $p(t)$. Whatever the domain, the interpolated pitch pulse length $P_I(\tilde{t}_i)$ is calculated and the signal is reconstructed based on the relation between $P(t_i)$ and $P_I(\tilde{t}_i)$. The pitch pulse of length $P_I(\tilde{t}_i)$ is formed from the corresponding pitch pulse of length $P(t_i)$ either by truncating the original pulse (if $P(t_i)$ is longer) or by adding to the original pulse a suitable extension (if $P(t_i)$ is shorter). There is no time warping.

The description of the interpolations in the different domains follows.

Interpolation in the Pitch Pulse Length Domain $P(t)$

The encoded position of frame k–1 marks the beginning of the first pulse of the region coded in frame k,

$$\tilde{t}_0 = \tau_{k-1}\,, \tag{4.10}$$

and the coded position of the frame k marks the end of the last pulse in the region,

$$\tilde{t}_{N_k} = \tau_k\,. \tag{4.11}$$

We require that

$$\sum_{j=0}^{N_k-1} P_I(\tilde{t}_j) = \tilde{t}_{N_k} - \tilde{t}_0 \tag{4.12}$$

where $P_I(\tilde{t}_i)$, …, $P_I(\tilde{t}_{N_k-1})$ are the pitch pulse lengths interpolated in frame k.

Linear interpolation in the $P(t)$ domain is performed as:

$$\left.\begin{aligned} P_I(\tilde{t}_j) &= P_I(\tilde{t}_{j-1}) + d_k \\ \tilde{t}_{j+1} &= \tilde{t}_j + P_I(\tilde{t}_j) \end{aligned}\right\} \quad \text{for} \quad 0 \le j < N_k \tag{4.13}$$

where $P_I(\tilde{t}_{-1})$ is the last pitch pulse length interpolated in the previous frame. We call this interpolation linear because the difference between consecutive pulse lengths

within a frame is constant,

$$d_k = P_I(\tilde{t}_j) - P_I(\tilde{t}_{j-1}) = \text{const}\,, \qquad 0 \le j < N_k\,. \tag{4.14}$$

The difference between the pulse lengths is calculated as

$$d_k = 2\frac{(\tau_k - \tau_{k-1}) - N_k P_I(\tilde{t}_{-1})}{N_k(N_k+1)}\,. \tag{4.15}$$

In the PPE coder described in Chapter 5, we use pulse-length interpolation based on this linear interpolation in $P(t)$.

Interpolation in the Fundamental Frequency Domain $f(t)$

For the interpolation in the $f(t)$ domain, we calculate the interpolated pitch pulse lengths $P_I(\tilde{t}_i)$ from the interpolated fundamental frequency $f_I(t)$.

For the frame k, we have

$$\int_{\tau_{k-1}}^{\tau_k} f_I(t)dt = N_k\,. \tag{4.16}$$

The coded time intervals are written as

$$T_{k-1} = \tau_{k-1} - \tau_{k-2}\,, \tag{4.17}$$

$$T_k = \tau_k - \tau_{k-1}\,. \tag{4.18}$$

$$T_{k+1} = \tau_{k+1} - \tau_k\,. \tag{4.19}$$

Linear Interpolation

The linearly changing frequency $f_I(t)$ can be specified in the frame k as

$$f_I(t) = a_k(t - \tau_{k-1}) + b_k\,, \quad \tau_{k-1} \le t < \tau_k\,. \tag{4.20}$$

The following approaches can be used to calculate the parameters a_k and b_k.

(i) Applying (4.16) for the previous and the current frame, we obtain a set of linear equations:

$$\begin{cases} a_k T_{k-1}^2 + b_k T_{k-1} = N_{k-1} \\ a_k T_k^2 + b_k T_k = N_k\,, \end{cases} \tag{4.21}$$

which we solve for a_k and b_k. In this formulation a_k and b_k do not depend on the a's and b's calculated for other frames. The resulting function $f_I(t)$ is piecewise continuous with jumps at the coded positions $\{\tau_k\}$.

(ii) To make the function $f_I(t)$ continuous, the parameter b_k can be calculated from the values a_{k-1} and b_{k-1} used in the previous frame,

$$b_k = a_{k-1}T_{k-1} + b_{k-1}\,. \tag{4.22}$$

We can obtain a_k as

$$a_k = \frac{N_k - b_k T_k}{T_k^2}\,. \tag{4.23}$$

Quadratic Interpolation

Assuming that within a frame $f_I(t)$ changes quadratically we have

$$f_I(t) = a_k(t-\tau_{k-1})^2 + b_k(t-\tau_{k-1}) + c_k\,, \quad \tau_{k-1} \le t < \tau_k\,. \tag{4.24}$$

To calculate the parameters a_k, b_k and c_k we can use one of the following methods.

(i) We solve the set of linear equations

$$\begin{cases} a_k T_{k-1}^3 + b_k T_{k-1}^2 + c_k T_{k-1} = N_{k-1} \\ a_k {T_k}^3 + b_k {T_k}^2 + c_k T_k = N_k \\ a_k T_{k+1}^3 + b_k T_{k+1}^2 + c_k T_{k+1} = N_{k+1}\,. \end{cases} \tag{4.25}$$

This interpolation will result in piecewise continuous $f_I(t)$, with discontinuities at the coded positions $\{\tau_k\}$.

(ii) To make $f_I(t)$ continuous, we set

$$c_k = a_{k-1}T_{k-1}^2 + b_{k-1}T_{k-1} + c_{k-1}\,, \tag{4.26}$$

and solve for a_k and b_k in

$$\begin{cases} a_k T_k^3 + b_k T_k^2 = N_k - c_k T_k \\ a_k T_{k+1}^3 + b_k T_{k+1}^2 = N_{k+1} - c_k T_{k+1}\,. \end{cases} \tag{4.27}$$

(iii) Using the description of the future pulse lengths to determine the lengths of the pulses in the current frame increases the coding delay. We can calculate the interpolation coefficients without the delay if we require that $f(t)$ is not only continuous but also smooth (the first derivative of $f_I(t)$ is continuous). This leads to the set of linear equations:

$$\begin{cases} c_k = a_{k-1}T_{k-1}^2 + b_{k-1}T_{k-1} + c_{k-1} \\ 2a_kT_{k-1} + b_k = 2a_{k-1}T_{k-1} + b_{k-1} \\ a_kT_k^3 + b_kT_k^2 = N_k - c_kT_k\,. \end{cases} \tag{4.28}$$

which we solve for a_k, b_k and c_k.

(iv) With extra bits of information we could specify a different boundary condition, based on which we could calculate the quadratic interpolation parameters of $f_I(t)$. This approach is used in STC.

Calculating the Pitch Pulse Lengths

Once the $f_I(t)$ interpolation parameters are determined, we calculate the interpolated pitch pulse lengths $P_I(\tilde{t}_i)$. With $\tilde{t}_0 = \tau_{k-1}$

$$\left.\begin{array}{l} \displaystyle\int_{\tilde{t}_j}^{\tilde{t}_{j+1}} f_I(t)dt = 1 \\ P_I(\tilde{t}_j) = \tilde{t}_{j+1} - \tilde{t}_j \end{array}\right\} \quad \text{for} \quad 0 \le j < N_k\,. \tag{4.29}$$

Note that, based on (4.16), we have $\tilde{t}_{N_k} = \tau_k$.

Interpolation in the Instantaneous Period Domain $p(t)$

The interpolated instantaneous period $p_I(t)$ should satisfy

$$\int_{\tau_{k-1}}^{\tau_k} \frac{1}{p_I(t)}dt = N_k\,. \tag{4.30}$$

In the linear interpolation of $p(t)$

$$p_I(t) = a_k(t - \tau_{k-1}) + b_k\,, \quad \tau_{k-1} \le t < \tau_k\,. \tag{4.31}$$

We can calculate a_k and b_k in one of the following ways.

(i) With T_{k-1} and T_k as specified in (4.17) and (4.18), we solve for a_k and b_k in the set of non-linear equations

$$\left\{ \begin{array}{rl} b_k e^{a_k N_{k-1}} &= a_k T_{k-1} + 1 \\ b_k e^{a_k N_k} &= a_k T_k + 1 \,. \end{array} \right. \tag{4.32}$$

(ii) We ensure the continuity of $p(t)$ by specifying the conditions as

$$\left\{ \begin{array}{rl} b_k &= a_{k-1} T_{k-1} + b_{k-1} \\ b_k e^{a_k N_k} &= a_k T_k + 1 \,. \end{array} \right. \tag{4.33}$$

Even in linear interpolation, the calculation of the pitch pulse lengths from the instantaneous period $p_I(t)$ leads to a set of non-linear equations. Linear interpolation in $p(t)$ is used in WI but the pulse lengths are not calculated there.

Calculating the Pitch Pulse Lengths

With $\tilde{t}_0 = \tau_{k-1}$, the interpolated pitch pulse lengths are obtained with:

$$\left. \begin{array}{l} \displaystyle\int_{\tilde{t}_j}^{\tilde{t}_{j+1}} \frac{1}{p_I(t)} dt = 1 \\ P_I(\tilde{t}_j) = \tilde{t}_{j+1} - \tilde{t}_j \end{array} \right\} \quad 0 \le j < N_k \,. \tag{4.34}$$

Based on (4.30) we have $\tilde{t}_{N_k} = \tau_k$.

The pitch pulse-length interpolation used in the PPE coder is described in Section 5.4.2 and Appendix A.

4.2 Pitch Pulse-Shape Interpolation

4.2.1 Spectral Interpolation

Spectral interpolation refers to the idea of the reconstruction of a two-dimensional time-spectrum signal $X_I(t, \omega)$ from an ensemble of spectra $X(t_1, \omega)$, $X(t_2, \omega), \ldots$. Given a one-dimensional time signal $y(t)$, first a two-dimensional signal $x(t, \tau)$ is formed and then a two-dimensional time-frequency signal $X(t, \omega)$ is generated by

applying to $x(t,\tau)$ a transform function with respect to τ. Sampling of $X(t,\omega)$ results in a set of spectra $X(t_1,\omega)$, $X(t_2,\omega),\ldots$. The signal $X_I(t,\omega)$ is formed by performing spectral interpolation on the ensemble $X(t_1,\omega)$, $X(t_2,\omega),\ldots$. A new signal $x_I(t,\tau)$ is constructed through an inverse transform of the interpolated $X_I(t,\omega)$. Finally a one-dimensional time signal $y_I(t)$, akin to the original $y(t)$, is obtained.

This idea has been applied initially, in the context of speech analysis, to the problem of time-scale modification in which the rate of the speech is changed without change of the perceived pitch. Portnoff formulated a mathematical representation of speech and described, based on that representation, a mapping between the one-dimensional speech signal and a two-dimensional time-frequency signal through the Short-Time Fourier Transform (STFT) (Portnoff 1981a,b). The objective was that "temporal features" of the speech appear as a functions of the time variable and "spectral features" appear as function of the frequency variable. The desired rate change of the speech signal was achieved by time-decimation/interpolation of the spectral representation parameters. Signal estimation from the modified, i.e., interpolated, STFT was studied in (Griffin and Lim 1984) and led to an improvement and simplification of the Portnoff system.

At present, speech analysis based on sinusoidal representation and spectral interpolation are used extensively. Harmonic coding (Almeida and Tribolet 1982, Marques *et al.* 1990), Sinusoidal Transform Coding (McAulay and Quatieri 1986, McAulay *et al.* 1991, McAulay and Quatieri 1995), and Multi-Band Excitation (Hardwick and Lim 1988, 1989, Brandstein *et al.* 1990) all use interpolation in the frequency domain. Spectral interpolation has been applied to coding the quasi-periodic LP excitation of voiced speech in the PWI method (Kleijn 1991, Kleijn and Granzow 1991, Kleijn 1993). As mentioned in Chapter 2 the LP model assumes separability between the excitation and the formant structure modelled by the LP filter. Separate analysis and interpolation are therefore carried out on the LP coefficients and on the features extracted from the LP residual. Individual pitch pulses obtained from the LP residual are aligned, coded infrequently and then interpolated.

The spectral interpolation is also employed in Time-Frequency Interpolation (TFI) (Shoham 1992, 1993b). In TFI, the waveform normalization and alignment are eliminated. The inverse Fourier transform is modified so that the reconstructed overlapping blocks of the LP excitation can be interpolated in the time domain.

In addition to mere interpolation of the spectra, Kleijn suggested other modifications of the two-dimensional signals (Kleijn and Haagen 1994a,b, 1995a). In WI, the spectra are filtered with respect to time t to separate their slowly and rapidly changing components[†]. The time signals corresponding to these separated spectra are called the slowly evolving waveform (SEW) and the rapidly evolving waveform (REW). In the context of speech compressing, different coding strategies can be employed with respect to the SEW and the REW (e.g., interpolation of the waveforms, uneven bit assignment).

We now describe the spectral interpolation in more detail.

Spectral Interpolation in WI

The LP residual $y(t)$ is represented as a two-dimensional signal $u(t,\tau)$ created from segments of $y(t)$. The instantaneous pitch period $p(t)$ of the signal $y(t)$ is estimated and then $u(t,\tau)$ is formed from the segments of length $p(t)$ extracted from $y(t)$. The extracted segments are centered near t and the signal $u(t,\tau)$ is given by

$$u(t,\tau) = y(t+\tau)\,, \quad \tau_1(t) \le \tau < \tau_2(t)\,, \tag{4.35}$$

where $\tau_1(t)$ and $\tau_2(t)$ are such that $p(t) = \tau_2(t) - \tau_1(t)$.

In fact $u(t,\tau)$ is formed only at discrete time instances t_i so that an ensemble $\{u(t_i,\tau)\}$ is created. Each of the signals in the set $\{u(t_i,\tau)\}$ is normalized in the τ domain forming a length-normalized

$$\underline{u}(t_i,\phi) = u(t_i, \frac{2\pi}{p(t)}\phi)\,. \tag{4.36}$$

The signals in the ensemble $\{\underline{u}(t_i,\phi)\}$ are periodically extended in ϕ and aligned for maximum correlation with respect to the previous, already aligned signal of the ensemble,

$$x(t_i,\phi) = \underline{u}(t_i,\phi+\psi) \tag{4.37}$$

[†]For a fixed frequency ω, the two-dimensional signal $X(t,\omega)$ becomes one-dimensional $X_\omega(t)$. In the filtering of $X(t,\omega)$ with respect to t, the set of the one-dimensional signals $X_\omega(t)$ is filtered. The filtered two-dimensional signal corresponding to $X(t,\omega)$ is reconstructed from the filtered one-dimensional signals.

with

$$\psi = \arg\max_{\psi}\Big(\int_{2\pi} \underline{u}(t_i, \phi+\psi)x(t_{i-1},\phi)d\phi\Big)\,. \tag{4.38}$$

In the description of WI the signals in the ensemble $\{x(t_i,\phi)\}$ are called characteristic waveforms.

Since $x(t_i,\phi)$ is periodic in 2π with respect to ϕ, its Fourier series is given by

$$X(t_i,k) = \frac{1}{2\pi}\int_{2\pi} x(t_i,\phi)\, e^{-jk\phi}d\phi\,. \tag{4.39}$$

The number of significant coefficients M_{t_i} is determined by the pitch period $p(t_i)$ and the bandwidth of the signal $u(t_i,\tau)$ with respect to τ. Since M_{t_i} depends on $p(t_i)$, the calculation of the spectra $X(t_i,k)$ requires the use of Discrete Fourier Transform (DFT) of different lengths, viz fast Fourier techniques cannot be used for all M_{t_i}.

The Fourier transforms $X(t_i,k)$ can be viewed as a sampled (decimated) version of the time-continuous signal $X(t,k)$. The missing spectra are obtained through an interpolation procedure $\mathcal{I}$,

$$X_I(t,k) = \mathcal{I}\{X(t_1,k), X(t_2,k), \ldots, X(t_i,k), \ldots\}\,. \tag{4.40}$$

A multitude of interpolations $\mathcal{I}$ can be specified but only linear interpolations have been used. Linearity of the interpolation $\mathcal{I}$ is understood in the sense that

$$\mathcal{I}\{X(t_{i-1},k), X(t_i,k)\} = \alpha(t)X(t_{i-1},k) + \beta(t)X(t_i,k)\,, \quad t_{i-1} \le t < t_i \tag{4.41}$$

with the interpolation coefficients $\alpha(t)$ and $\beta(t)$ not necessarily linear in time. The linear operator $\mathcal{I}$ can also be applied separately to the spectral magnitude and phase, in which case the interpolation is non-linear with respect to the complex values $X(t_i,k)$.

The signal $x_I(t,\phi)$ is created through the inverse Fourier transform of $X_I(t,k)$,

$$x_I(t,\phi) = \sum_k X_I(t,k)\, e^{jk\phi}\,. \tag{4.42}$$

The signal $y_I(t)$ can be obtained from $x_I(t,\phi)$ as

$$y_I(t) = x_I\Big(t, \phi(t_{i-1}) + \int_{t_{i-1}}^{t}\frac{2\pi}{p(t')}dt'\Big)\,, \quad t_{i-1} \le t < t_i\,, \tag{4.43}$$

where $\phi(t_{i-1})$ is the initial phase at time t_{i-1} such that $y(t_{i-1}) = x_I(t, \phi(t_{i-1}))$.

Spectral Interpolation in TFI

In TFI, the discrete LP residual is not regarded as continuous signal. The signal $u(n,m)$ is formed based on the discrete instantaneous pitch period $p(t)$,

$$u(n,m) = s(n+m)\,, \quad m_1(n) \le m < m_2(n) \tag{4.44}$$

where $m_1(n)$ and $m_2(n)$ are such that

$$p(n) = m_2(n) - m_1(n)\,. \tag{4.45}$$

Only a decimated version of the signal $u(n,m)$ is created so that the ensemble $\{u(n_i,m)\}$ is formed.

The TFI coder does not normalize the waveforms $\{u(n_i,m)\}$ with respect to m. One cannot therefore proceed the same way as in spectral interpolation used in WI: waveform alignment, Fourier transform, interpolation, inverse Fourier transform, mapping from the two-dimensional signal to the one-dimensional signal. In TFI it is assumed that these operations commute (Shoham 1992, 1993b). The waveform alignment is eliminated and the inverse DFT is based on an approximation to time-scale modification. The modified inverse DFT allows a gradual change of the instantaneous period of the time waveform. This is achieved by making the phase of the basis functions of the transform independent of the DFT size, and changing it according to the required instantaneous period. Linear interpolation is not performed on the decimated version of $X(n,k)$ (the ensemble $\{X(n_i,k)\}$), but on the two-dimensional time signal obtained through the modified inverse DFT, the ensemble $\{x_I(n_i,m)\}$.

The DFT is calculated as

$$X(n_i,k) = \sum_{m=m_1(n_i)}^{m_2(n_i)-1} u(n_i,m)\; e^{-j\frac{2\pi}{p(n_i)}km}\,, \quad k = 0,\ldots,p(n_i)-1\,. \tag{4.46}$$

The two-dimensional signal $x_I(n_i,m)$ is obtained via the inverse DFT modified to

$$x_I(n_i,m) = \sum_{k=0}^{p(n_i)-1} X(n_i,k)\; e^{-j\Phi(n_i,m)k}\,. \tag{4.47}$$

The phase of the basis functions of the inverse transform $\Phi(n_i, m)$ is computed as

$$\Phi(n_i, m) = \Phi(n_i, n_i) + 2\pi\, f_I(n_i, m)\,(m - n_i) \tag{4.48}$$

with the initial phase

$$\Phi(n_i, n_i) = \frac{2\pi}{p(n_i)} n_i\,. \tag{4.49}$$

The interpolated frequency $f_I(n_i, m)$ is given by

$$f_I(n_i, m) = \begin{cases} \dfrac{1 - \alpha_p(m)}{p(n_{i-1})} + \dfrac{\alpha_p(m)}{p(n_i)} & \text{for } n_{i-1} \le m < n_i \ , \\ \dfrac{1 - \alpha_p(m)}{p(n_i)} + \dfrac{\alpha_p(m)}{p(n_{i+1})} & \text{for } n_i < m < n_{i+1} \ . \end{cases} \tag{4.50}$$

The interpolation coefficient $\alpha_p(n)$ is specified as

$$\alpha_p(n) = \frac{n \bmod N}{N}\,. \tag{4.51}$$

A linear interpolation is applied to the two-dimensional time signals

$$y_I(n) = \Big(1 - \alpha(n)\Big) x_I(n_{i-1}, n) + \alpha(n) x_I(n_i, n)\,, \quad n_{i-1} \le n < n_i \tag{4.52}$$

with

$$\alpha(n) = \alpha_p(n)\,. \tag{4.53}$$

In both WI and TFI, the spectral interpolation is performed with fixed rate, i.e., the time interval $t_i - t_{i-1}$ in the WI and the interval $n_i - n_{i-1}$ in the TFI coder are constant.

4.2.2 Spectral Interpolation in the PPE model

In the PPE coder, spectral interpolation is used to interpolate waveshapes of pitch pulses. The pitch pulses are extracted from the LP residual in such a way that the pulses are aligned. Every pitch pulse is regarded as a separate entity and the pulses are padded with zeros to a common length. The underlying pitch pulses are estimated in the time domain and the estimated pulses are transformed into frequency domain with the Fast Fourier Transform (FFT) algorithm. Use of the computationally

efficient FFT is feasible because all pitch pulses are of the same length. We avoid the computationally expensive direct DFT calculations necessary in the WI and the TFI coders.

Once per frame, the pitch pulse-shape update information is transmitted. The waveshapes of the intermediate pulses are formed by linear interpolation. Since the number of pitch pulses varies from frame to frame, the number of pitch pulse shapes to be interpolated is not constant. Our interpolation is linear "in the number of interpolated pulses" and not linear "in time". It means that the coefficients of the linear interpolation depend on the number of pulses to be interpolated and not on the relative positions (lengths) of the pulses.

We do not modify the spectra of the pulses prior to interpolation. The smoothing of the evolution of the pulses is performed in the time domain in the process of estimation of the underlying pitch pulses as described in Section 3.3.4. The estimation of the underlying pulses can also be performed on the pulse spectra. This would correspond to the spectral modifications (filtering with respect to time t) employed prior to the interpolation in WI.

4.3 Summary

In this chapter we have analyzed the pitch interpolations used in WI, STC and RCELP. We have argued that the time warping of the pitch pulses resulting from the interpolations employed in those coders is not justified from the point of view of the characteristics of the LP residual. In the PPE model, the interpolation of the pitch pulse length and the interpolation of the pitch pulse waveshape are decoupled and the time warping is avoided.

The interpolation of the pitch pulse lengths has been described in terms of (i) linear interpolation of the pulse lengths, (ii) linear and quadratic interpolation of the fundamental frequency, (iii) linear interpolation of the instantaneous pitch period. Whatever the domain, the interpolated pitch pulse lengths are calculated and the LP excitation is formed based on the relation between the lengths of the original and the reconstructed pulses.

Finally, we have described spectral interpolation used in WI and TFI. The spectral interpolation is used in the PPE coder to interpolate waveshapes of the pitch pulses. The interpolation of the pulse waveshapes does not effect the interpolated pitch pulse

positions. The waveshape interpolation is performed in the frequency domain on the spectra of the underlying pitch pulses.

The presentation in this chapter was based on the Fourier transform and the interpolation was performed in the frequency domain. Any other transformation determined to be more appropriate for a particular type of interpolation or spectral modification can be used; the Discrete Cosine Transform (DCT) is one of the viable alternatives.

Chapter 5

Implementation of the 4 kb/s PPE Coder

The PPE model described in the previous chapters has been implemented as a 4 kb/s coder. The PPE coder is an analysis-by-synthesis based LP coder with emphasis on modelling the voiced LP excitation. The pitch pulse analysis is performed pitch synchronously and the unvoiced contribution to the LP excitation is coded with a fixed-block-length analysis. Individual pitch pulses are identified in the LP residual and the pulses are extracted. One pitch pulse position is encoded per frame and the intermediate pulse positions are determined via pitch pulse-length interpolation. A modified LP residual is created by shifting the original pitch pulses to the new positions and the modified speech signal is formed.

The underlying pitch pulses are estimated based on the pulses extracted from the LP residual. The underlying pulses are decimated and one pulse per frame is coded in the frequency domain. The intermediate underlying pitch pulses are obtained via pitch pulse-shape interpolation. The noise contribution is coded with the generalized analysis-by-synthesis procedure (with respect to the modified speech signal). The gain is encoded as the total gain and the gain ratio between the underlying pitch pulses and the superimposed noise.

A detailed description of the coder follows.

5.1 The Coder Structure

The block diagram of the encoder is presented in Fig. 5.1. The encoder includes the following stages:

- *Linear prediction analysis*: The LP coefficients are calculated and coded (Section 5.2). The LP residual is obtained by filtering the original speech using the unquantized LP coefficients.
- *Pitch pulse position analysis*: The pitch pulses are extracted from the LP residual. The extraction of the pitch pulses is executed (Section 5.3) and the pitch pulse positions are coded (Section 5.4).
- *Pitch pulse waveshape analysis*: The extracted pitch pulses are amplitude normalized. The pulse gain is quantized and pitch synchronously interpolated (Section 5.5).
  The underlying pitch pulses are estimated and transformed to the frequency domain. The current underlying pitch pulse is predicted and based on the prediction, the waveshape of the current underlying pitch pulse is coded. The coded pulses are interpolated to render the intermediate underlying pulses (Section 5.6).
- *Noise analysis*: The relative ratio between the noise component and the underlying pitch pulse component is calculated and coded. The difference between the extracted pitch pulses and the coded interpolated pulses is placed as the unvoiced residual at the coded pitch pulse positions. The thus formed noise signal is coded based on analysis-by-synthesis using a perceptual weighting filter based on the unquantized LP filter $A(z)$ (Section 5.7).

The following parameters are coded:

$\hat{A}(z)$	–	The LP filter with coefficients coded in the LSF domain.
$\hat{P}$	–	Pitch positions and number of pitch pulses between frames.
$\hat{g}$	–	Total gain of the LP residual (calculated pitch synchronously).
$\hat{D}(n)$	–	Pitch pulse shape coded in the frequency domain. One extra bit is used to switch between differential and non-differential coding.
$\hat{N}(n)$	–	Noise shape in the time domain.
$\hat{r}_{n/p}$	–	Relative gain of the superimposed noise.

The bit allocation of the 4 kb/s coder for all the coded parameters is presented in Table 5.1. In every frame, the LSF parameters, the pitch pulse positions and the total gain are coded. In frames with pitch pulses, the remaining bits are used for coding the pitch pulse shape, pitch pulse noise, and pulse to noise gain ratio. In frames with no pitch pulses, the remaining bits are used to code the noise shape.

Table 5.1 Bit allocation in the 4 kb/s PPE coder.

	Bits/Samples	Bits/Frame	Bits/Second	Update Rate
Line Spectral Frequencies	30 / 160	30	1500	50 Hz
Pitch Position	8 / 160	8	400	50 Hz
Total Gain	5 / 80	10	500	100 Hz
When Pitch Pulses Are Identified:				
Pitch Pulse Shape	9 / 160	9	450	50 Hz
Absolute Shape/Pulse Difference	1 / 160	1	50	50 Hz
Pitch Pulse Noise	4 / 40	16	800	200 Hz
Pulse to Noise Ratio	3 / 80	6	300	100 Hz
When No Pitch Pulse Is Identified:				
Noise Shape	8 / 40	32	1600	200 Hz
The Total Number Of Bits Used		80	4000	

The block diagram of the PPE decoder is presented in Fig. 5.2. The decoder includes the following stages:

- *Generating the pitch pulses*: The shape of the underlying pitch pulse is created based on the predicted pitch pulse shape and the coded pitch pulse difference. The intermediate pitch pulses are formed by interpolation.
- *Pitch pulse positioning*: The pitch pulses are placed at the coded pulse positions.
- *Adding the noise component*: The coded noise is added to form the LP excitation. The gain is pitch synchronously interpolated and applied to the excitation.
- *Adding the formant structure*: The excitation is filtered with the LP synthesis filter to produce the coded speech.

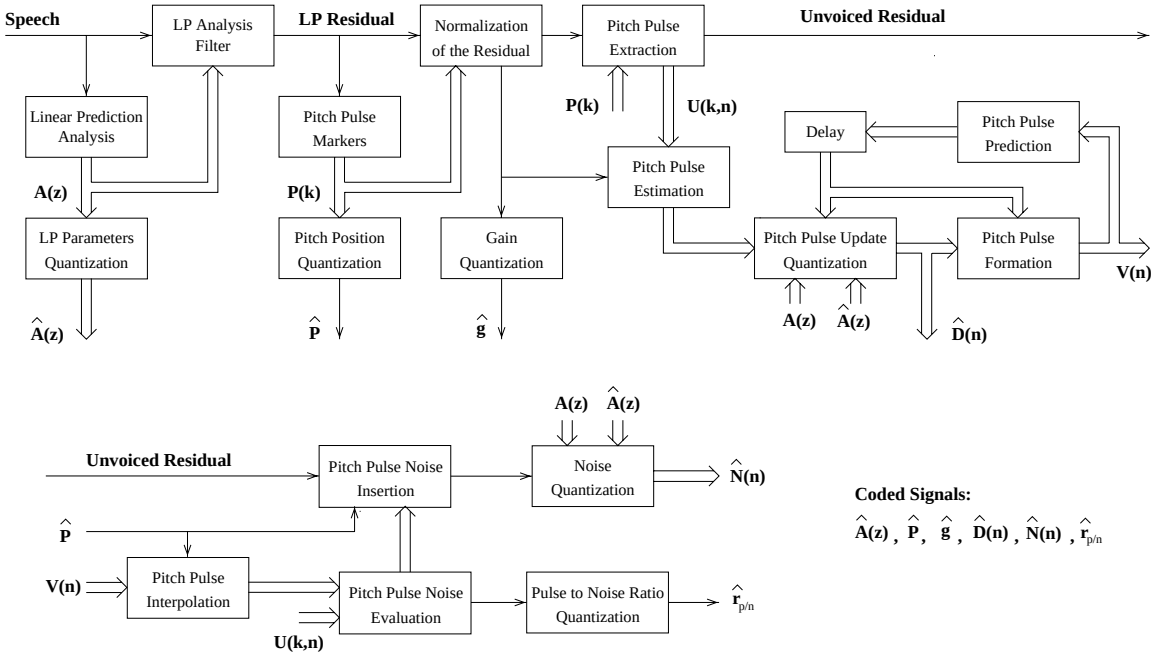


Fig. 5.1 Block diagram of the PPE encoder.

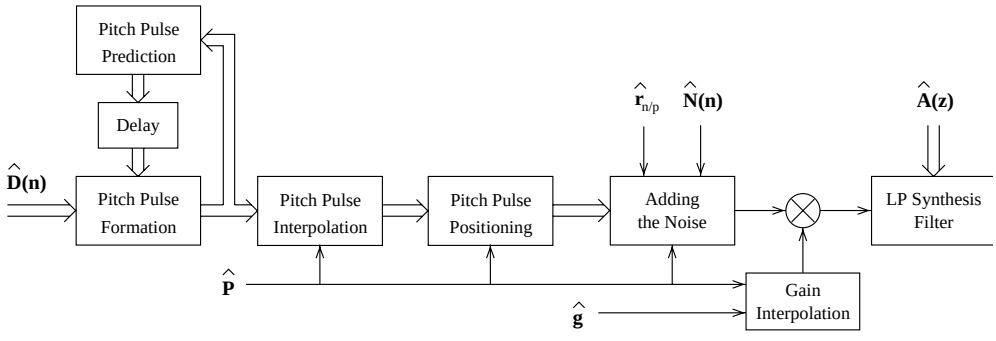


Fig. 5.2 Block diagram of the PPE decoder.

In the art of speech coding many extra components like pre- and post-processing are used to improve the quality of the coded speech. At present, we do not use any additional techniques in our coder. We have concentrated on the proper implementation of the features which are unique to our coder and on demonstrating that we can achieve a very high quality speech without the help of these extra components.

5.2 Linear Prediction Analysis and Coding

The 10-th order LP linear prediction analysis is performed every 20 ms with the autocorrelation method. We use a Hamming window[†] of length 240 ms centered on frame boundaries. The LP coefficients are calculated with the Levinson-Durbin recursion (Markel and Gray 1976, Rabiner and Schafer 1978). We use bandwidth expansion[‡] in which the LP coefficient a_n is multiplied by γ^n with $\gamma = 0.977$. The LP coefficients are converted to the LSF domain with the method described in (Kabal and Ramachandran 1986).

In our bit budget for the 4 kb/s coder we allocated 30 bits per frame for coding the LP parameters. In our initial implementation of the LSF codebooks however, we assumed coding the LP parameters with 24 bits per update. Subsequently all the testing presented in this chapter has been performed with LP parameters coded with only 24 bits/frame. We used two split codebooks, one for the first 4 LSFs and the other for the last 6 LSFs, trained with the Generalized Lloyd Algorithm (GLA) (Gersho and Gray 1992) on a database which did not include the test sentences. The error weighting used in the quantization of the LP parameters is as suggested in (Paliwal and Atal 1993). The training of the codebooks as well as the development of the software used in the training are part of our work.

Since we use only 24 bits/frame for coding the LP parameters, the actual bit rate of the implemented coder is 300 b/s lower than 4 kb/s (i.e., 3.7 kb/s). These bits could be reallocated to improve other aspects of the coder. Time did not permit implementation thereof within the scope of this work. The vectors of unquantized LSFs are linearly interpolated with an up-sampling rate of 10 and converted back to LP

[†]We note that modern coders have adopted non-symmetric windows which help in reducing the algorithmic delay of the coder. Our focus in this thesis is on the pitch pulse coding – any improvements to the LP analysis will also help PPE.

[‡]Bandwidth expansion has been shown to avoid unnaturally peaky formant structure and to help reducing quantization cross-overs of closely spaced LSFs.

coefficients. The LP residual is calculated by inverse filtering using the interpolated unquantized LP coefficients.

5.3 Pitch Pulse Extraction

As explained in Section 3.2, the extraction of pitch pulses is viewed as a problem of appropriate segmentation of the LP residual. The segmentation is based on the minimization of the prediction error between (i) model pulses (underlying pulses of the simplified estimation) (ii) the noisy pulses of the LP residual.

The implementation details are presented in this section. The constants used in our algorithm are specified in Table 5.2. Except for N_P and F_{ups}, all the integer values in the table are given in samples for the 8 kHz sampling rate. The values marked with an asterisk are subject to up-sampling rate F_{ups}, which in our coder is equal to eight.

5.3.1 Frame Classification

The residual is divided into frames of length L_F. The boundaries of frame k are written as t_k and t_{k+1},

$$t_{k+1} = t_k + L_F \,. \tag{5.1}$$

Four types of frames are identified:

noise frame – no pitch pulses,
start frame – the pitch pulses start,
continue frame – the series of pitch pulses continues,
end frame – the series of pitch pulses ends.

The following transitions between frames are allowed:

noise frame $\longrightarrow$ *noise* or *start* frame,
start frame $\longrightarrow$ *continue* or *end* frame,
continue frame $\longrightarrow$ *continue* or *end* frame,
end frame $\longrightarrow$ *noise* or *start* frame.

For the purpose of comparing alternative segmentations, four errors are determined in every frame:

Table 5.2 The constants used in the pitch extraction algorithm. The values marked with an asterisk are subject to up-sampling rate F_{ups}, which in the described coder is equal to eight.

Symbol	Value	Description
L_F	160 *	Frame length
P_{min}	20 *	Minimum pitch pulse length
P_{max}	150 *	Maximum pitch pulse length
δ_{ne}	0.7	Noise error scaling coefficient
N_P	2	Maximum number of peaks in a window of length L_P
L_P	20 *	Length of the window in which at most N_P peaks are allowed
L_{P_S}	10 *	Length of the shadow cast by a peak
P_{offs}	10 *	Pitch pulse offset
$L_S^{(s)}$	240 *	Length of the segmentation window in the *start* mode
$L_S^{(c)}$	320 *	Length of the segmentation window in the *continue* mode
$\delta_{P\uparrow}$	0.3	Maximum relative pulse length decrease (0.3 means by 30%)
$\delta_{P\downarrow}$	0.3	Maximum relative pulse length increase (0.3 means by 30%)
α_p	0.5	Estimation filter coefficient
P_{align}	8 *	Maximum pitch pulse alignment offset
L_{E_0}	5 *	Start sample for the Pitch pulse energy shaping
L_E	10 *	Length of the pitch pulse energy shaping window
D_{pred}	60	Maximum dimension of the model pitch pulse selection
D_{align}	60	Maximum dimension of the pulse alignment
L_{F_e}	40 *	Maximum frame extension for frame error calculation
δ_{pe}	0.9	Minimum normalized prediction error
F_{ups}	8	Up-sampling rate

$e^{(\mathrm{n})}$ – *noise* frame error,
$e^{(\mathrm{s})}$ – *start* frame error,
$e^{(\mathrm{c})}$ – *continue* frame error,
$e^{(\mathrm{e})}$ – *end* frame error.

The errors are calculated with a one frame look-ahead.

The cumulative error of the present and the next frame is calculated and, based on this error, the type of the present frame is determined. If the last frame was identified as *noise* or *end* we compute:

$$\varepsilon_n = e_k^{(\mathrm{n})} + \min(e_{k+1}^{(\mathrm{n})}, e_{k+1}^{(\mathrm{s})})\,, \tag{5.2}$$

$$\varepsilon_s = e_k^{(\mathrm{s})} + \min(e_{k+1}^{(\mathrm{c})}, e_{k+1}^{(\mathrm{e})})\,. \tag{5.3}$$

The current frame is classified as a *noise* frame if $\varepsilon_n < \varepsilon_s$ and as a *start* frame otherwise. If the last frame was identified as *start* or *continue* we calculate:

$$\varepsilon_c = e_k^{(\mathrm{c})} + \min(e_{k+1}^{(\mathrm{c})}, e_{k+1}^{(\mathrm{e})})\,, \tag{5.4}$$

$$\varepsilon_e = e_k^{(\mathrm{e})} + \min(e_{k+1}^{(\mathrm{n})}, e_{k+1}^{(\mathrm{s})})\,. \tag{5.5}$$

The current frame is classified as a *continue* frame if $\varepsilon_c \leq \varepsilon_e$ and as an *end* frame otherwise.

The errors $e^{(\mathrm{n})}$, $e^{(\mathrm{s})}$, $e^{(\mathrm{c})}$ and $e^{(\mathrm{e})}$ are computed based on the segmentation of the residual and the identified model pitch pulses. For the calculation of $e^{(\mathrm{n})}$ there is no segmentation; the errors $e^{(\mathrm{s})}$, $e^{(\mathrm{c})}$ and $e^{(\mathrm{e})}$ correspond to the segmentations obtained in the *start*, *continue* and *end* mode, respectively.

In determining the type of the current frame, the errors based on the segmentation of the next frame are used. The pitch pulse extraction look-ahead beyond the current frame would normally be equal to $L_S^{(\mathrm{c})}$ (the length of the longest segmentation window). To decrease the look-ahead, first the segmentations for the next frame are performed with the lengths of the segmentation windows limited to frame length L_F. This may result in a different segmentation than for longer windows, but since those segmentations are used only in error calculations they are not crucial. When the present "next frame" becomes the current frame, the segmentations are redefined based on the longer segmentation windows.

5.3.2 Error Calculation

The Error Between Pitch Pulses

The error between a model pitch pulse and a pulse identified in the LP residual is specified by the prediction error between the pulses. The prediction error between vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as

$$\mathcal{E}_p(\boldsymbol{x}, \boldsymbol{y}) = \min_{\beta}(\boldsymbol{y} - \beta\boldsymbol{x})^2 \,. \tag{5.6}$$

We want to predict the LP-residual noisy pulses from the model pitch pulses. In this formulation we predict vector $\boldsymbol{y}$ from vector $\boldsymbol{x}$ and hence $\boldsymbol{y}$ corresponds to the noisy pulses and $\boldsymbol{x}$ corresponds to the model pulses.

The optimal prediction gain β which minimizes the error is calculated as

$$\beta_{opt} = \frac{\boldsymbol{x}^T\boldsymbol{y}}{\boldsymbol{x}^T\boldsymbol{x}} \,. \tag{5.7}$$

With this choice of beta, the prediction error is given by

$$\mathcal{E}_p(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{y}^T\boldsymbol{y} - \beta_{opt}\boldsymbol{x}^T\boldsymbol{y} \,. \tag{5.8}$$

Writing

$$\mathcal{C}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T\boldsymbol{y} \quad \text{and} \quad E(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{x} \,, \tag{5.9}$$

we obtain

$$\beta_{opt} = \frac{\mathcal{C}(\boldsymbol{x}, \boldsymbol{y})}{E(\boldsymbol{x})} \quad \text{and} \quad \mathcal{E}_p(\boldsymbol{x}, \boldsymbol{y}) = E(\boldsymbol{y}) - \beta_{opt}\mathcal{C}(\boldsymbol{x}, \boldsymbol{y}) \,. \tag{5.10}$$

If the vectors $\boldsymbol{x}$, $\boldsymbol{y}$ are normalized in amplitude, i.e., $E(\boldsymbol{x}) = 1$ and $E(\boldsymbol{y}) = 1$, the above equations simplify to

$$\beta_{opt} = \mathcal{C}(\boldsymbol{x}, \boldsymbol{y}) \quad \text{and} \quad \mathcal{E}_p(\boldsymbol{x}, \boldsymbol{y}) = 1 - \beta_{opt}^2 \,. \tag{5.11}$$

This formulation assumes that vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are of equal length. If $\boldsymbol{x}$ is shorter than $\boldsymbol{y}$, we extend the vector $\boldsymbol{x}$ with zeros so that both vectors have the same length. If $\boldsymbol{x}$ is longer than $\boldsymbol{y}$, we truncate the vector $\boldsymbol{x}$ to the length of the vector $\boldsymbol{y}$. Note that it is always vector $\boldsymbol{x}$ which is extended or truncated. The vector $\boldsymbol{y}$ is predicted from $\boldsymbol{x}$ and so it keeps its initial length.

Since the prediction error is applied to the pitch pulses and we assume that the pulses are well correlated, we limit β to positive values. If $\beta_{opt} < 0$ we set $\beta_{opt} = 0$.

The Noise Error

The region in which no pitch pulses are identified is considered as "unvoiced only". Conceptually, the prediction error of an "unvoiced only" region is equal to the energy of the signal. We introduce a correction factor δ_{ne}, which is called the noise error scaling coefficient. We calculate the error of an unvoiced region as the energy of the region scaled by δ_{ne}.

The value of δ_{ne} affects our acceptance of a segmentation as a series of pitch pulses. If $\delta_{ne} = 1$, any segmentation with positive correlation between the model pulses and the segments of the residual signal will be considered as a series of pitch pulses. Also, once the series starts it might never end. We want, however, to identify the beginning of a series to be able to correctly specify and align the pulses. If $\delta_{ne} = 0$, none of the segmentations will be good enough to be considered for a pitch pulse series. We tested values of δ_{ne} in the range from 0.6 to 0.9. The influence of δ_{ne} on the frame classification is weakened by the calculation of the cumulative error of two frames and by frame extension, which is explained later in this section. We got very good results using $\delta_{ne} = 0.7$.

5.3.3 Segmentation of the LP Residual

The pitch pulses are defined by a segmentation of the LP residual. The series of pitch pulses form the voiced regions of the residual[†]. The segmentation is carried out in three modes: *start*, *continue* and *end*. For frame k, the three modes provide the pulse markers $\tau_{k,i}^{(s)}$, $\tau_{k,i}^{(c)}$ and $\tau_{k,i}^{(e)}$. The number of pulse markers for the frame identified in a particular mode is written as N where $0 \leq i \leq N$. The $N+1$ markers delimit N complete pulses.

The pitch markers $\tau_{k,i}^{(s)}$, $\tau_{k,i}^{(c)}$ and $\tau_{k,i}^{(e)}$ designate the positions of consecutive pitch pulses. To make our presentation more readable we will drop, for the time being, the superscripts $^{(s)}$, $^{(c)}$ and $^{(e)}$. We will use them only when we need to differentiate

[†]As noted earlier, in the PPE model the voiced regions contain a noise component which is here the difference between the estimated model pulses and the LP residual.

between the parameters and the pitch pulse markers specific to, or obtained in, a particular segmentation mode.

To simplify the notation we will also drop the subscript k on the pulse position markers τ and the corresponding pitch pulse vectors. It should be clear from the context that they belong to the segmentation performed for the frame k. What should be written as $\tau_{k,i}$ is now simply τ_i.

In our notation a continuous block of samples of the residual $\boldsymbol{r}$ beginning at t and ending at t' which includes the sample r_t but does not include the sample $r_{t'}$ is written as $\boldsymbol{r}\,[t : t')$. The pitch pulse between marks τ_i and τ_{i+1} is written as $\boldsymbol{p}_i$,

$$\boldsymbol{p}_i = \boldsymbol{r}\,[\tau_i : \tau_{i+1})\,. \tag{5.12}$$

The length of the pulse $L_{\boldsymbol{p}_i}$ is given by

$$L_{\boldsymbol{p}_i} = \tau_{i+1} - \tau_i\,. \tag{5.13}$$

The model pitch pulses are written as $\boldsymbol{q}_i$. The dimension of the vector $\boldsymbol{q}$ is always extended to P_{max}. When the vector $\boldsymbol{q}$ is formed from shorter vectors they are padded with zeros to length P_{max}.

Candidate Pitch Pulse Positions

We determine all possible positions of the pulses based on the energy maxima of the LP residual signal. No more than N_P energy maxima for every L_P samples are accepted. Each identified maximum casts a forward "shadow" of L_{P_S} samples within which no smaller energy peak is recognized. A larger maximum resets the "shadow" and casts one of its own. An energy maximum is accepted as a candidate pitch pulse position if it is one of the N_P largest energy peaks within $\pm L_P$ samples; the candidate pulse position is set at P_{offs} samples prior to the energy maximum. In the segmentation of the LP residual, the pitch pulse positions τ_i are chosen from the set of the candidate pitch pulse positions.

The Model Pulses

Every pulse $\boldsymbol{p}_i$ has a corresponding model pulse $\boldsymbol{q}_i$. Ideally, the model pulse $\boldsymbol{q}_i$ should be some estimate of the underlying pitch pulse. At this stage of the process, we use simplified model pulses to obtain a segmentation. The pulses defined by the segmentation are used in the underlying pulse estimation procedure which is invoked later. The model pulse $\boldsymbol{q}_i$ is one of the following:

- The previous pitch pulse vector $\boldsymbol{p}_{i-1}$.
- The next pitch pulse vector $\boldsymbol{p}_{i+1}$.
- The average of the previous and the next pulses $\boldsymbol{p}'_i$, given by

$$\boldsymbol{p}'_i = \frac{\boldsymbol{p}_{i-1} + \boldsymbol{p}_{i+1}}{2} \,. \tag{5.14}$$

- The average of the past pitch pulse vectors $\boldsymbol{q}'_i$ given by

$$\boldsymbol{q}'_i = (1 - \alpha_p)\, \boldsymbol{q}_{i-2} + \alpha_p\, \boldsymbol{p}_{i-1} \,. \tag{5.15}$$

  The coefficient α_p is the weight of the last noisy pitch pulse $\boldsymbol{p}_{i-1}$. If the model pitch pulse $\boldsymbol{q}_{i-2}$ is equal to $\boldsymbol{p}_{i-1}$, the above would render $\boldsymbol{q}'_i$ equal to $\boldsymbol{p}_{i-1}$. In this case we set $\boldsymbol{q}'_i$ as

$$\boldsymbol{q}'_i = (1 - \alpha_p)\, \boldsymbol{p}_{i-2} + \alpha_p\, \boldsymbol{p}_{i-1} \,. \tag{5.16}$$

The vector chosen for the model pulse $\boldsymbol{q}_i$ is the one which minimizes the prediction error with respect to the vector $\boldsymbol{p}_i$. We have

$$\boldsymbol{q}_i = \underset{\boldsymbol{x} \in \{\boldsymbol{p}_{i-1}, \boldsymbol{p}_{i+1}, \boldsymbol{p}'_i, \boldsymbol{q}'_i\}}{\arg\min} \mathcal{E}_p(\boldsymbol{x}, \boldsymbol{p}_i) \tag{5.17}$$

with $\boldsymbol{q}'_i$ and $\boldsymbol{p}'_i$ as specified above.

At the beginning of a frame, the past pulses $\boldsymbol{q}_{-2}$, $\boldsymbol{p}_{-2}$ and $\boldsymbol{p}_{-1}$ are taken from the previous frames. In the *start* mode the past pulses come from the last *end* frame; this "memory" of pitch pulses was found useful in identifying the beginning of a series. Note that between a *start* frame and an *end* frame there might be intervening *noise* frames. In the *continue* and *end* mode the pulses $\boldsymbol{q}_{-2}$, $\boldsymbol{p}_{-2}$, $\boldsymbol{p}_{-1}$ come from the last

start or *continue* frame.

Energy Shaping of The Model Pulses

To prevent misidentifying two pitch pulses as a single pulse, the model pulse $\boldsymbol{q}$ is energy shaped. The shaping is such that after the initial increase in the pulse energy, the energy may only decrease. If there are two pulses in the vector $\boldsymbol{q}$, the energy of the second pulse will be attenuated to the energy of the signal between the two pulses. The energy attenuation increases the prediction error between the model pulse $\boldsymbol{q}$ and the corresponding pulse $\boldsymbol{p}$, so that the total prediction error of a double-pitch-pulse segmentation is large and the segmentation is rejected.

The energy shaping starts at sample L_{E_0} from the beginning of a pulse. First, an energy maximum $E_{\max}$ at sample $n_{\max}$ is found such that the next L_E samples have smaller energy. Then, starting at $n_{\max} + L_E$, the pulse vector is searched for a sample with higher energy than $E_{\max}$. If at sample $n'_{\max}$ the energy $E'_{\max}$ is larger than $E_{\max}$, the rest of the vector is scaled by $E_{\max}/E'_{\max}$. The search for a sample with energy higher than $E_{\max}$ continues starting at $n'_{\max}$ and the vector is modified again if such a sample with higher energy is found. When the end of the vector is reached, the whole procedure is repeated starting at sample $n_{\max}$+1 (i.e., find an energy maximum $E_{\max}$ such that the next L_E samples have smaller energy...).

Pulse Alignment

We want to align the vector $\boldsymbol{p}_i$ with respect to the model pulse vector $\boldsymbol{q}_i$. To do this, we fix the length of the pulse $\boldsymbol{p}_i$, $L_{\boldsymbol{p}_i}$, and we shift τ_i to reduce the prediction error between $\boldsymbol{q}_i$ and the shifted pulse $\boldsymbol{p}_i$. The pulse $\boldsymbol{p}_i$ of length $L_{\boldsymbol{p}_i}$ starting at τ_i is equivalent to the segment of the LP-residuals given as $\boldsymbol{r}\,[\tau_i : \tau_i + L_{\boldsymbol{p}_i})$. The beginning of the aligned pulse is given by

$$\bar{\tau}_i = \underset{\tau_i - P_{align} \le l \le \tau_i + P_{align}}{\arg\min} \mathcal{E}_p\left(\, \boldsymbol{q}_i \,,\, \boldsymbol{r}\,[\tau_i : \tau_i + L_{\boldsymbol{p}_i})\,\right). \tag{5.18}$$

The constant P_{align} specifies the largest shift allowed when aligning the pulses.

The Segmentation Algorithm

So far we have defined the model pitch pulse $\boldsymbol{q}_i$ as one of the following: $\boldsymbol{p}_{i-1}$, $\boldsymbol{p}_{i+1}$, $\boldsymbol{p}'_i$ or $\boldsymbol{q}'_i$. The vector $\boldsymbol{p}'_i$ is the average of the vectors $\boldsymbol{p}_{i-1}$ and $\boldsymbol{p}_{i+1}$; the vector $\boldsymbol{q}'_{i-1}$ is specified by (5.15)–(5.16). In fact we do not know the next pulse, $\boldsymbol{p}_{i+1}$, when we want to establish the vector $\boldsymbol{q}_i$ and align the vector $\boldsymbol{p}_i$. To overcome the problem the segmentation is done with the following procedure:

1. Set $i = 0$ and establish the vectors $\boldsymbol{p}_{-1}$ and $\boldsymbol{q}_{-1}$.
2. Determine the pitch pulse candidate $\boldsymbol{p}_i$.
3. Calculate the average of the past pulses to form the vector $\boldsymbol{q}'_i$ as specified in (5.15)–(5.16).
4. Find an initial estimate of the model pitch pulse based only on the past pulses. The initial model pulse $\boldsymbol{q}_i^{(1)}$ is one of the vectors $\boldsymbol{p}_{i-1}$, $\boldsymbol{q}'_i$. Choose the vector which minimizes the prediction error with respect to $\boldsymbol{p}_i$.
5. Align the vector $\boldsymbol{p}_i$ with respect to $\boldsymbol{q}_i^{(1)}$.
6. If $i > 0$, reestimate the model vector corresponding to the last pulse $\boldsymbol{q}_{i-1}$. The model pulse $\boldsymbol{q}_{i-1}$ is one of the vectors: $\boldsymbol{q}_{i-1}^{(1)}$, $\boldsymbol{p}_i$ and $\boldsymbol{p}'_{i-1}$ equal to $(\boldsymbol{p}_{i-2} + \boldsymbol{p}_i)/2$. Choose the vector which minimizes the prediction error with respect to $\boldsymbol{p}_{i-1}$.
7. If the end conditions of the current segmentation mode are satisfied, end the segmentation. Otherwise increment i and continue starting from (2).

Note that the alignment is always carried out with respect to the past identified and aligned pulses.

This segmentation algorithm is used in all three modes: *start*, *continue* and *end*. In every mode all viable segmentations are tested. The modes differ in (i) the conditions the first and the last pulse must satisfy and (ii) the length of the segmentation windows. The segmentation in a particular mode is chosen if it minimizes the segmentation error ε_{seg}. The error ε_{seg} is given by

$$\varepsilon_{seg} = \varepsilon_b + \sum_{i=0}^{N-1} \mathcal{E}_p(\boldsymbol{q}_i, \boldsymbol{p}_i) + \varepsilon_e \tag{5.19}$$

where ε_b and ε_e are the errors at the beginning and at the end of the segmentation window respectively. The calculations of the errors ε_b and ε_e differ in the three segmentation modes.

Segmentation in the *Start* Mode

In the *start* mode we segment the residual of length $L_S^{(s)}$ from t_S to $t_{S'}$ with

$$t_S = t_k \quad \text{and} \quad t_{S'} = t_S + L_S \,. \tag{5.20}$$

The segmentation window $L_S^{(s)}$ should be at least the size of the frame, L_F.

The beginning of the first pulse must be within the frame boundaries. Every candidate mark $\tau_0 \in \{\tau\}$ such that

$$t_k \leq \tau_0 < t_{k+1} \tag{5.21}$$

is considered as a possible start of a series of pitch pulses. The beginning of the second pulse τ_1 (which is also the end of the first pulse) must lie within the distance of P_{min} to P_{max} from τ_0,

$$\tau_0 + P_{min} \leq \tau_1 \leq \tau_0 + P_{max} \,. \tag{5.22}$$

The beginning of every following pulse τ_i, $i>1$, must be within the limits set by P_{min}, P_{max} and the maximum allowed pitch pulse length change. We have

$$\tau_{i-1} + \max\Big(P_{min},\, (1-\delta_{\mathrm{P}\uparrow})L\boldsymbol{p}_{i-1}\Big) \;\leq\; \tau_i \;\leq\; \tau_{k_{i-1}} + \min\Big((1+\delta_{\mathrm{P}\downarrow})L\boldsymbol{p}_{i-1},\, P_{max}\Big) \tag{5.23}$$

where $\delta_{\mathrm{P}\uparrow}$ and $\delta_{\mathrm{P}\downarrow}$ specify the maximum allowed relative decrease and increase of the pitch pulse length.

We stop the segmentation at $i=N$ when $\tau_{i+1} \geq t_{S'}$. We have identified $N{+}1$ pitch pulses in which one pulse extends beyond the end of the segmentation window. The segmentation error ε_{seg} is calculated as in (5.19) with

$$\varepsilon_b = \delta_{ne}\, E\Big(\boldsymbol{r}\,[t_S, \tau_0)\Big)\,, \tag{5.24}$$

$$\varepsilon_e = E\Big(\boldsymbol{r}\,[\tau_N, t_{S'}) - \beta_N \boldsymbol{q}_N\Big) \tag{5.25}$$

where β_N is the prediction gain between vectors $\boldsymbol{q}_N$ and $\boldsymbol{p}_N$. The vector $\boldsymbol{q}_N$ is truncated to length $t_{S'}-\tau_N$. The segmentation which minimizes ε_{seg} is the *start* mode segmentation accepted for the current frame.

In the *start* frame, the boundaries of pitch pulse segments are further repositioned so that the signal energy on the pulse boundaries is minimized. The maximum allowed shift is equal to the maximum shift permitted during pulse alignment P_{align}. The repositioning is done only in the *start* frame.

Segmentation in the *End* Mode

In the end mode the beginning of the first pulse τ_0 and the beginning of the segmentation window t_S are set to the beginning of the last pulse of the previous frame. The first pulse is not subject to alignment. The candidates for the beginning of the next pulse have to satisfy the relation (5.22) if τ_0 is the beginning of the first pulse in the series, and relation (5.23) otherwise. In addition the beginning of the second pulse must be within the current frame boundaries, $t_k \leq \tau_1 \leq t_{k+1}$.

In the *end* mode the segmentation window extends only until the end of the frame,

$$t_{S'} = t_{k+1}\,. \tag{5.26}$$

The end of the last pulse τ_N must be within the segmentation window,

$$\tau_N < t_{S'}\,. \tag{5.27}$$

The segmentation error is given by (5.19) with

$$\varepsilon_b = 0 \quad \text{and} \quad \varepsilon_e = \delta_{ne}\, E\Big(\boldsymbol{r}\,[\tau_N, t_{S'})\Big)\,. \tag{5.28}$$

The segmentation which minimizes ε_{seg} is chosen as the *end* mode segmentation of the current frame.

Segmentation in the *Continue* Mode

In the *continue* mode the conditions for the beginning of the pulses and the beginning of the segmentation window t_S are the same as in the *end* mode. The end of the

segmentation window is specified by $L_S^{(c)}$ and is given by

$$t_{S'} = t_k + L_S^{(c)} \,. \tag{5.29}$$

The end of the last pulse τ_{N+1} must extend beyond the current frame, $\tau_{N+1} \geq t_{k+1}$, and the segmentation error is computed as in (5.20) with

$$\varepsilon_b = 0 \,, \tag{5.30}$$

$$\varepsilon_e = \begin{cases} \mathcal{E}_p(\boldsymbol{q}_N, \boldsymbol{p}_N) + \delta_{ne}\, E\Big(\boldsymbol{r}\,[\tau_{N+1}, t_{S'})\Big) & \text{if} \quad \tau_{N+1} < t_{S'} \\ E\Big(\boldsymbol{r}\,[\tau_N, t_{S'}) - \beta_N \boldsymbol{q}_N\Big) & \text{if} \quad \tau_{N+1} \geq t_{S'} \end{cases} \tag{5.31}$$

where β_N is the prediction gain between vectors q_N and $\boldsymbol{p}_N$ and the vector $\boldsymbol{q}_N$ is truncated to length $t_{S'} - \tau_N$. Again, the segmentation which minimizes this error is chosen as the *continue* segmentation of the frame.

Frame Error Calculation

Based on the best segmentations in the three modes, the frame errors are calculated. The frame is extended so that the frames overlap. This is done to reduce the effects of unfavorable frame positioning. Every frame is extended on each side to the closest boundary of the pitch identified in the *continue* mode, but by no more than $2L_{F_e}$. The extended frame k is specified by:

$$t'_k = \max(\tau_0^{(c)}, t_k - L_{F_e}) \,, \tag{5.32}$$

$$t'_{k+1} = \min(\tau_I^{(c)}, t_{k+1} + L_{F_e}) \tag{5.33}$$

where

$$\tau_I^{(c)} = \min_{\tau_i^{(c)} > t_{k+1}} \tau_i^{(c)} \,. \tag{5.34}$$

The error $e^{(\mathrm{n})}$ is set to the energy of the extended frame scaled by δ_{ne}. The errors $e^{(\mathrm{s})}$, $e^{(\mathrm{c})}$ and $e^{(\mathrm{e})}$ are calculated based on the corresponding segmentations (*start*, *continue*, *end*) applied to the extended frame.

5.3.4 Computational Savings

For long pitch pulses the tails of the pulses are not as well correlated as the initial, high energy part. We therefore reduce the computational complexity of our extraction method by limiting the dimension of the prediction error calculation. The dimension of the error calculated while choosing the model pitch pulse is the minimum of $L_{\boldsymbol{p}_i}$ and D_{pred}. The error dimension computed while aligning the pulses is the minimum of $L_{\boldsymbol{p}_i}$ and D_{align}. The segmentation error and the frame error, however, are always calculated with the length $L_{\boldsymbol{p}_i}$ after the length is established by aligning the next pulse.

The maximum number of segmentations which might have to be considered is still large. For the beginning of the first pulse in a series we consider N_1 candidates,

$$N_1 \leq L_F \frac{N_P}{L_P} \,. \tag{5.35}$$

For the beginning of the second pulse of a series we try N_2 candidates,

$$N_2 \leq (P_{max} - P_{min}) \frac{N_P}{L_P} \,. \tag{5.36}$$

For the pulse i, $i > 1$ there are N_i candidate positions,

$$P_{min}(\delta_{\mathrm{P}\uparrow} + \delta_{\mathrm{P}\downarrow}) \frac{N_P}{L_P} \;\leq\; N_i \;\leq\; P_{max}(\delta_{\mathrm{P}\uparrow} + \delta_{\mathrm{P}\downarrow}) \frac{N_P}{L_P} \,. \tag{5.37}$$

The total number of segmentations to be considered N_S is given by

$$N_S = \prod_{i=0}^{I-1} N_i \tag{5.38}$$

with

$$\frac{L_S}{P_{min}} \leq I \leq \frac{L_S}{P_{max}} \,, \tag{5.39}$$

where L_S is the length of the segmentation window.

Reducing by one the number of candidate positions for the pulse j eliminates N_e possible segmentations,

$$N_e = \prod_{i=j+1}^{I-1} N_i \,. \tag{5.40}$$

The smaller the j, the more the computational savings.

We reject a segmentation at pitch pulse j if

- after the alignment of the next pulse the length of the pulse j is smaller than P_{min} or larger than P_{max},
- the normalized error between the model pulse $\boldsymbol{q}_j$ and the pitch pulse $\boldsymbol{p}_j$ is larger than the threshold δ_{pe}.

The rejection of segmentations reduces the computational complexity of the implemented method significantly.

The residual signal is up-sampled by a factor of F_{ups} and the pulses are aligned with the up-sampled resolution, but for the purpose of calculating the errors the pulses are decimated to the original 8 kHz sampling rate. Note that the down-sampling does not, in general, produce the original signal because of a possible fractional shift introduced by the alignment.

In the alignment procedure, first a rough alignment is performed with the 8 kHz resolution and then a fine alignment is carried out around the rough estimate.

5.4 Coding the Pitch Pulse Positions

The pitch information is encoded once per frame. The coded parameters are: (i) a pitch pulse position τ_k, (ii) the number of pulses between this and the last coded pulse position, N_k pulses between τ_{k-1} and τ_k.

As in Section 5.3, the four types of frames are handled differently: *noise* frames – frames with no pulses; *start* frames – frames in which pulses start; *continue* frames – when pulses continue; *end* frames – frames in which pulses end.

The number of pulses between frames is coded indirectly. First, an expected number of pulses $\hat{N}_k$ is calculated, and then a correction value C_k is determined. If $\tilde{\tau}_{k-1}$ is the position quantized in the last frame, $\tilde{P}_{k-1}$ is the average pitch length of the last frame, and $\tilde{\tau}_k$ is the quantized position of the current frame, the expected number of pitch pulses in this frame is:

$$\hat{N}_k = \frac{\tilde{\tau}_k - \tilde{\tau}_{k-1}}{\tilde{P}_{k-1}} \,. \tag{5.41}$$

The correction value C_k is given by

$$C_k = \begin{cases} 0 & \text{for} \quad 0.0 \leq |\hat{N}_k - N_k| < 0.5 \\ 1 & \text{for} \quad 0.5 \leq |\hat{N}_k - N_k| < 1.0 \\ 2 & \text{for} \quad 1.0 \leq |\hat{N}_k - N_k| < 1.5 \\ 3 & \text{for} \quad 1.5 \leq |\hat{N}_k - N_k| < 2.0 \\ \vdots & \end{cases} \tag{5.42}$$

The decoder calculates the expected number of pulses $\hat{N}_k$ and, based on the coded correction value C_k, determines the number of coded pulses.

If a *continue* frame follows a *start* frame, the encoding strategy is different. In this case there is no $\tilde{P}_{k-1}$ available. For a *continue* k-th frame following a *start* frame, the number of pulses, N_k, is calculated as

$$N_k = C_k M_{\tilde{\tau}_{k-1}} + C_{k-1}\,, \tag{5.43}$$

where C_k and C_{k-1} are the coded correction values for the current and for the last frame, and $M_{\tilde{\tau}_{k-1}}$ is the maximum correction value allowed at the quantized start position $\tilde{\tau}_{k-1}$. Given $M_{\tilde{\tau}_{k-1}}$ and N_k the correction values are calculated as

$$\begin{aligned} C_{k-1} &= N_k \bmod M_{\tilde{\tau}_{k-1}} \\ C_k &= \lfloor N_k / M_{\tilde{\tau}_{k-1}} \rfloor\,. \end{aligned} \tag{5.44}$$

If an *end* frame follows a *start* frame, the coding strategy is the same as when *continue* follows *start*.

In every frame, the pitch pulse positions and the number of intermediate pulses are specified by an index to a pair of values in a quantization table. Each pair of values contains two integers: the first integer specifies the quantized pitch pulse position, the second integer is the correction value based on which the estimated number of intermediate pitch pulses is modified.

A quantization table is specified by integer pairs with each pair describing one allowable pitch pulse position. The set of first elements of the pairs determines the permitted quantized positions in the current frame. The second element is the number of correction values assigned to the position described by the pair. The number of the

codewords represented by a quantization table is equal to the sum of the correction-value numbers.

We code the pitch pulse position information with eight bits, which allows us to specify 256 codewords. Two quantization tables are used: one for a *start* frame, and one for a *continue/end* frame. In both cases one codeword is reserved for "*noise* frame" information. This reduces the number of available codewords to 255.

The quantizing tables used in the implemented coder are given in Table 5.3 and Table 5.4. With these tables the beginning of a first pulse is quantized with three-sample resolution, a pulse position in a *continue* frame is quantized with two-sample resolution, and the end of a last pulse is quantized with five-sample resolution. The coded residual is synchronized with the original at least once per frame within two samples in the *start* frame, one sample in the *continue* frame, and three samples in the *end* frame. This relaxed synchronization was found to perform very well and the semi-synchronized speech was judged to be perceptually equivalent to the original (see Section 5.8).

5.4.1 Choosing the Pitch Pulse Position to Code

There may be a number of pitch pulses in a frame and we can choose which pulse position to code. One of the strengths of our coding method is that we have this choice. Some parameters (pitch pulse shape, pulse length, pulse gain) are interpolated between the coded positions; so if we choose the position at which the change in these parameters is large we can gain a coding advantage.

Each pulse position in a frame is first tested if it is "coding valid". A pulse position at time τ is "coding valid" if the correction number resulting from coding this position is smaller than or equal to the maximum correction number allowed at this position,

$$C_k(\tau) \leq M_\tau \,. \tag{5.45}$$

In our implementation we code the position which would result in the smallest maximum change of the coded (and interpolated) pitch pulse lengths with respect to the lengths of the original pulses. The k-th frame boundaries are written as t_{F_k} and $t_{F_{k+1}}$. The last coded position τ_k is equal to t_i, and

$$\tau_k \leq t_{i+j} < t_{F_k} \qquad \text{for} \quad 0 \leq j < N_{F_k} \tag{5.46}$$

Table 5.3 Pitch quantizing table used in the *start* frame.

$(0,6)$	$(3,6)$	$(6,6)$	$(9,6)$	$(12,6)$	$(15,6)$
$(18,6)$	$(21,6)$	$(24,6)$	$(27,6)$	$(30,6)$	$(33,6)$
$(36,5)$	$(39,5)$	$(42,5)$	$(45,5)$	$(48,5)$	$(51,5)$
$(54,5)$	$(57,5)$	$(60,5)$	$(63,5)$	$(66,5)$	$(69,5)$
$(72,5)$	$(75,5)$	$(78,5)$	$(81,4)$	$(84,4)$	$(87,4)$
$(90,4)$	$(93,4)$	$(96,4)$	$(99,4)$	$(102,4)$	$(105,4)$
$(108,4)$	$(111,4)$	$(114,4)$	$(117,4)$	$(120,4)$	$(123,4)$
$(126,4)$	$(129,4)$	$(132,4)$	$(135,4)$	$(138,4)$	$(141,4)$
$(144,4)$	$(147,4)$	$(150,4)$	$(153,4)$	$(156,4)$	$(159,4)$

Table 5.4 Pitch quantizing table used in the *continue/end* frame

Continue part:

$(0,2)$	$(2,2)$	$(4,2)$	$(6,2)$	$(8,2)$	$(10,2)$	$(12,2)$	$(14,2)$
$(16,2)$	$(18,2)$	$(20,2)$	$(22,2)$	$(24,2)$	$(26,2)$	$(28,2)$	$(30,2)$
$(32,2)$	$(34,2)$	$(36,2)$	$(38,2)$	$(40,2)$	$(42,2)$	$(44,2)$	$(46,2)$
$(48,2)$	$(50,2)$	$(52,2)$	$(54,2)$	$(56,2)$	$(58,2)$	$(60,2)$	$(62,2)$
$(64,2)$	$(66,2)$	$(68,2)$	$(70,2)$	$(72,2)$	$(74,2)$	$(76,2)$	$(78,2)$
$(80,2)$	$(82,2)$	$(84,2)$	$(86,2)$	$(88,2)$	$(90,2)$	$(92,2)$	$(94,2)$
$(96,2)$	$(98,2)$	$(100,2)$	$(102,2)$	$(104,2)$	$(106,2)$	$(108,2)$	$(110,2)$
$(112,2)$	$(114,2)$	$(116,2)$	$(118,2)$	$(120,2)$	$(122,2)$	$(124,2)$	$(126,2)$
$(128,2)$	$(130,2)$	$(132,2)$	$(134,2)$	$(136,2)$	$(138,2)$	$(140,2)$	$(142,2)$
$(144,2)$	$(146,2)$	$(148,2)$	$(150,2)$	$(152,2)$	$(154,2)$	$(156,2)$	$(158,2)$

End part:

$(1,2)$	$(6,3)$	$(11,3)$	$(16,3)$	$(21,3)$	$(26,3)$	$(31,3)$	$(36,3)$
$(41,3)$	$(46,3)$	$(51,3)$	$(56,3)$	$(61,3)$	$(66,3)$	$(71,3)$	$(76,3)$
$(81,3)$	$(86,3)$	$(91,3)$	$(96,3)$	$(101,3)$	$(106,3)$	$(111,3)$	$(116,3)$
$(121,3)$	$(126,3)$	$(131,3)$	$(136,3)$	$(141,3)$	$(146,3)$	$(151,3)$	$(156,3)$

$$t_{F_k} \leq t_{i+j} < t_{F_{k+1}} \quad \text{for} \quad N_{F_k} \leq j < N_{F_{k+1}} \,. \tag{5.47}$$

The selected coded position in this frame τ_k is equal to t_{i+m}, where

$$m = \underset{N_{F_k} \leq n < N_{F_{k+1}}}{\arg\min} \; (\max_{0 \leq j < n} |P(t_{i+j}) - \tilde{P}(t_{i+j})|) \,. \tag{5.48}$$

We have found that for a *continue* frame, the maximum correction value of 1 is sufficient. For long pitch pulses there are fewer pulses for coding to choose from, but the correction number of 1 means a very large change in the pulse lengths. For shorter pulses there are more pulse positions to choose from, and therefore there are always few pulses which are "coding valid".

5.4.2 Pitch Pulse Length Interpolation

In the process of pitch pulse length interpolation, the block of $T_k = \tau_k - \tau_{k-1}$ samples is segmented into N_k pulses. The original lengths of the pulses are $P(t_{i+j})$, with $t_i = \tau_{k-1}$ and $0 \leq j < N_k$. The new pitch pulse lengths are $\tilde{P}(t_{i+j})$, $0 \leq j < N_k$. The segmentation is determined by the applied pitch interpolation technique (Section 4.1.4).

We implemented the interpolation directly on the pitch pulse lengths $P(t)$ (see Section 4.1). The linear interpolation described in Section 4.1.4 does not deal with the constraint of $\tilde{P}(t_{i+j})$ being integer. With this constraint

$$\tilde{P}(t_{i+j}) - \tilde{P}(t_{i+j-1}) \neq \text{const} \,. \tag{5.49}$$

We calculate the integer pitch pulse lengths $\tilde{P}(t_{i+j})$, $0 \leq j < N_k$, so that the *variation* in the pitch pulse length *differences* is minimized. For

$$c(j) = \tilde{P}(t_{i+j}) - \tilde{P}(t_{i+j-1}) \,, \quad 0 \leq j < N_k \,, \tag{5.50}$$

we want to minimize

$$d_c = \sum_{j=1}^{N_k - 1} | \, c(j) - c(j-1) | \,. \tag{5.51}$$

The algorithm used for segmenting the block of T_k samples into N_k pulses so that d_c is minimized is presented in Appendix A.

5.5 Coding the Gain

The total gain of the residual is coded with 5 bits, twice per frame, in the log domain. In frames with no pitch pulses, the gain is calculated every 10 ms (80 samples). The difference with respect to the last quantized gain is encoded. The coded gain is applied to the second half of the 10 ms gain sub-frame. The gain in the first half is interpolated from two adjacent coded gains. This insures that there are no abrupt changes in the gain envelope.

In frames with pitch pulses, the gain is coded pitch synchronously. For every frame, we code the gain of the pulse which ends at the encoded pulse position. The other coded gain depends on the number of pulses between the coded positions, i.e., the number of interpolated pulses. If there is an odd number of pulses to be interpolated, we code the gain of the middle pulse. If there is an even number of pulses to be interpolated, we code the average gain of the two middle pulses. If there are no pulses to be interpolated, we have two gains corresponding to the same coded pitch pulse. In this case the second gain becomes a refinement of the first one so that the gain of this pulse is more accurate. In a *start* frame this may improve the adaptability of the coder to large energy changes at voiced onsets. The gains of the pulses whose amplitudes were not coded are linearly interpolated from the gains of the adjacent pulses. The interpolation is linear in the number of pulses and not in time, which means that the interpolation coefficients do not depend on the pitch pulse lengths, but only on how many intermediate pulses are between the coded pulses.

Special care is given to frames where the pulses start and end. In a *start* frame, we code the gain of the noise prior to the first pulse and the gain of the first pulse. In an *end* frame, we code the gain of the last pulse and the gain of the noise following it. The coding is still differential but a slightly coarser quantization table is used to enable a faster build-up of the energy at voiced onsets, and a quicker die-off when the voiced region ends. There is no gain interpolation between a gain corresponding to a pitch pulse and a gain corresponding to noise.

In calculation of the noise, a window of at least 10 ms is used. This means that if the pulses start within 10 ms of the frame beginning or end within 10 ms of the frame end, the gain of the noise will be calculated using some samples from the last frame or from the next frame. This was found necessary for obtaining a smooth gain envelope.

5.6 Coding the Shape of the Pitch Pulses

The pitch pulse shapes of the underlying pitch pulses are coded. One of two types of coding is used: (i) differential coding with respect to the predicted pitch pulse shape, or (ii) direct coding of the underlying pitch pulse. For the two coding types, two different codebooks are used; one bit specifies the coding method selected. The coding is carried out in the frequency domain and one pulse per frame is coded, and the intermediate pulses are interpolated from coded pulses.

Predicting the Underlying Pitch Pulse

At present, the predicted underlying pitch pulse is simply the last coded pulse. Reliable prediction of the current underlying pitch pulse from the past coded pulses is one of the suggested topics for future work.

Estimation of the Underlying Pulses

The extracted pitch pulses are normalized and the underlying pitch pulses are estimated with the algorithm described in Section 3.3.4. The weighted average algorithm is used with the error weight $\omega = 0.7$. The algorithm is initialized with the underlying pulses equal to the extracted pulses. In a *start* frame the previous underlying vector $\boldsymbol{v_0}$ is unknown; in a *continue* and *end* frame the previous underlying pulse is set to the pulse which was coded in the last frame. The pulses used in the estimation procedure are extracted from a block of the LP residual starting at the pulse position coded in the last frame, and ending at the end of the current frame.

Coding Pitch Pulses

First, the underlying pulses are transformed into the frequency domain with a fixed length DFT using an FFT algorithm. Then, a linear fit is applied to the set of the underlying pulses. The linear fit is applied to the pulses so that the intermediate pulses are reconstructed (via linear interpolation) with minimum error. The weighted mean square error linear fit is described in Appendix B. The pulse which ends at the encoded pitch pulse position is coded. In the coding of the pitch pulses we use one of two codebooks. The first codebook $\mathcal{CB}_p$ contains sample spectra of underlying pitch

pulses of various speakers. The second codebook $\mathcal{CB}_D$ contains differences between pulse spectra.

Training of the Codebooks

In training the codebook of pitch pulse spectra $\mathcal{CB}_p$, we used pitch pulses of male speakers with pulse lengths larger than 5 ms. A spectrum of an underlying pulse was included in the selection set if its normalized correlation with respect to one of its neighbours was larger than 0.8. The selection set was first used as a training set for the GLA algorithm to compute an initial set of codebook vectors. The selection set was searched for the pulses which maximize the correlation with respect to the initial codebook vectors. The codebook $\mathcal{CB}_p$ was populated with these pulses from the selection set. This was done so that the codebook $\mathcal{CB}_p$ would contain true spectra of estimated underlying pulses and not an average between them. In particular we wanted to avoid averaging underlying pulses coming from different voiced segments.

The second codebook $\mathcal{CB}_D$ contains differences between spectra. The difference between the spectra of two underlying pulses was included in the training set if the corresponding extracted pulses were less than 20 ms apart and the normalized correlation between them was larger than 0.5. Note that the criteria were applied to the extracted pulses but the difference was taken between the spectra of the underlying pulses. The $\mathcal{CB}_D$ codebook was also trained with the GLA algorithm.

Searching the Codebooks

In the coding procedure first the codebook $\mathcal{CB}_p$ is searched for the entry which minimizes weighted mean square error with respect to the target spectrum. The weighted mean square error is specified by two fixed weights. The weight $w_a(f)$ emphasizes the importance of the frequencies around 1 kHz and is given by

$$w_a(f) = \begin{cases} 0.1 & \text{for} \quad 0 < |f| \leq 300 \\ 0.5 & \text{for} \quad 300 < |f| \leq 600 \\ 1.0 & \text{for} \quad 600 < |f| \leq 1200 \\ 0.4 & \text{for} \quad 1200 < |f| \leq 2400 \\ 0.1 & \text{otherwise.} \end{cases} \tag{5.52}$$

The weight $w_{ri}(f)$ is used to deemphasize the phase of higher frequencies and is specified by

$$w_{ri}(f) = \begin{cases} 0.4 & \text{for} \quad 0 < |f| \leq 300 \\ 0.6 & \text{for} \quad 300 < |f| \leq 600 \\ 0.4 & \text{for} \quad 600 < |f| \leq 1200 \\ 0.2 & \text{for} \quad 1200 < |f| \leq 2400 \\ 0.0 & \text{otherwise.} \end{cases} \tag{5.53}$$

The weighted error between two spectra is calculated as a sum of the weighted errors between the real parts, the imaginary parts and the amplitudes of the spectra,

$$E_{ri}(f) = \Big(X_r(f) - Y_r(f)\Big)^2 + \Big(X_i(f) - Y_i(f)\Big)^2 , \tag{5.54}$$

$$E_a(f) = \Big(X_a(f) - Y_a(f)\Big)^2 , \tag{5.55}$$

$$E_t = \sum_f w_a(f)\Big[w_{ri}(f)E_{ri}(f) + \Big(1 - w_{ri}(f)\Big)E_a(f)\Big] \tag{5.56}$$

where X_r, X_i, X_a, Y_r, Y_i and Y_a denote the real part, the imaginary part and the amplitude of the spectra X and Y respectively. We can also write

$$E_t = \sum_f \Big(w'(f)E_{ri}(f) + w''(f)E_a(f)\Big) \tag{5.57}$$

where

$$w'(f) = w_a(f)w_{ri}(f) \quad \text{and} \quad w''(f) = w_a(f) - w'(f)\,. \tag{5.58}$$

The codebook $\mathcal{CB}_D$ is searched for the entry which minimizes the weighted mean square error with respect to the difference between the target spectrum and the last coded spectrum.

The codebook $\mathcal{CB}_p$ must be able to quickly update the pitch pulse at the start of a voiced segment or within the voiced segment when the pitch pulse waveshape changes abruptly (which we have observed). Within regions where the pitch pulse changes slowly, the update of the pulse spectrum should be supplied by the codebook $\mathcal{CB}_D$. We observed excessive periodicity when the same entry of codebook $\mathcal{CB}_p$ was repeatedly chosen as the coded spectrum. To eliminate that problem, the codebook which is used to code the current pulse spectrum is chosen as follows.

If the minimum weighted error of $\mathcal{CB}_D$ is smaller than the minimum weighted error of $\mathcal{CB}_p$, the codebook $\mathcal{CB}_D$ is used. In this case the coded spectrum is the sum of the last spectrum and the selected entry of the codebook $\mathcal{CB}_D$. If the minimum weighted error of $\mathcal{CB}_p$ is smaller, we calculate the weighted error between the selected entry of $\mathcal{CB}_p$ and the last coded spectrum. The codebook $\mathcal{CB}_p$ is used if this error is larger than the error between the spectrum coded with $\mathcal{CB}_D$ and the last spectrum. To make the above more clear, we write the last coded spectrum as $\tilde{X}_{i-1}$, the target spectrum as X_i, and the current coded spectrum as $\tilde{X}_i$, the selected entry of the codebook $\mathcal{CB}_p$ as Y_p, the selected entry of the codebook $\mathcal{CB}_D$ as Y_D and the weighted error between spectra X and Y as $E_\mathrm{w}(X, Y)$. We choose the current coded spectrum $\tilde{X}_i = Y_\mathrm{p}$ if

$$E_\mathrm{w}(Y_\mathrm{p}, X_i) < E_\mathrm{w}(\tilde{X}_{i-1} + Y_\mathrm{D}, X_i) \tag{5.59}$$

$$\text{and} \quad E_\mathrm{w}(Y_\mathrm{p}, \tilde{X}_{i-1}) > E_\mathrm{w}(\tilde{X}_{i-1} + Y_\mathrm{D}, \tilde{X}_{i-1})\,. \tag{5.60}$$

Otherwise the coded spectrum is chosen as $\tilde{X}_i = \tilde{X}_{i-1} + Y_\mathrm{D}$.

Pitch Pulse Interpolation

The pitch pulses are interpolated in the spectral domain. We use linear interpolation on complex spectra. The interpolation is linear in the number of pulses and not in time, i.e., a fixed number of pulses will have the same interpolation coefficients regardless of the interval between pulse positions.

After the interpolation and the inverse transform, the time-domain voiced LP excitation is formed. The shape-interpolated pulses are placed at the pulse positions determined by the pulse-length interpolation.

5.7 Coding the Noise Component

The noise component of the LP excitation is coded with a CELP-like procedure via analysis-by-synthesis with the error calculated in the perceptually-weighted speech domain.

The perceptually-weighted speech in the sub-frame l is created by filtering the

LP residual with the perceptually-weighted filter

$$H(z) = \frac{1}{\hat{A}(z)} \frac{A(\gamma_1 z)}{A(\gamma_2 z)}, \tag{5.61}$$

where $A(z)$ and $\hat{A}(z)$ are, respectively, the LP coefficients and the quantized LP coefficients obtained as specified in Section 5.2. The parameters γ_1 and γ_2 regulate the strength of the perceptual weighting. We use γ_1 and γ_2 fixed at 1.0 and 0.8. The impulse response of the filter $H(z)$ is written as $h(n)$.

It is useful to decompose the output of the filter into two components: one which depends on the past inputs and the other which depends on the samples of the current sub-frame. When the input of the filter for the current sub-frame is a zero vector, the output of the filter is the zero input response. The zero input response depends on the past inputs to the filter[†]. When the memory of a filter is set to zero, the output of the filter is the zero-state response. The total output of the filter in the current sub-frame is a superposition of the zero input and the zero-state responses. The filter output depends therefore, at a given time, on (i) the filter coefficients, (ii) the past inputs to the filter, and (iii) the current input to the filter. We write the reconstructed weighted speech as

$$\mathcal{H}_{\langle x \rangle}(\boldsymbol{x}^{(l)}). \tag{5.62}$$

This is the output of the filter with (i) the coefficients specified by $H(z)$, (ii) the past inputs equal to the past samples of the LP excitation x, and (iii) the current input equal to the vector of samples of the current excitation sub-frame $\boldsymbol{x}^{(l)}$. With the symbols $\mathcal{H}_{ZI}$ and $\mathcal{H}_{ZS}$ denoting, respectively, the zero input and the zero-state response of the filter $\mathcal{H}$, we have

$$\mathcal{H}_{\langle x \rangle}(\boldsymbol{x}^{(l)}) = \mathcal{H}_{ZI}{}^{(l)}_{\langle x \rangle} + \mathcal{H}_{ZS}(\boldsymbol{x}^{(l)}). \tag{5.63}$$

The zero-state response of a filter is equal to the convolution of the impulse response of the filter with the input of the filter. In a matrix notation this convolution can be written as

$$\mathcal{H}_{ZS}(\boldsymbol{x}^{(l)}) = \boldsymbol{H}\, \boldsymbol{x}^{(l)}, \tag{5.64}$$

[†]In perceptual weighting the filter coefficients change from frame-to-frame (and possibly from sub-frame to sub-frame). As a result the zero input response also depends on the past coefficients of the filter. Our notation does not show this dependency.

where $\boldsymbol{H}$ is the impulse response matrix specified as

$$\boldsymbol{H} = \begin{bmatrix} h(0) & h(1) & h(2) & \cdots & h(N-1) \\ 0 & h(0) & h(1) & \cdots & h(N-2) \\ 0 & 0 & h(0) & \cdots & h(N-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & h(0) \end{bmatrix} . \tag{5.65}$$

The reconstructed, weighted speech the sub-frame l is given by:

$$\hat{\boldsymbol{s}}_w{}^{(l)} = \mathcal{H}_{\langle x \rangle}(\boldsymbol{x}^{(l)}) \tag{5.66}$$

$$= \mathcal{H}_{ZI}{}^{(l)}_{\langle x \rangle} + \mathcal{H}_{ZS}(\boldsymbol{x}^{(l)}) . \tag{5.67}$$

With r denoting the LP residual, the perceptually-weighted speech of the sub-frame l is given by

$$\boldsymbol{s}_w{}^{(l)} = \mathcal{H}_{\langle r \rangle}(\boldsymbol{r}^{(l)}) \tag{5.68}$$

$$= \mathcal{H}_{ZI}{}^{(l)}_{\langle r \rangle} + \mathcal{H}_{ZS}(\boldsymbol{r}^{(l)}) . \tag{5.69}$$

The error between the perceptually weighted speech and the reconstructed, weighted speech is

$$\boldsymbol{s}_w{}^{(l)} - \hat{\boldsymbol{s}}_w{}^{(l)} = \mathcal{H}_{ZI}{}^{(l)}_{\langle r \rangle} - \mathcal{H}_{ZI}{}^{(l)}_{\langle x \rangle} + \mathcal{H}_{ZS}(\boldsymbol{r}^{(l)}) - \mathcal{H}_{ZS}(\boldsymbol{x}^{(l)}) \tag{5.70}$$

$$= \mathcal{H}_{ZI}{}^{(l)}_{\langle r-x \rangle} + \mathcal{H}_{ZS}(\boldsymbol{r}^{(l)}) - \mathcal{H}_{ZS}(\boldsymbol{x}^{(l)}) \tag{5.71}$$

$$= \mathcal{H}_{\langle r-x \rangle}(\boldsymbol{r}^{(l)}) - \mathcal{H}_{ZS}(\boldsymbol{x}^{(l)}) . \tag{5.72}$$

The notation $\mathcal{H}_{\langle r-x \rangle}(\boldsymbol{r}^{(l)})$ represents the output for a filter with (i) the coefficients specified by $H(z)$, (ii) the past inputs equal to the difference between the past LP residual r and the past coded excitation x, (iii) the current input equal to the vector $\boldsymbol{r}^{(l)}$. We want to minimize

$$\boldsymbol{s}_w{}^{(l)} - \hat{\boldsymbol{s}}_w{}^{(l)} , \tag{5.73}$$

which is equivalent to finding $\boldsymbol{x}^{(l)}$ such that

$$\mathcal{H}_{\langle r-x \rangle}(\boldsymbol{r}^{(l)}) - \mathcal{H}_{ZS}(\boldsymbol{x}^{(l)}) \tag{5.74}$$

is minimized.

The Noise Codebooks

The noise is coded on the basis of 40-sample long sub-frames and so the dimension of the noise codebook vectors is 40. The codebook vectors were populated with random independent Gaussian numbers, one value per five vector elements. The noise codebook entries are therefore sparse vectors with eight non-zero elements each. The positions of the non-zero elements are chosen randomly (uniform-distribution random selection of one element out of five). The absolute gain of the non-zero elements is bounded by 0.5 and 1.5.

For frames with no pitch pulses, the noise is coded with 8 bits per sub-frame and so the size of the noise codebook for those frames is 256. For the frames with pitch pulses, the noise is coded with only 4 bits per sub-frame; so the size of the noise codebook is only 16. With such a small codebook, it is possible that the same codebook entry might be chosen in consecutive sub-frames making the noise contribution periodic. Therefore, in coding the pitch pulse noise, we cycle over four codebooks of size 16 (we use 64 vectors of the 256-vector codebook).

Coding

The noise component is added to the voiced part of the excitation. The voiced contribution is created, as described in the previous sections, by placing the pulse-shape-interpolated underlying pulses at pulse-length-interpolated positions. We have

$$\boldsymbol{x}^{(l)} = \boldsymbol{x}_p{}^{(l)} + \boldsymbol{x}_n{}^{(l)}\,, \tag{5.75}$$

where $\boldsymbol{x}_p{}^{(l)}$ and $\boldsymbol{x}_n{}^{(l)}$ denote, respectively, the pitch pulse and the noise contributions to the excitation $\boldsymbol{x}^{(l)}$. Now

$$\mathcal{H}_{ZS}(\boldsymbol{x}^{(l)}) = \mathcal{H}_{ZS}(\boldsymbol{x}_p{}^{(l)}) + \mathcal{H}_{ZS}(\boldsymbol{x}_n{}^{(l)})\,. \tag{5.76}$$

Based on (5.74) and (5.76), the target vector for the noise contribution to the excitation of the sub-frame l is

$$\boldsymbol{n}_t{}^{(l)} = \mathcal{H}_{\langle r-x\rangle}(\boldsymbol{r}^{(l)}) - \mathcal{H}_{ZS}(\boldsymbol{x}_p{}^{(l)})\,. \tag{5.77}$$

The target vector is specified in the perceptually weighted speech domain and corresponds to the vector $\mathcal{H}_{ZS}(\boldsymbol{x}_n{}^{(l)})$. We calculate $\boldsymbol{n}_t{}^{(l)}$ and then search the noise codebook filtered with $\mathcal{H}_{ZS}(.)$ for the best mean-square-error match. Filtering a vector with $\mathcal{H}_{ZS}(.)$ is equivalent to multiplying the vector with the impulse response matrix $\boldsymbol{H}$ specified in (5.65).

The relative ratio between the noise contribution and the pitch pulse contribution ($\hat{r}_{n/p}$) is coded differentially with 3 bits. The update rate of the gain ratio and of the total gain are the same (every 10 ms).

5.8 Testing and Remarks

In the PPE coder only approximate time synchrony with the original signal is maintained. The relaxed synchrony renders the SNR measurement between the original and the coded speech inappropriate as a fidelity criterion. Subjective testing is necessary to evaluate the performance of the coder.

The quality of the coded speech has been assessed in informal listening tests. The testing was performed through A-B comparison tests[†] with respect to the original recording and with respect to speech coded with the G.729 coder[‡]. Since we did not use a post-filter in the PPE coder, to make the comparison fair, we used G.729 without post-filtering. The tests were performed on sentences from outside the database used for training the LSF and the pitch pulse shape codebooks. The testing was done over headphones and the listeners were considered untrained.

The tests were performed with respect to every coded parameter. After coding the LP parameters, the pitch pulse positions and the gain (the pitch pulse shapes uncoded) the reconstructed speech was judged perceptually equivalent to the original, i.e., the listeners could not tell the difference between the original and reconstructed signals. When the coding of the noise component was added, audible differences between the original and the reconstructed speech appeared. We then switched to A-B comparisons with respect to the G.729 coder. At this point (coded: the LP parameters, the pitch pulse positions, the gain, the noise component), the PPE coded

[†]In A-B testing the listener is asked to identify the better quality recording out of two consecutively played audio files presented in random order.

[‡]The G.729 coder is the ITU-T toll-quality 8 kb/s standard. The MOS score of the G.729 coder has been assessed at near 4.0.

speech was consistently better than speech coded with the G.729 coder.

To further verify our coding of the noise contribution, we used the G.729 Algebraic Structure Codebook in the place of our noise codebook. The reconstructed speech sounded slightly different but was neither better nor worse than the one coded with our noise codebook. The PPE reconstructed speech was still consistently preferred to the one coded with G.729. The Algebraic Structure Codebook has the advantage of very low computational complexity and we were pleased to confirm that it could be used in the PPE coder.

The G.729 took upper hand after the pitch pulse shapes were coded. The distortion introduced by the PPE was often difficult to identify, particularly when the audio files were played over the speakers. Over the headphones, however, speech coded with the G.729 coder was, most of the time, judged better.

Unfortunately, time did not permit more extensive or more formal testing of the implemented coder. From the tests performed, however, it is evident that the coding of the underlying pitch pulses requires more attention. A strong indication of insufficient pitch pulse codebook training is the variability introduced to the coder performance when coding of the pitch pulses is added – the quality of some coded sentences was clearly better than other sentences also from outside the training set. We did not have an access to a sufficiently large database to train the pitch pulse codebooks properly. Ideally, we would like every pulse in the pulse-shape codebook to be extracted from a different voiced region. Yet a one-minute-long speech utterance consists of, typically, only up to six voiced regions. Better coding of the underlying-pitch-pulse shapes is the main area for future research.

The Delay and Complexity of the Coder

The algorithmic delay of the implemented coder is dictated by (i) the pitch extraction algorithm, (ii) the pitch pulse position coding. In the described implementation the look-ahead of the pitch pulse extraction algorithm is equal to one frame (20 ms) (see page 90). The pitch pulse position coding introduces an extra coding delay of up to one frame, depending on which pitch pulse position was chosen for coding in the last frame. When the coded pulse position in the last frame is near the frame beginning, the coding delay introduced by the pulse position coding could be up to 20 ms. The total algorithmic delay of the implemented coder is up to 60 ms.

The coder has been implemented in the C language in the UNIX environment and, to facilitate the testing of various aspects of the coding procedure, it is distributed over a number of programs. Particularly in the early development stages we needed a full control over every parameter involved in the coding algorithm. Therefore, at present, the coder is not optimized for an optimum computational performance. We estimate, however, that the coder can be implemented in real time on a single fixed-point DSP chip performing less than 40 MIPS (a typical high-performance DSP chip), based on the fact that coders of similar computational complexity have been shown to be implementable under such restrictions (see for example Kleijn *et al.* 1996).

Chapter 6

Final Remarks, Contributions and Future Work

6.1 Summary of Our Work

In Chapter 1 the problem of speech coding was introduced. We provided an overview of a number of existing coders which are considered the state-of-the-art for speech compression at low bit rates. The scope and the objectives of our research were outlined; our goal was to represent narrowband-limited speech (200 Hz–3.4 kHz) with a bit stream of about 4 kb/s with reconstructed speech close to or equivalent to toll quality.

In Chapter 2 the principles of linear prediction coding were presented and the problem of modelling the LP excitation was examined. It was pointed out that, in linear predictive coding, the poor modelling of the voiced LP excitation is, to a large extent, responsible for degradation of the quality of the reconstructed speech.

The analysis-by-synthesis technique and CELP coding were described. We analyzed a number of improvements and modifications which have been proposed to better the representation of the LP excitation in the context of the CELP coder. Those included: harmonic noise weighting, constrained excitation, pitch synchronous innovation, comb filtering, and pitch sharpening. The paradigm of generalized analysis-by-synthesis coding was introduced.

We classified the LP coders according to the analysis block length and analysis rate used. The CELP coder is an example of a coder with fixed-rate, fixed-block-length

analysis. The analysis performed in a WI coder is also fixed-rate but the analysis block lengths are pitch synchronous.

Also in Chapter 2, the coding technique was outlined. In the PPE model the LP excitation is represented as a series of evolving pitch pulses obscured by noise. The pitch pulse-related analysis is pitch synchronous in the block lengths and the analysis rate. The coding of the noise contribution is based on generalized analysis-by-synthesis coding.

In Chapter 3 we formulated the principles of the PPE model in more detail. The LP excitation is modelled as a series of underlying pulses buried in noise; the advantage of this approach is that we can model individually every pitch pulse and effectively separate the periodic (voiced) component and the noise (unvoiced) component of the excitation. The periodic part is identified via estimation of underlying pitch pulses based on noisy pulses extracted from the LP residual. The noise contribution is coded using a generalized analysis-by-synthesis procedure.

In this chapter we set demanding goals for the pitch pulse extraction algorithm. The algorithm should identify individual pitch pulses in a way that the error between the underlying pitch pulses and the noisy pulses is minimized. At this point the estimation of the underlying pulses was simplified — we chose a pulse from a set of model pulses. With similar consecutive model pulses, the effect of the pulse segmentation is that the extracted noisy pulses are properly aligned.

A number of estimation methods to reliably identify the underlying pitch pulses from the LP residual noisy observation were investigated. The described methods were:

- linear filtering with fixed coefficients,
- maximum ratio combining (linear filtering with adaptive coefficients),
- error minimization between an underlying pulse and a number of noisy pulses (also linear filtering with adaptive coefficients),
- an algorithm which minimizes the sum of weighted errors (i) between the consecutive underlying pulses, (ii) between the underlying and the noisy pulses.

It was concluded that the last method, the error minimization algorithm, is the most effective. In the subsequent estimation of the underlying pulses we used the weighted-average version of the algorithm (the weighted-average algorithm is computationally

less expensive than the SVD algorithm and both algorithms are equally effective for the value of the error weight used).

In Chapter 4 the pitch interpolation methods used in other coders were investigated. We argued against time-warping and supported our view with examples based on the LP residual. Separate interpolations of the pitch pulse length and the pitch pulse waveshapes were proposed. The separate interpolation improves the control over the evolution of the pitch pulse characteristics. We formulated the pitch pulse-length interpolation in terms of (i) the pulse length, (ii) the fundamental frequency, and (iii) the instantaneous pitch period. Finally, the spectral interpolation was explained based on the interpolation used in WI and TFI. The PPE coder uses spectral interpolation to interpolate the pitch pulse waveshapes.

In Chapter 5 an implementation of a PPE coder was presented. In the implementation, the practical aspects of some of the key units of the PPE coder were developed. In particular, we designed a robust pitch pulse extractor which satisfies the tough requirements set by the model. A very low-rate pitch pulse position encoding scheme was devised in which the reconstructed signal maintains only a rough time synchrony with the original signal. It was verified that the relaxed time synchrony provides reconstructed speech equivalent to the original.

Also in this chapter, the results of informal listening tests of the implemented PPE coder were discussed. The coder was tested with respect to the original recording and the speech coded with G.729 (toll quality speech at 8 kb/s). The coding of LP coefficients, the pitch pulse positions and the gain provided speech quality equivalent to the original. After coding of the noise component (the pitch pulses uncoded), the PPE reconstructed signal was still better than speech coded with G.729. Only after coding the shapes of the pitch pulses (speech fully coded), did the quality of the reconstructed signal seem to be, in some cases, below the speech processed with G.729. We concluded that the codebooks and the weighted error criterion used for coding the pitch pulses still need more attention.

6.2 PPE Coding Versus WI Coding

The PPE model is akin in many ways to WI coding. As a part of the final remarks we would like to compare the two coding techniques. Both methods (i) use LP analysis,

(ii) extract individual pitch cycles, (iii) separate the noisy part and the slowly evolving part of the pitch waveform, (iv) encode the pitch information and the pitch cycle waveshape infrequently, and (v) use interpolation to reconstruct intermediate pulse waveforms. The main differences between the implemented PPE coder and a WI coder can be summarized as follows.

- *PPE*: The pitch pulses are identified based on the error between the model pulses and the LP residual.
  WI: The pitch period is estimated from the LP residual based on the autocorrelation function calculated from the residual.
- *PPE*: The pitch pulses are extracted pitch synchronously, one pitch pulse per pitch period. The rate of extraction depends on the pitch pulse lengths.
  WI: The characteristic waveforms are extracted with fixed rate. The lengths of the waveforms are based on the estimated pitch period but the extraction is pitch asynchronous.
- *PPE*: Every sample is used in one and only one pitch pulse. The case in which the pulses overlap and a new pulse is superimposed on the tail of the old pulse may be considered but so far has not been used.
  WI: With higher extraction rates the characteristic waveforms overlap, with lower extraction rates some of the residual samples are not included in any of the waveforms.
- *PPE*: The position of a pitch pulse and number of pitch pulses between the pulse positions is coded and transmitted.
  WI: The pitch period is coded and transmitted.
- *PPE*: The frequency transformation is done with fixed dimension DFT (the FFT algorithm is used).
  WI: The transformation into the frequency domain is performed with variable dimension DFT.
- *PPE*: The alignment is incorporated into the pitch pulse extraction procedure. The extraction of the pulses is such that the pulses are aligned for maximum correlation. There is no "circular shift" of the waveforms.
  WI: Characteristic waveforms are aligned using periodic extension of the extracted pitch pulses. The waveforms are extracted pitch asynchronously and

then undergo a "circular shift" which brings them into alignment. As a result of the alignment, a pulse represented by a characteristic waveform might have its front after its tail.

- *PPE*: The underlying pitch pulses and the pitch pulse noise are estimated via adaptive filtering or other methods which minimize a specified weighted error criterion. At present, we estimate underlying pulses in the time domain but the analysis in the frequency domain is also possible.
  WI: A slowly evolving waveform (SEW) and a rapidly evolving waveform (REW) are obtained by linear filtering of the characteristic waveforms. The linear filter has fixed coefficients.

- *PPE*: There is a separate interpolation of the pitch pulse lengths and of the pitch pulse waveshapes. Time-warping of the residual is avoided and approximate synchrony with the original is maintained.
  WI: The spectral interpolation results in time warping of the residual and the time synchrony with the original is not maintained. In long voiced sections, the synthesized waveform may gain or lose pulses.

Speech coders are usually divided into waveform coders and parametric coders. The difference has become blurred over the years with many coders resisting clear classification into one group or the other. In general, coders which reconstruct a signal which converges to the original with decreasing quantization error are called waveform coders. The coders whose reconstructed signal does not converge to the original are called parametric coders (Kleijn and Paliwal 1995). The LP analysis-by-synthesis coders are classified as waveform coders while sinusoidal coders and waveform interpolation coders are classified as parametric coders. The PPE coder can in fact be seen either as a waveform coder or as a parametric coder depending on the bit rate used in the encoding of the pitch information; more specifically it depends on how the pitch length is interpolated[†]. Provided the bit rate for the pitch information is high enough so that the pitch pulse lengths do not need to be interpolated (about 44 bits per 160 samples) the PPE model leads to a waveform

[†]In most of the coders in which the pitch interpolation is used, the interpolation mechanism is incorporated into the coder structure in such a way that, even for a high bit rate coding, the original and the reconstructed signals do not converge. In particular they do not converge in the coders which use time warping (e.g., in WI and in sinusoidal coding).

coder. When the number of bits for the pitch information is lower as in the case of the implemented coder (we use 8 bits per 160 sample frame), the model results in a parametric coder.

6.3 Our Contributions

In this work we developed a new speech coding model which was designed for appropriate representation of the LP excitation, particularly during the quasi-periodic voiced segments. The model combines a number of different coding techniques which we described and analyzed throughout the thesis. The proposed method also offers new solutions with respect to the problems identified in other speech coding systems.

We developed estimation techniques, which identify underlying pitch pulses in a noisy observation based on noise error minimization. A new pitch pulse extraction algorithm was implemented to satisfy the demanding requirements of the PPE model. Our pitch extraction method combines (i) pitch period estimation, (ii) pitch pulse extraction, and (iii) waveform alignment.

We suggested and implemented coding of the pitch information based on the positions of the pitch pulses. With this encoding we maintain the low bit-rate pitch coding rate of the WI coder and the time synchrony of a sinusoidal coder.

In the PPE coder the time and the frequency domain analysis are combined in a unique way. In particular, we proposed and implemented separate interpolations on the pitch pulse lengths (time domain) and the pitch pulse waveshapes (frequency domain). We also incorporated into the PPE coder the generalized analysis-by-synthesis paradigm. The noise component of the LP excitation is coded in the time domain using analysis-by-synthesis with respect to the modified (through pitch pulse-length interpolation) speech signal.

We implemented a 4 kb/s speech coder which produces very high quality coded speech. In informal listening tests the coder was judged to provide toll quality reconstructed speech when all the parameters were coded except for the coding of the underlying pitch pulse shapes. At present, quantizing the pitch pulses introduces slight distortion which we believe can be reduced by better training of the codebooks and by improving the weighted error criterion.

6.4 Claims of Originality

In this thesis, we have developed a speech coder based on several new concepts. The novel aspects include:

1. Modelling the LP excitation as a series of underlying pitch (glottal) pulses obscured by noise. Both the noise and the underlying pulses are present in the excitation simultaneously (in varying degrees).

2. An algorithm for robust identification of pitch pulse boundaries. Segmentation of the LP residual into noisy pitch pulses is based on an error criterion with respect to a set of model pulses.

3. Estimating the underlying pitch pulses based on the error between the underlying pulses and the noisy pulses and the error between the consecutive underlying pitch pulses. This approach is better than simple filtering of the pulses as it takes into account the evolution of the underlying pulse shapes.

4. Separate interpolation of the pitch pulse shapes and of the pitch pulse lengths. In our interpolation of the pulse shapes, time warping of the pulses is avoided. In the interpolation of the pitch pulse lengths, we determine the position of every pulse.

5. The encoding of the pitch information and the reconstruction of the coded speech with relaxed time synchrony with respect to the original signal.

6.5 Future Work

We demonstrated that the proposed model is capable of achieving near toll quality speech coding at rates around 4 kb/s. The implemented coder, however, was built in an experimental setting. The PPE model provides a powerful framework in which many, so far independent, analysis blocks may be integrated providing for a possibility of more optimal performance. The potential of the PPE model has not yet been fully exploited within the presented implementation.

The prediction of the underlying pitch pulse at the receiver is simply the last encoded pitch pulse. The problem of reliable prediction was not addressed in this

thesis although it was briefly investigated. The problem of prediction of the current underlying pitch pulse from the past pulses is worth pursuing.

The estimation of the underlying pitch pulses is executed in the time domain. The developed estimation algorithm can be, however, applied to the spectra of the extracted pulses. Differences between the estimation in the time domain and in the frequency domain could be further explored.

More research should be done to enhance the coding of the underlying pitch pulses. This might include a large database training of the pitch pulse codebooks, and more careful design of the weighted error criterion used in selecting the codebook entries. In our tests we experienced an excellent performance of the coder on some test sentences (both male and female) and poorer performance in some other cases.

A large number of computational improvements are possible to reduce the present complexity of the coder. In particular, the pitch pulse extraction could be further simplified and the handling of the sub-sample resolution could be made more efficient.

Combining the LP analysis with the extraction and estimation of the underlying pitch pulses is yet another area of possible research. Initial experiments conducted in this direction are very promising (Zad-Issa and Kabal 1997).

The coder should also be examined with reference to background acoustic noise and bit sensitivity to transmission errors. Possible modifications to make the coder more robust could be investigated.

Most of the coders use pre- and post-filtering to enhance the perceptual quality of the reconstructed signal. Often the filters are designed for the specific implemented method and use the coded parameters as guidance to the strength and type of required filtering. A PPE-specific pre- and post-filter for the PPE coder could be designed as a possible extension of our work.

The PPE model was explained and implemented in the setting of pitch synchronous analysis. Some of the ideas developed in this thesis can be used however in conjunction with many other coding methods. In particular the suggested pitch pulse estimation techniques can be used in the context of WI to obtain the slowly evolving waveforms and in the context of CELP to obtain more perceptually relevant adaptive codebooks.

A form of the PPE extraction algorithm could be used as a speech pre-processing unit in a CELP coder in order to eliminate fractional-pitch coding. The pulse positions

would not be interpolated but only rounded to the nearest integer value (to one-sample resolution).

We believe that the PPE paradigm of coding accurately reflects speech production with the parameterization appropriate for generating high quality speech at very low bit rates. We hope that work will continue to evolve the PPE technique into a robust, low-complexity/high-quality speech coder.

Appendix A

The Pitch Pulse Length Interpolation Algorithm

The pitch pulse length interpolation problem is formulated as follows:

> Given the last pitch pulse length, segment the block of T samples into N pitch pulses so that the sum of differences between consecutive pulse lengths is minimized.

```
/* Input:
     lastP - the last pitch pulse length
      T    - number of samples between the pitch pulse position
             coded in the last frame and the pitch pulse position
             coded in the current frame
      N    - number of pulses coded in this frame (number of
             pulses in T samples)
   Output:
     PLen  - the lengths of the N pulses coded in this frame
*/

void
ppSegment (lastP, T, N, PLen)

  int lastP, T, N, PLen[] ;

{
  int D, c [MAX_NO_OF_PULSES], i, k, n ;

  /* If the length of the last pitch pulse is not available, set its
     value to the average pitch pulse length of the current frame
   */
  if ( lastP == 0 )
    lastP = (int) round ((float) T/N) ;
```

```
  /* The number of samples which would have to be added (or taken
     away) from N pulses of length lastP
   */
  D = T - round (N*lastP) ;

  if ( D == 0 ) {
    for ( i=0 ; i<N ; i++ )
      PLen[i] = lastP ;
  }
  else {
    /* Calculate the differences between the lengths of consecutive
       pitch pulses. The sum of these differences is minimized
     */
    i = (N+1)*N/2 ;
    n = abs(D)/i ;          /* integer division */
    k = abs(D)%i ;
    for ( i=0 ; i<N ; i++ ) {
      c[i] = n ;
      if ( k >= N-i ) {
        c[i]++ ;
        k -= N-i ;
      }
    }

    /* Calculate the lengths of the coded pulses */
    if ( D > 0 ) {
      PLen[0] = lastP + c[0] ;
      for ( i=1 ; i<N ; i++ )
        PLen[i] = PLen[i-1] + c[i] ;
    }
    else {
      PLen[0] = lastP - c[0] ;
      for ( i=1 ; i<N ; i++ )
        PLen[i] = PLen[i-1] - c[i] ;
    }
  }
}
```

Appendix B

Weighted Minimum Square Linear Fit

In this description we use the following notation: $\boldsymbol{M}(i)$ is the i-th column vector of the matrix $\boldsymbol{M}$, $\boldsymbol{M}[i]$ is the i-th row vector of the matrix $\boldsymbol{M}$, $\boldsymbol{M}^T$ is the transpose of the matrix (vector) $\boldsymbol{M}$.

The problem is to find a line in D-dimensional space which will minimize the error $\boldsymbol{E}$,

$$\underset{\mathrm{D\times N}}{\boldsymbol{E}} = \big(\underset{\mathrm{D\times N}}{\boldsymbol{X}} - \underset{\mathrm{D\times 2}}{\boldsymbol{Y}} \; \underset{\mathrm{2\times N}}{\boldsymbol{T}} \big) \underset{\mathrm{N\times N}}{\boldsymbol{W}} , \tag{B.1}$$

where $\boldsymbol{X}$ is a matrix of N given vectors of dimension D, $\boldsymbol{T}$ is a matrix of reference positions of the given vectors, $\boldsymbol{Y}$ is a matrix of two vectors describing the D-dimensional line, and $\boldsymbol{W}$ is a diagonal matrix of weights specifying the relative importance of the N given vectors.

The vector $\boldsymbol{T}[0]$ specifies the positions of the vectors of the matrix $\boldsymbol{X}$ normalized with respect to the positions of the vectors of the matrix $\boldsymbol{Y}$. The vector $\boldsymbol{Y}(0)$ corresponds to position 0.0 and the vector $\boldsymbol{Y}(1)$ corresponds to position 1.0. The vector $\boldsymbol{T}[1]$ is given by

$$\boldsymbol{T}[1] = \mathbf{1} - \boldsymbol{T}[0] \, . \tag{B.2}$$

where $\mathbf{1}$ is a vector of ones.

Equation (B.1) is rearranged as

$$\boldsymbol{YTW} = \boldsymbol{XW} + \boldsymbol{E} \, . \tag{B.3}$$

We write

$$\underset{\mathrm{2\times N}}{\boldsymbol{T}_{\mathrm{w}}} = \boldsymbol{TW} \tag{B.4}$$

and we multiply (B.3) by $\boldsymbol{T}_{\mathrm{w}}^T$

$$\boldsymbol{Y}\boldsymbol{T}_{\mathrm{w}}\boldsymbol{T}_{\mathrm{w}}^T = \boldsymbol{X}\boldsymbol{W}\boldsymbol{T}_{\mathrm{w}}^T + \boldsymbol{E}\boldsymbol{T}_{\mathrm{w}}^T \, . \tag{B.5}$$

From the orthogonality principle we have

$$\underset{\mathrm{D}\times\mathrm{N}}{\boldsymbol{E}}\,\underset{\mathrm{N}\times\mathrm{D}}{\boldsymbol{T}_{\mathrm{w}}^{T}} = 0\,. \tag{B.6}$$

With

$$\underset{\mathrm{N}\times 2}{\boldsymbol{A}} = \frac{\boldsymbol{W}\boldsymbol{T}_{\mathrm{w}}^{T}}{\boldsymbol{T}_{\mathrm{w}}\boldsymbol{T}_{\mathrm{w}}^{T}}\,, \tag{B.7}$$

the unknown $\boldsymbol{Y}$ is given by

$$\underset{\mathrm{D}\times 2}{\boldsymbol{Y}} = \underset{\mathrm{D}\times\mathrm{N}}{\boldsymbol{X}}\;\underset{\mathrm{N}\times 2}{\boldsymbol{A}}\,. \tag{B.8}$$

If $\boldsymbol{Y}(0)$ is fixed then we write (B.1) as

$$\underset{\mathrm{D}\times\mathrm{N}}{\boldsymbol{E}} = \big(\underset{\mathrm{D}\times\mathrm{N}}{\boldsymbol{X}} - \underset{\mathrm{D}\times 1}{\boldsymbol{Y}(0)}\,\underset{1\times\mathrm{N}}{\boldsymbol{T}[0]} - \underset{\mathrm{D}\times 1}{\boldsymbol{Y}(1)}\,\underset{1\times\mathrm{N}}{\boldsymbol{T}[1]}\big)\;\underset{\mathrm{N}\times\mathrm{N}}{\boldsymbol{W}}\,. \tag{B.9}$$

Now

$$\boldsymbol{Y}(1)\boldsymbol{T}[1]\boldsymbol{W} = \big(\boldsymbol{X} - \boldsymbol{Y}(0)\boldsymbol{T}[0]\big)\boldsymbol{W} + \boldsymbol{E}\,. \tag{B.10}$$

We write

$$\underset{1\times\mathrm{N}}{\boldsymbol{t}_{\mathrm{w}}} = \boldsymbol{T}[1]\boldsymbol{W} \tag{B.11}$$

and multiply (B.10) by $\boldsymbol{t}_{\mathrm{w}}^{T}$

$$\boldsymbol{Y}(1)\boldsymbol{t}_{\mathrm{w}}\boldsymbol{t}_{\mathrm{w}}^{T} = \big(\boldsymbol{X} - \boldsymbol{Y}(0)\boldsymbol{T}[0]\big)\boldsymbol{W}\boldsymbol{t}_{\mathrm{w}}^{T} + \boldsymbol{E}\boldsymbol{t}_{\mathrm{w}}^{T}\,. \tag{B.12}$$

From the orthogonality principle we have

$$\boldsymbol{E}\boldsymbol{t}_{\mathrm{w}}^{T} = 0\,. \tag{B.13}$$

With

$$\underset{\mathrm{N}\times 1}{\boldsymbol{a}} = \frac{\boldsymbol{W}\boldsymbol{t}_{\mathrm{w}}^{T}}{\boldsymbol{t}_{\mathrm{w}}\boldsymbol{t}_{\mathrm{w}}^{T}} \tag{B.14}$$

the unknown $\boldsymbol{Y}(1)$ is given by

$$\underset{\mathrm{D}\times 1}{\boldsymbol{Y}(1)} = \big(\underset{\mathrm{D}\times\mathrm{N}}{\boldsymbol{X}} - \underset{\mathrm{D}\times 1}{\boldsymbol{Y}(0)}\,\underset{1\times\mathrm{N}}{\boldsymbol{T}[0]}\big)\;\underset{\mathrm{N}\times 1}{\boldsymbol{a}} \tag{B.15}$$

The matrix $\boldsymbol{A}$ and the vector $\boldsymbol{a}$ can be pre-calculated for a given number of vectors N and fixed weights $\boldsymbol{W}$.

Bibliography

L. B. Almeida and J. M. Tribolet. Harmonic coding: a low bit-rate good quality speech coding technique. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 1664–1667, Paris, 1982.

B. S. Atal and B. E. Caspers. Beyond multipulse and CELP towards high quality speech at 4 kb/s. In B. S. Atal, V. Cuperman, and A. Gersho, editors, *Advances in Speech Coding*, pp. 191–201. Kluwer Academic Publishers, 1991.

B. S. Atal and J. R. Remde. A new model of LPC excitation for producing natural-sounding speech at low bit rates. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 614–617, Paris, 1982.

B. S. Atal and M. R. Schroeder. Stochastic coding of speech signals at very low bit rates. In *Proc. IEEE Int. Conf. Commun.*, pp. 1610–1613, Amsterdam, 1984.

M. S. Brandstein, J. C. Hardwick, and J. Lim. The multi-band excitation speech coder. In B. S. Atal, V. Cuperman, and A. Gersho, editors, *Advances in Speech Coding*, pp. 215–223. Kluwer Academic Publishers, 1991.

M. S. Brandstein, P. A. Monta, J. C. Hardwick, and J. S. Lim. A real time implementation of the improved MBE speech coder. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 5–8, Albuquerque, 1990.

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

J. P. Campbell, Jr., V. C. Welch, and T. E. Tremain. An expandable error-protected 4800 bps CELP coder (U.S. Federal Standard 4800 bps voice coder). In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 735–738, Glasgow, 1989.

Y. M. Cheng and D. O'Shaughnessy. Automatic and reliable estimation of glottal closures instant and period. *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 37, pp. 1805–1815, Dec. 1989.

D. G. Childers and H. T. Hu. Speech synthesis by glottal excited linear prediction. *J. Acoustical Society of America*, vol. 96, pp. 2026–2038, Oct. 1994.

S. Cucchi, M. Fratti, and M. Ronchi. On improving performance of analysis by synthesis speech coders. *IEEE Trans. Speech and Audio Processing*, vol. 4, pp. 243–247, May 1996.

G. Davidson and A. Gersho. Complexity reduction methods for vector excitation coding. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 3055–3058, Tokyo, 1986.

Aloknath De. *Auditory Distortion Measures for Speech Coder Evaluation.* Ph.D. Thesis, McGill University, Montreal, Canada, 1993.

A. DeJaco, W. Gardner, P. Jacobs, and C. Lee. QCELP: The North American CDMA digital cellular variable rate speech coding standard. In *Proc. IEEE Workshop on Speech Coding for Telecom.*, pp. 5–6, Sainte-Adèle, Québec, 1993.

J. R. Deller Jr., J. G. Proakis, and J. H. L. Hansen. *Discrete-Time Processing of Speech Signal.* Macmillan, 1993.

A. El-Jaroudi and J. Makhoul. Discrete all-pole modelling. *IEEE Trans. Signal Processing*, vol. 39, pp. 411–423, Feb. 1991.

M. Elshafei-Ahmed and M. I. Al-Suwaiyel. Fast methods for code search in CELP. *IEEE Trans. Speech and Audio Processing*, vol. 1, pp. 315–325, July 1993.

M. Festa and D. Sereno. A speech coding algorithm based on prototype interpolation with critical bands and phase coding. In *Proc. European Conf. on Speech Commun. and Technology*, pp. 229–232, Madrid, 1995.

B. Fette, C. Bergstrom, S. You, and C. Jaskie. High quality 2400 bps vocoder research. In *Proc. IEEE Workshop on Speech Coding for Telecom.*, pp. 45-46, Sainte-Adèle, Québec, 1993.

H. Fujisaki and M. Ljungqvist. Proposal and evaluation of models for the glottal source waveform. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 1605–1608, Tokyo, 1986.

A. Gersho. Advances in speech and audio compression. *Proc. IEEE*, vol. 82, pp. 900–918, June 1994.

A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

I. A. Gerson and M. A. Jasiuk. Techniques for improving the performance of CELP type speech coders. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 205–208, Toronto, 1991a.

I. A. Gerson and M. A. Jasiuk. Vector sum excited linear prediction (VSELP). In B. S. Atal, V. Cuperman, and A. Gersho, editors, *Advances in Speech Coding*, pp. 69–79. Kluwer Academic Publishers, 1991b.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, second edition, 1989.

R. M. Gray. Vector quantization. *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-34, April 1984.

D. W. Griffin and J. S. Lim. Signal estimation from modified short-time Fourier transform. *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-32, pp. 236–243, April 1984.

J. C. Hardwick and J. S. Lim. A 4.8 kbps multi-band excitation speech coder. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 374–377, New York, 1988.

J. C. Hardwick and J. S. Lim. A 4800 bps improved multi-band excitation speech coder. In *Proc. IEEE Workshop on Speech Coding for Telecom.*, Vancouver, 1989.

P. Hedelin. High quality glottal LPC-vocoding. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 465–468, Tokyo, 1986.

N. Jayant, J. Johnston, and R. Safranek. Signal compression based on models of human perception. *Proc. IEEE*, vol. 81, pp. 1385–1422, Oct. 1993.

Y. Jiang and V. Cuperman. Encoding prototype waveforms using a phase codebook. In *Proc. IEEE Workshop on Speech Coding for Telecom.*, pp. 21–22, Annapolis, 1995.

P. Kabal and R. P. Ramachandran. The computation of line spectral frequencies using Chebyshev polynomials. *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-34, pp. 1419–1426, Dec. 1986.

W. B. Kleijn. Continuous representations in linear predictive coding. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 201–204, Toronto, 1991.

W. B. Kleijn. Encoding speech using prototype waveforms. *IEEE Trans. Speech and Audio Processing*, vol. 1, pp. 386–399, Oct. 1993.

W. B. Kleijn and W. Granzow. Methods for waveform interpolation in speech coding. *Digital Signal Processing*, vol. 1, pp. 215–230, Jan. 1991.

W. B. Kleijn and J. Haagen. A general Waveform-Interpolation structure. In *Proc. European Signal Processing Conf.*, pp. 1665–1668, Edinburg, 1994a.

W. B. Kleijn and J. Haagen. Transformation and decomposition of the speech signal for coding. *IEEE Signal Processing Letters*, vol. 1, pp. 136–138, Sept. 1994b.

W. B. Kleijn and J. Haagen. Speech coder based on decomposition of characteristic waveforms. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 508–511, Detroit, 1995a.

W. B. Kleijn and J. Haagen. Waveform interpolation for coding and synthesis. In W. B. Kleijn and K. K. Paliwal, editors, *Speech Coding and Synthesis*, pp. 175–208. Elsevier, 1995b.

W. B. Kleijn and D. J. Krasinski. Fast methods for the CELP speech coding algorithm. *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 38, pp. 1330–1342, Aug. 1990.

W. B. Kleijn, D. J. Krasinski, and R. H. Ketchum. An efficient stochastically excited linear predictive coding algorithm for high quality low bit rate transmission of speech. *Speech Commun.*, vol. 7, pp. 305–316, Oct. 1988a.

W. B. Kleijn, D. J. Krasinski, and R. H. Ketchum. Improved speech quality and efficient vector quantization in SELP. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 155–158, New York, 1988b.

W. B. Kleijn, P. Kroon, L. Cellario, and D. Sereno. A 5.85 kbps CELP algorithm for cellular applications. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 596–599, Minneapolis, 1993.

W. B. Kleijn, P. Kroon, and D. Nahumi. The RCELP speech coding algorithm. *European Trans. on Telecom.*, vol. 5, pp. 573–582, Sept./Oct. 1994.

W. B. Kleijn and K. K. Paliwal. An introduction to speech coding. In W. B. Kleijn and K. K. Paliwal, editors, *Speech Coding and Synthesis*, pp. 1–47. Elsevier, 1995.

W. B. Kleijn, R. P. Ramachandran, and P. Kroon. Generalized analysis-by-synthesis coding and its application to pitch prediction. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 337–340, San Francisco, 1992.

W. B. Kleijn, R. P. Ramachandran, and P. Kroon. Interpolation of the pitch predictor parameters in analysis-by-synthesis speech coders. *IEEE Trans. Speech and Audio Processing*, vol. 2, pp. 42–54, Jan. 1994.

W. B. Kleijn, Y. Shoham, D. Sen, and R. Hagen. A low-comlexity Waveform Interpolation coder. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 212–215, Atlanta, 1996.

A. K. Krishnamurthy. Glottal source estimation using a sum-of-exponentials model. *IEEE Trans. Signal Processing*, vol. 40, pp. 682–686, March 1992.

P. Kroon. Evaluation of speech coders. In W. B. Kleijn and K. K. Paliwal, editors, *Speech Coding and Synthesis*, pp. 467–494. Elsevier, 1995.

P. Kroon and B. S. Atal. Pitch predictors with high temporal resolution. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 661–664, Albuquerque, 1990.

P. Kroon, E. F. Deprettere, and R. J. Sluyter. Regular-Pulse Excitation: A novel approach to effective and efficient multi-pulse coding of speech. *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-34, pp. 1054–1063, Oct. 1986.

P. Kroon and F. Deprettere. A class of analysis-by-synthesis predictive coders for high quality speech coding at rates between 4.8 and 16 kbits/s. *IEEE J. Selected Areas Commun.*, vol. 6, pp. 353–363, Feb. 1988.

P. Kroon and W. B. Kleijn. Linear-prediction based analysis-by-synthesis coding. In W. B. Kleijn and K. K. Paliwal, editors, *Speech Coding and Synthesis*, pp. 79–115. Elsevier, 1995.

G. Kubin, B. S. Atal, and W. B. Kleijn. Performance of noise excitation for unvoiced speech. In *Proc. IEEE Workshop on Speech Coding for Telecom.*, pp. 35–36, Sainte-Adèle, Québec, 1993.

C. Ma, Y. Kamp, and L. F. Willems. A Frobenius norm approach to glottal closure detection from the speech signal. *IEEE Trans. Speech and Audio Processing*, vol. 2, pp. 258–265, April 1994.

J. Makhoul. Linear prediction: A tutorial review. *Proc. IEEE*, vol. 63, pp. 561–580, April 1975.

J. Makhoul, S. Roucos, and H. Gish. Vector quantization in speech coding. *Proc. IEEE*, vol. 73, pp. 1551–1588, Nov. 1985.

K. Mano, T. Moriya, S. Miki, H. Ohmuro, K. Ikeda, and J. Ikedo. Design of a pitch synchronous innovation CELP coder for mobile communications. *IEEE J. Selected Areas Commun.*, vol. 13, pp. 31–40, Jan. 1995.

J. D. Markel and A. H. Gray. *Linear Prediction of Speech*. Springer-Verlag, Berlin, Germany, 1976.

J. S. Marques, I. M. Trancoso, J. M. Tribolet, and L. B. Almeida. Improved pitch prediction with fractional delays in CELP coding. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 665–668, Albuquerque, 1990.

R. McAulay, T. Parks, T. Quatieri, and M. Sabin. Sine-wave amplitude coding at low data rates. In B. S. Atal, V. Cuperman, and A. Gersho, editors, *Advances in Speech Coding*, pp. 203–213. Kluwer Academic Publishers, 1991.

R. J. McAulay and T. F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-34, pp. 744–754, Aug. 1986.

R. J. McAulay and T. F. Quatieri. Sinusoidal coding. In W. B. Kleijn and K. K. Paliwal, editors, *Speech Coding and Synthesis*, pp. 121–174. Elsevier, 1995.

A. McCree, K. Truong, E. B. George, T. P. Barnwell III, and V. Viswanathan. A 2.4 kbit/s MELP coder candidate for the new U.S. Federal Standard. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 200–203, Atlanta, 1996.

A. V. McCree and T. P. Barnwell III. Implementation and evaluation of a 2400 bps mixed excitation LPC vocoder. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 159–162, Minneapolis, 1993.

A. V. McCree and T. P. Barnwell III. Mixed excitation LPC vocoder model for low bit rate speech coding. *IEEE Trans. Speech and Audio Processing*, vol. 3, pp. 242–250, July 1995.

N. Moreau and P. Dymarski. Selection of excitation vectors for the CELP coders. *IEEE Trans. Speech and Audio Processing*, vol. 2, pp. 29–41, Jan. 1994.

D. Nahumi and W. B. Kleijn. An improved 8 kb/s RCELP coder. In *Proc. IEEE Workshop on Speech Coding for Telecom.*, pp. 39–40, Annapolis, 1995.

D. O'Shaughnessy. *Speech Communication: Human and Machine*. Addison-Wesley Publishing Company, 1987.

B. Paillard, J. Soumagne, P. Mabilleau, and S. Morissette. PERCEVAL: Perceptual evaluation of the quality of audio signals. *J. Audio Eng. Society*, vol. 40, pp. 21–31, January/February 1992.

K. K. Paliwal. Interpolation properties of linear prediction parametric representations. In *Proc. European Conf. on Speech Commun. and Technology*, Madrid, 1995.

K. K. Paliwal and B. S. Atal. Efficient vector quantization of LPC parameters at 24 bits/frame. *IEEE Trans. Speech and Audio Processing*, vol. 1, pp. 3–14, Jan. 1993.

K. K. Paliwal and W. B. Kleijn. Quantization of LPC parameters. In W. B. Kleijn and K. K. Paliwal, editors, *Speech Coding and Synthesis*, pp. 433–466. Elsevier, 1995.

M. R. Portnoff. Short-time Fourier analysis of sampled speech. *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-29, pp. 364–373, June 1981a.

M. R. Portnoff. Time-scale modification of speech based on short-time Fourier analysis. *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-29, pp. 374–390, June 1981b.

L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall, 1978.

R. P. Ramachandran and P. Kabal. Stability and performance analysis of pitch filters in speech coders. *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-35, pp. 937–946, July 1987.

R. P. Ramachandran and P. Kabal. Pitch prediction filters in speech coding. *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 37, pp. 467–477, April 1989.

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 proposed ITU-T 8-kb/s speech coding standard. In *Proc. IEEE Workshop on Speech Coding for Telecom.*, pp. 3–4, Annapolis, 1995.

R. Salami, C. Laflamme, J.-P. Adoul, and D. Massaloux. A toll quality 8 kb/s speech codec for the personal communications system (PCS). *IEEE Trans. Vehicular Technology*, vol. 43, pp. 808–816, Aug. 1994.

M. R. Schroeder and B. S. Atal. Code-excited linear predictive (CELP): High quality speech at very low bit rates. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 937–940, Tampa, 1985.

M. R. Schroeder, B. S. Atal, and J. L. Hall. Optimizing digital speech coders by exploiting masking properties of the human ear. *J. Acoustical Society of America*, vol. 66, pp. 1647–1652, Dec. 1979.

D. Sen and W. B. Kleijn. Synthesis methods in sinusoidal and Waveform-Interpolation coders. In *Proc. IEEE Workshop on Speech Coding for Telecom.*, Annapolis, 1995.

Y. Shoham. Constrained-stochastic excitation coding of speech at 4.8 kb/s. In B. S. Atal, V. Cuperman, and A. Gersho, editors, *Advances in Speech Coding*, pp. 339–348. Kluwer Academic Publishers, 1991.

Y. Shoham. Low-rate speech coding based on time-frequency interpolation. In *Proc. Int. Conf. on Spoken Language Processing*, pp. 37–40, 1992.

Y. Shoham. High-quality speech coding at 2.4 based on time-frequency interpolation. In *Proc. European Conf. on Speech Commun. and Technology*, pp. 741–744, Berlin, 1993a.

Y. Shoham. High-quality speech coding at 2.4 to 4.0 kbps based on time-frequency interpolation. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 167–170, Minneapolis, 1993b.

R. Smits and B. Yegnanarayana. Determination of instants of significant excitation in speech using group delay function. *IEEE Trans. Speech and Audio Processing*, vol. 3, pp. 325–333, Sept. 1995.

F. K. Soong and B. Juang. Line spectrum pair (LSP) and speech data compression. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 1.10.1–1.10.4, San Diego, 1984.

F. K. Soong and B. Juang. Optimal quantization of LSP parameters. *IEEE Trans. Speech and Audio Processing*, vol. 1, pp. 15–24, Jan. 1993.

J. Stachurski and P. Kabal. A pitch pulse evolution model for a dual excitation linear predictive speech coder. In *Proc. Seventeenth Biennial Symposium on Communications*, pp. 107–110, Kingston, 1994.

Y. Tanaka and H. Kimura. Low-bit-rate speech coding using a two-dimensional transform of residual signals and Waveform Interpolation. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 173–176, Adelaide, 1994.

K. W. Tang and B. M. G. Cheetham. Fixed bit-rate PWI speech coding with variable frame length. In *Proc. IEEE Globecom Conf.*, pp. 1600–1603, 1995.

T. Taniguchi, M. Johnston, and Y. Ohta. Pitch-sharpening for perceptually improved CELP and the sparse-delta codebook for reduced computation. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 241–244, Toronto, 1991.

I. M. Trancoso and B. S. Atal. Efficient procedures for selecting the optimum innovation in stochastic coders. *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 38, pp. 385–396, March 1990.

I. M. Trancoso, J.S. Marques, and C. M. Ribeiro. CELP and sinusoidal coders: Two solutions for speech coding at 4.8–9.6 kbps. *Speech Commun.*, vol. 9, pp. 389–400, Dec. 1990.

F. F. Tzeng. Analysis-by-synthesis linear predictive speech coding at 4.8 kbits/s and below. In B. S. Atal, V. Cuperman, and A. Gersho, editors, *Advances in Speech Coding*, pp. 135–143. Kluwer Academic Publishers, 1991.

S. Wang and A. Gersho. Improved excitation for phonetically-segmented VXC speech coding below 4 kb/s. In *Proc. IEEE Globecom Conf.*, pp. 946–950, San Diego, 1990.

S. Wang, A. Sekey, and A. Gersho. An objective measure for predicting subjective quality of speech coders. *IEEE J. Selected Areas Commun.*, vol. 10, pp. 819–829, June 1992.

G. Yang, H. Leich, and R. Boite. Voiced speech coding at very low bit rates based on forward-backward waveform prediction. *IEEE Trans. Speech and Audio Processing*, vol. 3, pp. 40–47, Jan. 1995.

M. R. Zad-Issa and P. Kabal. Smoothing the evolution of the spectral parameters in linear prediction of speech using target matching. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 1699–1702, Munich, 1997.