A Study on Efficient Algorithms for Temporal Decomposition of Speech

by

Phu Chien NGUYEN

submitted to
Japan Advanced Institute of Science and Technology
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Supervisor: Professor Masato AKAGI

School of Information Science
Japan Advanced Institute of Science and Technology

July 15, 2003

Abstract

The temporal decomposition (TD) model of speech is a description of speech in terms of event targets that possibly describe the ideal articulatory configurations of the successive acoustic events that occur in speech, and event functions that describe their temporal evolutionary patterns. Following Atal's initial paper, a number of modifications and applications have been explored in the literature. Modified algorithms for TD have been mainly proposed to overcome the two major drawbacks of high computational cost and the high parameter sensitivity of the number and locations of the events. On the application side, the concept of TD has attracted many researchers in recent years, especially in application areas such as speech coding, speech recognition, speech segmentation, and speech synthesis.

There were many spectral parameter sets being considered for TD such as log-area, log area ratios, cepstrum, etc. However, due to the stability problems in the LPC model, not all types of parametric representations of speech can be used. This is because there is no guarantee that the selected spectral parameters are valid after the spectral transformation performed by TD. Accordingly, only the LPC parameters which can directly be tested for the system stability are used as input for TD. This results in the fact that LSF parameters have rarely been considered as a candidate spectral parameter for TD. Also, the TD performance depends significantly on the type of parameters used, particularly due to the Euclidean distance measurement performed in the space of the parameters selected.

It is worth making LSF parameters possible for TD since the LSFs are the favored format for the LPC parameter representation. The LSFs are useful because of sensitivity (an adverse alteration of one coefficient results in a spectral change only around that frequency) and efficiency (LSFs result in low spectral distortion while being interpolated and/or quantized). This does not occur with other representations. As long as the LSF coefficients are ordered in the interval $(0; \pi)$, the resulting LPC filter is stable. Another desirable property of LSFs is that they are related to formants. Closer LSFs produce a sharper formant peak. This property provides a useful, practical check for the stability after the LSFs has been interpolated and/or quantized. The LSFs can be checked for a minimum spacing, and separated slightly if necessary.

For the above reasons, TD of LSF parameters has the following advantages: (i) it can achieve high reconstruction accuracy; (ii) it has desirable properties to be applied in voice modification; and (iii) it can be beneficially integrated into most of current speech coding systems. Therefore, it is crucial to make LSF parameters possible for TD. Also, for the use in real-time applications, it is desirable to have a method of TD with short algorithmic delay and low computational cost. However, most algorithms for TD method require more than 200 ms buffering delay which is not suitable for such kinds of applications. Moreover, they are very computationally costly, which has been mainly attributed to the use of the singular value decomposition (SVD) routine and the Gauss-Seidel iterations. On the application side, the primitive and also major application of TD is in speech coding, in which producing high-quality speech at rates below 2.4 kbps is still a challenging issue.

In this thesis, we have developed a method of TD for LSF parameters called Modified RTD (MRTD) which was derived from the Restricted Temporal Decomposition (RTD) algorithm. The RTD method intends to make LSF parameters possible for TD by considering the LSF ordering property of event targets. However, RTD still has not completely preserved the LSF ordering property of the event targets. In addition, event functions obtained from RTD analysis may be ill-shaped, i.e., some of them may have more than one peak, which is undesirable from speech coding point of view. The MRTD algorithm imposes a new constraint on the event functions so that the drawbacks of RTD in terms of ill-shaped event functions have been overcome. In addition, it uses an improved procedure for preserving the LSF ordering property of event targets so that the stability of the corresponding LPC synthesis filter after spectral transformation performed by MRTD has been completely ensured. The MRTD method employs the restricted second order TD model, in which only two event functions at any moment of time can overlap and all event functions sum up to one. Besides, this method uses a spectral stability criterion in event localizing. It is shown in the thesis that both speech spectral and excitation information of speech can be well described and quantized using the MRTD technique.

We have also developed the second algorithm for TD, called Limited Error Based Event Localizing Temporal Decomposition (LEBEL-TD). LEBEL-TD uses a limited error criterion for initially estimating the event locations, and then refines them using a local optimization strategy. It requires only 75 ms algorithmic delay, which has been known to be the shortest algorithmic delay required for TD analysis so far. In addition, LEBEL-TD has significantly reduced the computational cost of TD since it requires neither the SVD routine nor the Gauss-Seidel iterations.

Being motivated by the fact that STRAIGHT (stands for "Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum") is a very high-quality vocoder, we have therefore developed two STRAIGHT-based very low-bit-rate speech coding methods using the proposed TD algorithms. The two speech coders operate at the bit rates of around 1.2 and 1.8 kbps using MRTD and LEBEL-TD, respectively. The former can give the speech quality close, while the later can give the speech quality comparable, to that of the 4.8 kbps FS-1016 CELP coder. The application of MRTD is not limited to speech coding, it is shown in the thesis that the event targets derived from LSF parameters using MRTD were found effective when used in vector quantization based speaker identification systems as a feature set.

In summary, we have proposed two efficient algorithms for TD of LSF parameters and investigated their applications in speech coding and speaker recognition. But it is more than that, the geometric interpretation of MRTD and LEBEL-TD as an effective breakpoint analysis procedures gives a means of speech segmentation and speech recognition. The fact that event targets extracted by MRTD can convey speaker identity and the localized nature of each LSF provide necessary motivation to investigate the application of MRTD in voice conversion. More interestingly, using MRTD we can control spectral envelopes, durations, and fundamental frequencies independently and flexibly, which suggests its potential applications in emotional speech, song synthesis, text-to-speech synthesis, etc. To prepare for future research towards this end, we have developed a voice transformation system based on the modification of formants in the LSF domain.

Key words: temporal decomposition, event targets, event functions, LSF, speech coding, speaker recognition, voice transformation, STRAIGHT

Acknowledgments

It was my good fortune to have Professor Masato Akagi as my supervisor while at Japan Advanced Institute of Science and Technology (JAIST). He taught me how to be a fruitful researcher, write good papers and, above all, have a good attitude. His thorough scientific approach and unending quest for excellence have been inspirational in the years of my thesis research. I only wish I had listened to his advice more often.

I would like to express my sincere thanks, appreciation, and gratitude to Professor Ho Tu Bao of JAIST for his guidance, support, and encouragement, which made possible for me to successfully complete this research program. Without his understanding and inspiration, this thesis would never have existed.

I would like to thank Professor Hideki Kawahara of Wakayama University, Professor Teruo Matsuzawa and Associate Professor Kazunori Kotani of JAIST, for many detailed and valuable comments on the thesis. Special thanks go to Professor Hideki Kawahara for kindly making available his STRAIGHT system which was used in my thesis research.

I would also like to acknowledge Associate Professor Jianwu Dang, now on leave at Institut de la Communication Parlée (France), Associate Professor Hiroshi Shimodaira, and Research Associate Masashi Unoki of JAIST, for their numerous suggestions, advice, and comments, which helped improving the final work.

I am grateful to my former supervisors, Dr. Luong Chi Mai of Vietnam Institute of Information Technology, Dr. Ho Cam Ha and Dr. Do Van Thanh of Hanoi University of Pedagogy, for their advice and encouragement throughout my studies.

I owe a debt of gratitude to Sung-Joo Kim of Samsung Corporation (Korea), formerly with Korea Advanced Institute of Science and Technology, and Athaudage C.R. Nandasena of Melbourne University (Australia), for kindly providing me with their doctoral theses on temporal decomposition of speech and patiently answering my countless questions. I also wish to thank Tomi Kinnunen of Joensuu University (Finland), for many fruitful discussions on speaker recognition. Thanks are also due to Robert Morris of Georgia Institute of Technology (USA), for his help in implementing the formant modification algorithm. Tomoki Toda of Nara Institute of Science and Technology (NAIST) and ATR Human Information Science Laboratories, now with Carnegie Mellon University (USA), never hesitated in offering me his experience in voice conversion.

I sincerely thank all my friends and colleagues who always supported me in times of need. I greatly appreciate to my lab-mates for their contributions in making a wonderful and supportive academic environment. Specially, Kazuhito Ito (now graduated from his doctoral course), Yuichi Ishimoto − my "tutor," Fumiyasu Ikarashi, and Hironori Nishimoto gave me many helps through the years. The life would be nothing without friends.

It is impossible to mention all of them here, but some had nevertheless a direct influence on this thesis work. Thank you, Thinh, Huong, Tam, and the others, for sharing joys and sorrows with me. To my many Vietnamese friends at JAIST, thanks for the good times over the past three years.

But the life would be also difficult without financial support. I am deeply indebted to the Japanese Ministry of Education, Culture, Sports, Science and Technology for granting me a scholarship, which made possible for my study in Japan. Thanks also go to CREST (Core Research for Evolutional Science and Technology) of JST (Japan Science and Technology Corporation), the Foundation for C&C Promotion, the Telecommunications Advancement Foundation, the JAIST Foundation, and the IBM Corporation for providing me with their travel grants which supported me to attend and present my work at some international conferences.

JAIST offered me the greatest learning environment I have ever had − the computing environment, the brilliant faculty, the hard-working students, and the chance to meet famous researchers all over the world. Among the friendly administrators, I owe a great deal to the International Student Section for the kind and constant assistance they provided. Without them, I would certainly have run into much trouble.

Finally, I have saved the best for the last. I wish to express my endless love and gratitude to my family, Mom, Dad, Binh, Minh and Hieu, for always being there when I needed them and supporting me through all my years of school. I am especially grateful to my parents for everything they taught me and for all the sacrifices they made in my upbringing. Most of all, I thank my wonderful fiancée. She patiently looked after me, even though we were thousands of miles away. She never had a word of complaint when I was negligent. She comforted me when I was discouraged. I may never be able to repay her, but fortunately, I have a lifetime to try.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Temporal Decomposition

In articulatory phonetics, speech production is considered as a sequence of overlapping articulatory gestures, each of which may be thought of as a movement towards and away from an ideal, but often not reached, articulatory target. The sound produced by such an articulatory movement corresponds to a phoneme or a sub-phoneme in speech. In other words, each gesture produces an acoustic event that should approximate a phonetic target. Adjacent gestures overlap one another resulting in the characteristic transitions between phonemes that can be observed in almost any parametric representation of the acoustic speech signal. Due to co-articulation and reduction in fluent speech, a target may not be reached before articulation towards the next phonetic target begins. It has long been a difficult task to determine such targets and their temporal evolutionary patterns from the acoustic signal alone.

The so-called *temporal decomposition* (TD) method [8] for analyzing speech achieves the objective of decomposing speech into targets and their temporal evolutionary patterns, without any recourse to any explicit phonetic knowledge. This model of speech takes into account the above articulatory considerations and results in a description of speech in terms of event functions and their corresponding event targets. The event targets are supposed to model ideal articulatory targets of which the event functions describe the temporal evolution. Therefore, it tries to achieve an optimal transformation from the multidimensional spectral parameter space to the phonetic space which can be considered for many applications to be a powerful speech analysis technique.

Each acoustic event in speech starts, gradually grows in magnitude and vanishes with a certain degree of overlapping between them. For this reason, the event functions which are representative of the temporal evolutionary patterns of these events should be: (i) *time-limited* to describe explicitly, the start and end points in time and the duration of each event, (ii) *non-negative* to describe the magnitude of the events during their existence, and (iii) *smooth* to describe the gradualness of growth and decay of the events resembling the gradualness of movement of the articulators in speech production. In the temporal decomposition analysis point of view, these properties of the event functions can be used as mathematical constraints in determining the event functions.

Fig. 1.1 shows an example of event functions and event targets extracted by using the temporal decomposition technique. Here, the plot of event functions is shown together with their corresponding event targets in the log power spectrum form. The log power

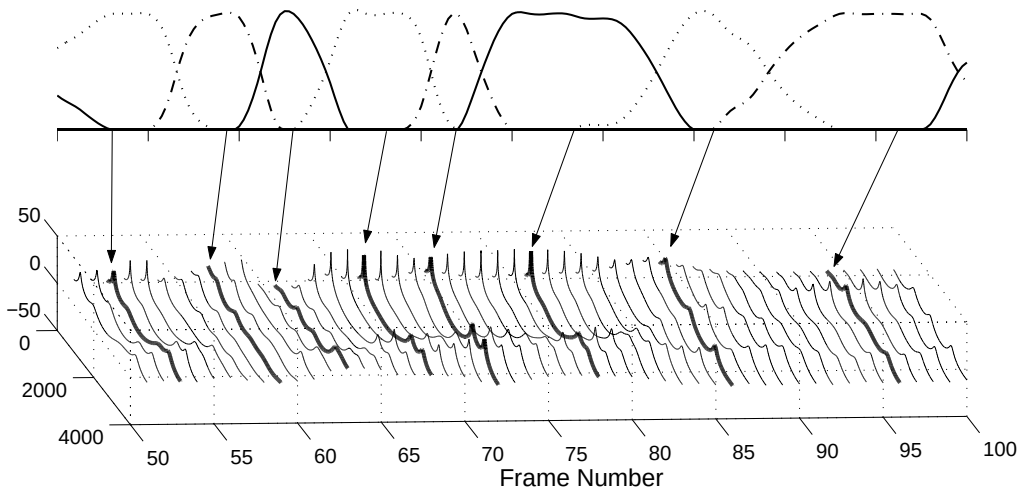


Figure 1.1: Example of event functions (top) and corresponding event targets (bottom). The top figure shows the plot of event functions, while in the bottom figure, dark solid lines show the log power spectra of event targets. The log power spectra of original spectral parameters are also given.

spectra of the original spectral parameters are also provided for reference.

1.2 Applications of TD

Originally, the temporal decomposition (TD) method was proposed by Atal [8] as a technique for economical speech coding. However, subsequent researches on TD have found its applications in many other areas of speech processing such as speech recognition, segmentation, and synthesis. The fact that TD decomposes the speech parameters into two elementary components, which occur at a lower rate than the original speech parameters, gives a means of coding speech efficiently at a lower bit rate, e.g., [21, 127, 48, 85]. The strong relationship between the temporal decomposition representation of speech and the speech production mechanism has provided the necessary motivation to investigate its application in speech recognition [17, 73, 106, 141]. Its usefulness in speech segmentation [34] and speech synthesis [3, 27, 16] has also been investigated.

In this section, some applications of temporal decomposition (including potential ones) which are directly related to this thesis research are briefly described.

1.2.1 Very Low-Bit-Rate Speech Coding

Speech coding is the process of obtaining a compact representation of the speech signal for efficient transmission over band-limited wired and wireless channels and/or storage. Today, speech coders have become essential components in telecommunications and in the multimedia infrastructure. Commercially systems that rely on efficient speech coding include cellular communication, voice over internet protocol (VoIP), video conference, electronic toys, archiving, and digital simultaneous voice and data, as well as numerous PC-based games and multimedia applications.

Speech coders differ primarily in bit rate, which is often measured in bits/second (bps) or kilobits/second (kbps); complexity, which is measured in multiplications/second; delay, which is measured in milliseconds (ms) between recording and playback; and perceptual quality of the synthesized speech. Speech coding can also be roughly classified into two classes: wideband coding and narrowband coding. Wideband coding refers to coding of 7 kHz-bandwidth speech signals (sampling rate is increased to 16 kHz) while narrowband speech coding refers to coding of speech signals whose bandwidth is less than 4 kHz (8 kHz sampling rate). Narrowband speech coding is more common than wideband speech coding mainly since speech coders often interface with the conventional telephone network with its narrowband nature (300-3400 Hz) [76].

Speech coding is the art of creating minimally redundant representation of the speech signal that can be efficiently transmitted or stored in digital media, and decoding the signal with the best possible perceptual quality. Redundancies, introduced in the speech signal during the speech production process, make it possible to encode speech at low-bit rates. Therefore, speech coding can be regarded as a selection process, to select what information is redundant, and how to encode the important information accurately. Moreover, our hearing system is not equally sensitive to distortions at different frequencies and has a limited dynamic range. Speech coding techniques take advantage of these properties for reducing the bit rate.

Low-bit-rate speech coding is commonly classified whose bit rate range below 8 kbps and down to 2.4 kbps, and the term *very low-bit-rate* is used for coders operating below 2.4 kbps [131]. Very low-bit-rate speech coding is developed mainly for secured communications. Other applications include restricted rate voice communications, voice answering machine, voice mail, human-machine interaction, voice recording and storage, high capacity voice archiving. Typically, coder quality diminishes with decreasing bit rate. This occurs because redundant information is removed to reduce the bit rate. However, as the number of bits is reduced beyond a certain point, the ability to remove bits with redundant information is more difficult.

Low-bit-rate speech coders generally quantize the parameters of a parametric model of the speech production system. Such a model is the *linear predictive coding* (LPC) model, where the synthetic speech is generated by exciting a filter with pitch pulses or random noise depending whether the speech is voiced or unvoiced. Therefore, in the LPC model, speech production can be completely synthesized with knowledge of the filter parameters, whether the speech sample is voiced or unvoiced, and the pitch period if the speech is voiced. The parameters all vary with time.

The analysis of excitation is primarily concerned with determining the voiced/unvoiced characteristic of the speech, and the pitch period if the speech is voiced. The estimation of the fundamental frequency or pitch period in the excitation source is an important process in speech analysis. From the pitch estimation, a voiced/unvoiced decision of the speech is made. Incorrectly estimating an unvoiced segment of speech as voiced results in a dull sound, whilst incorrectly estimating of a voiced segment as unvoiced results in objectionable distortion. There exists many techniques to determine the pitch period. Pitch estimation algorithms can operate either on the time domain waveform or on the spectrum of the speech signal. A number of pitch detection algorithms are based on computing the short-term autocorrelation of the speech signal. The concept behind the method, is that the correlation will have peaks at multiples of the pitch period, where the voiced speech is correlated with itself. Pitch detection is a very difficult process and

its emphasis should not be underestimated. Post-processing of the pitch contour is often performed on multiple frames to improve the accuracy of the pitch estimation algorithm.

Unfortunately, naturalness is a personal component of speech which cannot be accurately modelled. The simple excitation used in the LPC model sacrifices the naturalness of the speech. Different methods are available which dedicate additional bits to encoding the unique components in the excitation. Since unique components are unpredictable, i.e. random, existing audio encoding techniques such as *Adaptive Differential Pulse Code Modulation* (ADPCM) can be used to encode the excitation. Improvement in quality can be obtained by using a more complex excitation than simple pulses. A common method is to used several pulses selectively placed to reduce the some error criteria. Another technique is to generate a codebook of common excitations and choose the optimal codeword to minimize some error criteria. This is known as *Code-Excited Linear Prediction coder* (CELP) and can reduce the bit rate with good quality down to 4.8 kbps [10].

The filter parameters also undergo much analysis to reduce the bit rate. The filter is typically an all-pole design. No zeros in the filter are required, since each zero can effectively be represented by a complex pole pair. The all-pole filter also has the advantage that its parameters can be predicted using only previous samples. Thus, this is why the model is called a linear predictive coding. The parameters of the LPC filter can be coefficients of the characteristic equation. However, filter coefficients have poor encoding qualities, mainly due to their wide range of values that they accept. For this reason, it is preferable to transform the set of filter coefficients to a number of other parameter sets. Such alternative representations can provide different quantization and stability properties. The filter parameters represent the spectral properties of the speech signal. They are often referred to as spectral parameters.

Of the parameters of the model, the spectral parameters occupy most of the bits in the final bit rate. Consequently, improved encoding of the spectral parameters will have considerable impact on the system. *Scalar quantization* techniques quantize the spectral parameters separately using uniform or non-uniform quantization. Scalar Quantization is a relatively simple procedure, but its performance in bit rate is limited. The spectral parameters of a frame are correlated with each other, and *vector quantization* (VQ) techniques quantize a set of spectral parameters jointly as a single vector. Given an input vector of spectral parameters, the best match of this vector from a finite set of code vectors stored in a codebook is determined according to a minimum distortion rule. VQ encodes more efficiently the dependence of the components of the input vector, which cannot be captured in scalar quantization. An 800 bps vector quantization vocoder is comparable in performance to a 2.4 kbps LPC vocoder [146]. Further bit reduction in quantization of spectral parameters can be achieved by employing the interframe dependence of the spectral parameters. An *adaptive codebook*, dependent on the previously selected codeword, leads to a reduction in the number of codevectors used in quantizing each input vectors.

Different sounds in speech change at different rates. Therefore, if the rate of parameters is reduced to match the rate of changes in the speech, then we can achieve further bit rate reduction. *Variable-frame rate* (VFR) quantizes the spectral parameters only when the properties of the speech signal have changed sufficiently [45, 144]. The spectral parameters of the non-quantized frame are constructed using linear interpolation between quantized frames. An optimal VFR technique is proposed in [28], resulting in a speech coder operating at 300 bps with good intelligibility.

One approach is to perform vector quantization on multiple frame vectors. This

method is referred to as segment or matrix quantization [137]. Segment quantization is based on the principle that rate-distortion performance can be obtained using a longer block for quantization. In VQ, individual parameter frames, i.e. speech parameter vectors, are quantized and encoded separately. Therefore, this technique only accounts for the correlation within frames, i.e. intra-frame correlation. In contrast, several speech parameter frames, i.e. block of frames, are taken as the data unit for the quantization process in segment quantization, thus exploiting both intra-frame and inter-frame correlations in the speech parameters. This approach has the potential to achieve speech coding below 300 bps. An additional improvement is *variable-length segmentation* [126] with segmentation of the spectral sequence into variable-length segments. The length of the segments is determined such that the speech within a segment is relatively similar. Thus, ideally a segment should contain a single phoneme for optimal coding. Therefore, after all these improvements to the original LPC model, variable-length segmentation actually tries to determine the phonemes of speech. However, determining the boundaries between segments has proved to be a very difficult task.

Both vector and segment quantization are, however, bound to frame-based analysis. An alternative is event-based analysis, where the non-uniform articulation of speech is represented by non-uniform spaced, variable-length, phoneme-like events. Temporal decomposition (TD) method [8] is used to represent the continuous variation of the spectral parameter vectors by a linear combination of a series of data vectors in the same dimensional space, called event targets or event vectors, using an associated series of time-overlapping interpolation functions of different lengths called event functions. Therefore, we can represent speech by a set of events, each consisting of an event target and an associated event function. Since the event rate, i.e. the number of events per second, is much less than the frame rate, and both event targets and functions can be quantized efficiently, TD has been considered as a useful technique for efficient coding of LPC parameters.

As the name suggests, in TD, we attempt to decompose the speech signal into events. An event can be considered as a phoneme or sub-phoneme, however, this is an oversimplification of the process. Thus, temporal decomposition is a powerful tool which is not limited to speech coding.

1.2.2 Speaker Recognition

Speaker recognition is the process of automatically recognizing the person speaking based in individual information included in speech waves. In this thesis, speaker recognition is taken to be a general process whereas *speaker identification* and *speaker verification* refer to specific tasks or decision modes associated with this process. In speaker identification (ID), the goal is to determine which voice in a known group of voices best matches the speaker. In speaker verification, the goal is to determine if the speaker is whom he or she claims to be. The fundamental difference between these two modes is the number of decision alternatives.

Some of the important applications of speaker recognition include secure access control by voice, customizing services or information to individuals by voice, indexing or labelling speakers in recorded conversations or dialogues, surveillance, and criminal and forensic investigation involving recorded voice samples. Among them, the most frequently mentioned application nowadays is access control. Access control applications include voice dialing, banking transactions over telephone network, access to bank accounts through

telephones, control on the use of credit cards, telephone shopping, database access services, information and reservation services, voice mail, and remote control to computers. Another potentially important application of speaker recognition technology is its use for forensic purpose. Speaker recognition is described in detail in [20, 117].

Speaker recognition consists of two stages, namely, *feature extraction* and *classification* as shown in Fig. 1.2. Feature extraction is associated with deriving the characteristic patterns of the signal that are representative of a given speaker. The parameters or features used in speaker recognition are a transformation of the speech signal into a compact acoustic representation that contains information useful for the identification of the speaker. This is often done using short-term spectral analysis. The classifier uses these features to render a decision as to the speaker identity or verifies the claimed identity of the speaker. There have existed many classification techniques proposed in the literature for speaker recognition, such as Dynamic Time Warping (DTW), Vector Quantization (VQ), Hidden Markov Modeling (HMM), and Gaussian Mixture Modeling (GMM).

The speaker ID problem may further be subdivided into closed set and open set. The closed set speaker ID problem refers to a case where the speaker is known a priori to belong to a set of M speakers. In the open set case, the speaker may be out of the set and hence, a "none of the above" category is necessary. Another distinguishing aspect of speaker ID systems is that they can be either text-independent or text-dependent depending upon the application. In the text-independent case, there is no restriction on the sentence or phrase to be spoken, whereas in the text-dependent case, the input sentence or phrase is fixed for each speaker. A text-dependent scenario is commonly encountered in speaker verification systems in which a person's password is critical for verifying his/her identity. Text-independent recognition is more difficult but also more flexible.

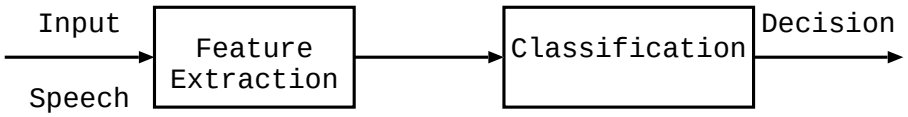


Figure 1.2: A general diagram of a recognition system.

In speech recognition, the main goal of the acoustic processing module is to extract features that are invariant to the speaker and channel characteristics and that are representative of the lexical content. In contrast, speaker recognition requires the extraction of speaker characteristic features, which may be independent of the particular words that were spoken. Such characteristics include the gross properties of the spectral envelope (such as the average formant positions overmany vowels) or the average range of fundamental frequency. Unfortunately, since these features are often difficult to estimate reliably, particularly for a short enrollment period, current systems often use acoustic parameters that have been developed for use in speech recognition. However, LPC parameters (or log power spectra), which have fallen out of favor in automatic speech recognition (ASR) because of their strong dependence on individual speaker characteristics, tend to be preferred in speaker recognition for this very reason [56].

Features derived from spectrum of speech have proven to be the most effective in automatic speaker recognition systems [119]. This is mainly attributed to the fact that different speakers have different spectra for similar sounds because the anatomical structure (vocal tract/glottis) is conveyed in speech spectrum. As presented in Section 1.1, the

temporal decomposition technique can be used to decompose spectral parameters into two elementary components, namely, event targets and event functions. Since the TD model is based on an assumption of the co-articulation in speech production systems, where the event targets may be associated with ideal articulatory positions and the event functions describe the temporal evolution of these targets [141]. It would be of interest to investigate the use of event targets as a feature set for speaker recognition. In addition, TD is promising as a means of segmenting speech into a sequence of overlapping events closely related to phonetic structure of the speech signals [141, 143]. Meanwhile, in VQ-based speaker ID, phoneme-classes and speakers are simultaneously recognized [43]. Furthermore, the VQ approach to speaker ID is not only simple, but also efficient. Therefore, the use of event targets as a set of features for speaker recognition should be first explored in VQ-based speaker ID systems.

1.2.3 Voice Transformation and Voice Conversion

The speech signal conveys several levels of information. Primarily, the speech signal conveys the meaning of the message being uttered, but on a secondary level, the signal also conveys the identity of a speaker. *Voice transformation* techniques attempt to transform the speech signal uttered by a given speaker so as to alter the characteristics of his/her voice. This notion of voice transformation is related to, but distinct from research on *voice conversion* [101]. The goal of voice conversion is to modify the speech of a source speaker so that it sounds as if it were uttered by a target speaker. In this thesis, the terms transformation and conversion will be used interchangeably.

The individuality of voices plays an important part in our daily communication. For example, it allows us to differentiate between speakers in a conference call or on a radio program. Consequently, voice transformation technology has many applications in all systems that make use of prerecorded speech, such as text-to-speech (TTS) synthesizers. Today's state-of-the-art TTS systems are based on the selection and concatenation of acoustical units. In these systems, voice transformation would be a simple and efficient way to create the desired variety of voices while avoiding collecting and handling of different speakers which is known to be an extremely expensive process [101]. Another reason why the individual voice characteristics are useful is that they make it possible to identify the speaker. Voice transformation is thus an important aspect of ongoing projects in interpreted telephony [132]. Such systems would make communication between speakers of different languages easier by first recognizing the sentences uttered by each speaker, and then translating and synthesizing them in a different language. In this application, it is important for the naturalness of the conversation that the characteristics of each speaker's voice are to be maintained through the whole process. For the same reason, voice conversion techniques would also be needed in the context of speaking aids for the speech impaired [132]. Another potential application of voice conversion is in the area of low-bit-rate speech coding, where the speaker identity cannot be well preserved during transmission. For these systems, voice conversion algorithms can be used to render the decoded speech at the receiver so that it matches the speaker identity of the transmitting speaker. Finally, it is interesting to note that the voice conversion problem is closely related to other familiar speech research topics that involve speaker identity such as speaker adaptation or speaker recognition. The main difference between the latter research topics and voice conversion is that in the case of voice conversion, the final output is a speech

signal targeted for a human listener.

Previous studies in speaker recognition by humans indicate that voice individuality should be considered a consequence of combining several factors. Among these factors, suprasegmental speech characteristics such as the speaking rate, the pitch contour or the duration of the pauses have been shown to contribute greatly to speaker individuality [80]. In many cases, it also appears that specific characteristics of the perceived voice are influenced by the linguistic style of the speech. In the current state of our knowledge, the processing of such features of speech by an automatic system is difficult because high-level considerations are involved. In particular, the fact that both the meaning of the spoken message and the intention of the speaker have a strong influence on prosodic features clearly hinders their automatic processing in cases where the text of the speech utterance is not fixed a priori. Fortunately, it turns out that the average values of these features (average pitch frequency, overall speech dynamics) already carry a great deal of the speaker-specific information [80]. There is also strong evidence that distinct speakers can be efficiently discriminated at the segmental level by comparing their respective spectral envelopes [75, 43]. Accordingly, most current speaker recognition techniques are based on the characterization of the statistical distribution of the spectral envelopes [123]. It is generally admitted that the overall shape of the envelope together with the formant characteristics are the major speaker-identifying features of the spectral envelope [80]. However, some uncertainty remains about the respective contributions of these acoustics features to the individuality of the speaker's voice.

The temporal decomposition technique decomposes speech into phoneme-like events which can be considered as basic units for concatenative speech synthesis. These units can be classified according to the structure of event functions [3, 16]. Therefore, the mapping between features in the problem of voice conversion can be regarded as that between events. It is worth considering the application of TD in this application area. More interestingly, using TD we can control spectral envelopes, durations, and fundamental frequencies (F0) independently and flexibly. This suggests the potential applications of TD in many application areas, such as speaker individuality, emotional speech, song synthesis, text-to-speech (TTS) synthesis, etc. It is also worth noting here that the characteristics of spectral parameter set used as input for TD should be considered.

1.3 Motivation and Scope of the Research

As presented earlier, Atal [8] proposed a method for efficient coding of linear predictive coding (LPC) parameters based on temporal decomposition (TD) of speech. The TD model of speech is a description of speech in terms of event targets that possibly describe the ideal articulatory configurations of the successive acoustic events that occur in speech, and event functions that describe their temporal evolutionary patterns. Following Atal's initial paper [8], a number of modifications and applications have been explored. Modified algorithms for TD have been mainly proposed to overcome the drawbacks of high computational cost and the high parameter sensitivity of the number and locations of the events. Meanwhile, on the application side, the concept of TD has attracted many researchers in recent years, especially in application areas such as speech coding, speech recognition, speech segmentation, and speech synthesis.

There have been many spectral parameter sets being considered for TD such as log-

area (LA), log area ratios (LAR), cepstrum, reflection coefficients (RC), etc. However, due to the stability problems in the LPC model, not all types of parametric representations of speech can be used. This is because there is no guarantee that the selected spectral parameters are valid after the spectral transformation performed by TD. Accordingly, only the LPC parameters which can directly be tested for the system stability are used as input for TD. This results in the fact that LSF parameters have rarely been considered as a candidate spectral parameter for TD. On the other hand, the TD performance depends significantly on the type of parameters used, particularly due to the Euclidean distance measurement performed in the space of the parameters selected [142, 49].

The reason why TD of LSF is difficult mainly comes from the different characteristics of LSF parameters from the others. If consider reflection coefficients (RC), log area (LA), log-area ratios (LAR), or cepstrum parameters, they are representing some sorts of intensity. A reflection coefficient k_i represents the degree of reflection of the air flow at the ith section. A log-area coefficient LA_i represents the degree of wideness of the ith cross-sectional area. Log-area ratio is the ratio of LA to its neighboring section. Cepstrum c_i also represents the magnitude of the ith frequency component. Consequently, each component of these spectral parameter vectors takes no value restriction with respect to other components. Therefore, it is intuitively acceptable to add/subtract two or more intensity vectors and to decompose them into appropriately superposed component vectors. On the contrary, a LSF parameter represents a location/position, but not an intensity. Thus, it is neither addable nor subtractable, which makes them impossible for the conventional TD method.

It is worth making LSF parameters possible for TD since the LSFs are the favored format for the LPC parameter representation. The LSFs are useful because of sensitivity (an adverse alteration of one coefficient results in a spectral change only around that frequency) and efficiency (LSFs result in low spectral distortion while being interpolated and/or quantized). This does not occur with other representations [60]. As long as the LSF coefficients are ordered in the interval $(0; \pi)$, the resulting LPC filter is stable. Another desirable property of LSFs is that they are related to formants. Closer LSFs produce a sharper formant peak. This property provides a useful, practical check for the stability after the LSFs has been interpolated and/or quantized. The LSFs can be checked for a minimum spacing, and separated slightly if necessary [57].

Temporal decomposition of LSF parameters has many advantages. Firstly, it has possibilities to achieve the highest reconstruction accuracy since the LSF parameters have been found to provide the best interpolation and quantization properties over the other LPC related spectral parameters [109, 110]. Secondly, because of the localized nature of LSFs and the concept of TD, it has desirable properties to be applied in voice modification. Finally, since the LSF representation of speech is employed in most of current speech coders, TD of LSF parameters can be beneficially integrated into such systems to reduce the bit-rate required for encoding speech. Therefore, it is crucial to make LSF parameters possible for TD.

For the use in real-time applications, it is desirable to have a method of TD which requires short algorithmic delay and low computational cost. However, most of algorithms for TD method require more than 200 ms buffering delay which is not suitable for such kinds of applications. Moreover, they are very computationally costly, which has been mainly attributed to the use of the singular value decomposition (SVD) routine and the Gauss-Seidel iterations. It is expected to reduce the algorithmic delay and computational

cost required for TD analysis.

On the application side, the primitive and also major application of TD is in low-bit-rate speech coding. Despite numerous methods proposed in the literature for coding speech at very low-bit rates, producing high-quality speech at rates below 2.4 kbps is still a challenging issue. It is well-known that STRAIGHT (stands for "Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum") [67] is a very high-quality vocoder. Therefore, it is worth investigating whether a combination of STRAIGHT and the proposed TD algorithms can produce a new generation of speech coders which can give high-quality speech at very low-bit rates.

In addition to speech coding, it is of interest to find out other applications of TD, which will make it become a more powerful speech analysis technique. Since the event targets are assumed to describe the ideal articulatory targets, it is also worth investigating whether the event targets can convey speaker identity. More concretely, it is necessary to investigate the application of TD in speaker recognition. Also, as mentioned earlier, each LSF has the localized nature and using TD we can control spectral envelopes, durations, and fundamental frequencies (F0) independently and flexibly. This gives a hint for investigating the potential applications of TD of LSF parameters in many application areas such as speaker individuality, voice conversion, emotional speech, song synthesis, text-to-speech (TTS) synthesis, etc.

Being motivated by these research objectives, this thesis work has been conducted and some research results have been revealed. The major contributions presented in the thesis can be summarized as follows:

1. Development of a method for TD of line spectral frequency (LSF) parameters, called Modified Restricted Temporal Decomposition (MRTD), by improving the Restricted Temporal Decomposition (RTD) method (Chapter 4).

2. Investigation on the application of the MRTD technique to modeling and quantizing speech spectral and excitation parameters (Chapter 4).

3. Development of a very low-bit-rate speech coding scheme based on STRAIGHT using the MRTD technique (Chapter 5).

4. Investigation on the application of the MRTD technique in the feature extraction stage of the problem of speaker recognition (Chapter 6).

5. Development of a new method of voice transformation in the context of STRAIGHT based on the modification of formants in the LSF domain (Chapter 7).

6. Development of a new low-delay method for TD of LSF parameters, Limited Error Based Event Localizing Temporal Decomposition (LEBEL-TD), and investigation on the application of LEBEL-TD to variable-rate speech coding (Chapter 8).

Also, during this thesis investigation, some other findings have also been revealed.

1. Development of a well-defined evaluation procedure for the Spectral Stability Based Event Localizing Temporal Decomposition (S^2BEL-TD) method (Chapter 3).

2. Establishment of a mathematical proof for the convergence property of the iterative refinement procedure in the S^2BEL-TD method (Appendix A).

1.4 Outline of the Thesis

The rest of this thesis is organized as follows: In **Chapter 2**, a survey on temporal decomposition of speech is described. Before the survey, **Chapter 2** provides a brief introduction to linear predictive coding (LPC) analysis that is being used in most speech coders to model the short-term spectral parameters. We further discuss other alternative representations of LPC coefficients with an emphasis on line spectral frequency (LSF) parameters, where some properties of LSFs are explained in detail. Finally, the problems occurred while performing temporal decomposition of LSF parameters are pointed out.

Chapter 3 presents three algorithms for TD that have a direct impact on this thesis research: the original TD method (Atal's method) [8], the Spectral Stability Based Event Localizing Temporal Decomposition (S^2BEL-TD) method [103], and the Restricted Temporal Decomposition method for LSF parameters (RTD) [71]. Furthermore, a well-defined evaluation procedure is added to the S^2BEL-TD method to confirm its efficiency. The advantages and shortcomings of the three methods of TD are also discussed here.

In **Chapter 4**, it is indicated that the RTD method, however, has not ensured the LSF ordering property for the event targets and therefore, cannot always be applied to decomposing LSF parameters. Instead, the Modified RTD (MRTD) is proposed to overcome this drawback. Moreover, the application of the MRTD method to modeling and quantizing speech spectral and excitation parameters is also investigated.

In **Chapter 5**, a method of very low-bit-rate speech coding using MRTD is developed. This speech coder is implemented in the context of STRAIGHT. **Chapter 5** also presents subjective test results and demonstrates that the proposed speech coding method gives the speech quality close to that of the 4.8 kbps FS-1016 CELP coder at the rates of around 1.2 kbps.

Chapter 6 introduces a new application of the TD technique in speaker recognition. The event targets obtained from MRTD analysis of LSF parameters are found to be effective when applied in VQ-based speaker identification systems as a feature set. Their performance is found to be superior to that of the popular mel-frequency cepstral coefficients (MFCC) features in the case of testing on clean speech, while the number of feature vectors required for both training and testing phases has been reduced by a factor of five. It is shown that the iterative refinement of event targets has positively affected their speaker-specific information.

Chapter 7 presents a pilot study of voice transformation. Since TD of LSF parameters has the potential to be employed in many application areas, such as voice conversion, speaker individuality, and emotional speech, it is crucial to have a voice transformation system working in the LSF domain. This work-in-progress introduces a voice gender transformation method based on modification of formants in the LSF domain which can be extended to a general task of voice transformation. It is expected that TD of LSF parameters can be incorporated with this voice transformation algorithm to produce a high-quality voice conversion method and for the use in other applications relating to voice modification.

In **Chapter 8**, a low-delay method for TD of LSF parameters, called Limited Error Based Event Localizing Temporal Decomposition (LEBEL-TD), is presented. The proposed method can achieve the reconstruction accuracy comparable to other TD methods, such as S^2BEL-TD, RTD, and MRTD, with lower algorithmic delay and less computational cost. Furthermore, **Chapter 8** proposes a method of variable-rate speech coding

based on STRAIGHT using LEBEL-TD. This coder is also implemented in the context of STRAIGHT, and it can provide the speech quality comparable to that of the 4.8 kbps FS-1016 CELP coder at the rates of around 1.8 kbps. A comparison of LEBEL-TD with the conventional TD and interpolation methods is also discussed here.

Finally, **Chapter 9** summarizes the contributions and achievements of the thesis. Thesis conclusions, suggestions, and opportunities for further research are also presented.

In addition, during this thesis research, we have also studied the S^2BEL-TD algorithm and proposed several improvements to the method. In **Appendix A**, a mathematical proof of the convergence property of the iterative refinement procedure involved in the S^2BEL-TD is introduced. Some modifications are made to the original S^2BEL-TD method to improve its robustness in this respect.

Chapter 2

Research Background

2.1 Linear Predictive Modeling of Speech

To be efficient, the speech analysis at the acoustic level must take advantage of our understanding of how the speech signal is produced. Making a suitable functional model of the human vocal tract level is the first step for proper analysis of the speech waveform.

One of the most powerful speech analysis methods is that of Linear Predictive Coding or LPC analysis as it is commonly referred to [88]. The basic idea behind LPC analysis is that each speech sample can be represented as a linear combination of previous samples. Such a representation leads to a model of the short-term spectral parameters. In this section, the basics of linear prediction model, different methods of determining linear prediction parameters for speech signals, and some of different LPC parameter sets are introduced.

2.1.1 Linear Prediction Model

The human speech production process reveals that the generation of each phoneme is characterized basically by two factors: the source excitation and the vocal tract shaping. In order to model speech production we have to model these two factors. To understand the source characteristics, it is assumed that the source and the vocal tract model are independent. The vocal tract model $H(z)$ is excited by a discrete time glottal excitation signal $u(n)$ to produce the speech signal $s(n)$. During unvoiced speech, $u(n)$ is a flat spectrum noise source modelled by a random noise generator. On the other hand, during voiced speech, the excitation uses an estimate of the local pitch period to set an impulse train generator that drives a glottal pulse shaping filter. The speech production process is shown in Fig. 2.1.

The most powerful and general linear parametric model used to model the vocal tract is the autoregressive moving average (ARMA) model. In this model, a speech signal $s(n)$ is considered to be the output of a system whose input is the excitation signal $u(n)$. The speech sample $s(n)$ is modelled as a linear combination of the past outputs and the present and past inputs. This relation can be expressed in the following difference equation:

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + G \sum_{l=0}^{q} b_l u(n-l), \quad b_0 = 1, \tag{2.1}$$

where G (gain factor) and $\{a_k\}$, $\{b_l\}$ (filter coefficients) are the system parameters. The

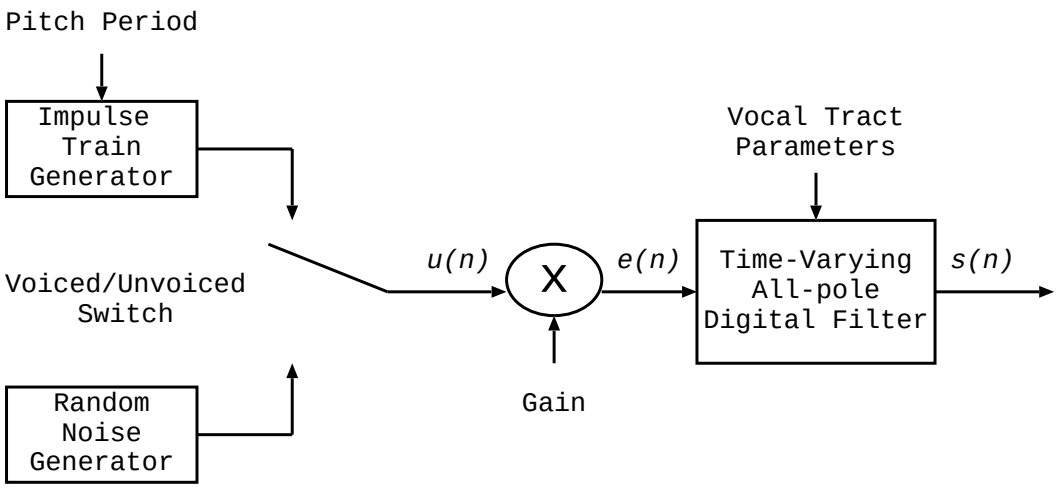


Figure 2.1: Speech production model for Linear Predictive Coding (LPC).

number p implies that the past p output samples are being considered, which is also the order of the linear prediction. The transfer function $H(z)$ of the system is obtained by applying z-transform on Equation (2.1):

$$H(z) = \frac{S(z)}{U(z)} = G\frac{1 + \sum_{l=1}^{q} b_l z^{-l}}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{2.2}$$

Clearly, $H(z)$ is a pole-zero model. The zeros represent the nasals, while the formants in a vowel spectrum are represented by the poles of $H(z)$. There are two special cases of this model:

- When $b_l = 0$, for $1 \leq l \leq q$, $H(z)$ reduces to an all-pole model, which is also known as an autoregressive model.
- When $a_k = 0$, for $1 \leq k \leq p$, $H(z)$ becomes an all-zero or moving average model.

The all-pole or autoregressive model is widely used for its simplicity and computational efficiency. It can model sounds such as vowels well enough. The zeros arise only in nasals and in unvoiced sounds like fricatives. These zeros are approximately modelled by the poles. Moreover, it is easy to solve an all-pole model. To solve a pole-zero model, it is necessary to solve a set of nonlinear equations, but in the case of an all-pole model, only a set of linear equations need to be solved.

The transfer function of the all-pole model is

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}}, \tag{2.3}$$

Actually an all-pole model is a good estimate of the pole-zero model. According to [30], any causal rational system $H(z)$ can be decomposed as

$$H(z) = G' H_{min}(z) H_{ap}(z), \tag{2.4}$$

where, G' is the gain factor, $H_{min}(z)$ is the transfer function of a minimum phase filter and $H_{ap}(z)$ is the transfer function of an all-pass filter.

Now, the minimum phase component can be expressed as an all-pole system:

$$H_{min}(z) = \frac{1}{1 - \sum_{i=1}^{I} a_i z^{-i}}, \tag{2.5}$$

where I is theoretically infinite but practically can take a value of a relatively small integer. The all-pass component contributes only to the phase. Therefore, the pole-zero model can be estimated by an all-pole model.

The inverse z-transform of Equation (2.3) is given by:

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n). \tag{2.6}$$

If the gain factor $G = 1$, then from Equation (2.3), the transfer function becomes

$$H(z) = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} = \frac{1}{A(z)}, \tag{2.7}$$

where the polynomial $(1 - \sum_{k=1}^{p} a_k z^{-k})$ is denoted by $A(z)$. The filter coefficients $\{a_k\}$ are called the LP (linear prediction) or linear predictive coding (LPC) coefficients.

A linear predictor with prediction coefficients, α_k, is defined as a system whose output is

$$\hat{s}(n) = \sum_{k=1}^{p} \alpha_k s(n-k). \tag{2.8}$$

The desired prediction coefficients, α_k, are the summation's optimal weights which obtain the closest match to the original speech. Therefore, the prediction coefficients, α_k, are estimates for the filter coefficients, a_k.

The error signal $e(n)$ is the difference between the input speech and the estimated speech. Thus, the following relation holds:

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^{p} \alpha_k s(n-k). \tag{2.9}$$

In the z-domain it is equivalent to

$$E(z) = S(z)A(z), \tag{2.10}$$

Now, the whole model can be decomposed into the following two parts, the analysis part and the synthesis part (see Fig. 2.2).

The analysis part analyzes the speech signal and produces the error signal. The synthesis part takes the error signal as an input. The input is filtered by the synthesis filter $1/A(z)$, and the output is the speech signal. The error signal ($e(n)$) is sometimes called the residual signal or the excitation signal. If the error signal from the analysis part is not used in synthesis, or if the synthesis filter is not exactly the inverse of the analysis filter, the synthesized speech signal will not be the same as the original signal. To differentiate between the two signals, we use the notation $\hat{s}(n)$ for the synthesized speech signal.

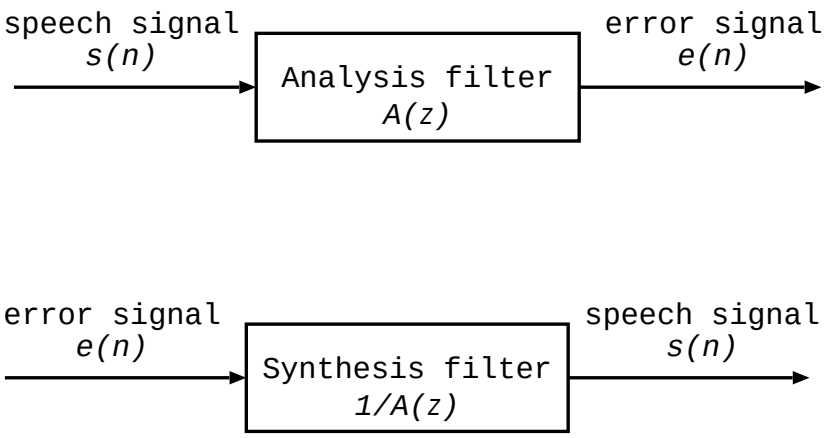


Figure 2.2: LPC analysis and synthesis model.

2.1.2 Estimation of LPC Coefficients

There are two widely used methods for estimating the LPC coefficients:

- Autocorrelation.
- Covariance.

Both methods choose the short-term filter coefficients (LPC coefficients) $\{a_k\}$ in such a way that the residual energy (the energy in the error signal) is minimized. The classical least square technique is used for that purpose.

Windowing

Speech is a time varying signal, and some variations are random. Usually during slow speech, the vocal tract shape and excitation type do not change in 200 ms. But phonemes have an average duration of 80 ms. Most changes occur more frequently than the 200 ms time interval [124]. Signal analysis assumes that the properties of a signal usually change relatively slowly with time. This allows for short-term analysis of a signal. The signal is divided into successive segments, analysis is done on these segments, and some dynamic parameters are extracted. The signal $s(n)$ is multiplied by a fixed length analysis window $w(n)$ to extract a particular segment at a time. This is called windowing. Choosing the right shape of window is very important, because it allows different samples to be weighted differently. The simplest analysis window is a rectangular window of length N_w:

$$w(n) = \begin{cases} 1, & \text{if } 0 \leq n \leq N_w - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{2.11}$$

Truncating the speech signal is not sufficient, as truncating introduces discontinuities at the beginning and the end of the interval, which subsequently introduce poor side-lobe structure. Therefore, the aim of the window function, $w(n)$, is to taper the signal to zero so as to minimize the signal discontinuities. A typical window function if the Hamming window, which has the form:

$$w(n) = \begin{cases} 0.54 - 0.46\cos(2\pi\frac{n}{N_w-1}), & \text{if } 0 \le n \le N_w - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{2.12}$$

The Hanning window of Equation (2.13) also has a similar form:

$$w(n) = \begin{cases} 0.5 - 0.5\cos(2\pi\frac{n}{N_w-1}), & \text{if } 0 \le n \le N_w - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{2.13}$$

There are other types of tapered windows, such as the Blackman, Kaiser and the Bartlett window. Additionally, a window can also be hybrid.

Although the Hamming (also Hanning) window provides improved side-lobe behavior, it does so at the expenses of broadening the main-lobe of the spectral estimator. Therefore, to maintain the resolution properties that are needed to justify representing the speech spectral properties using linear prediction, the window width must be at least $2\frac{1}{2}$ times the average pitch period.

Usually, the successive windows are chosen to overlap, and the distance between successive windows is called the *frame shift* or *frame period*. The smaller the frame period the better the quality, because we are better able to capture the transitions of the speech signals [112].

Autocorrelation Method

At first the speech signal $s(n)$ is multiplied by a window $w(n)$ to get a windowed speech segment $s_w(n)$, where,

$$s_w(n) = w(n)s(n) \tag{2.14}$$

The next step is to minimize the energy in the residual signal. The residual energy E is defined as follows:

$$E = \sum_{n=-\infty}^{\infty} e^2(n) = \sum_{n=-\infty}^{\infty} \left(s_w(n) - \sum_{k=1}^{p} \alpha_k s_w(n-k) \right)^2 \tag{2.15}$$

The values of α_k that minimize E are found by assigning the partial derivatives of E with respect to α_k to zeros. If we set $\frac{\partial E}{\partial \alpha_k} = 0$, for $k = 1, \ldots, p$, we get p equations with p unknown variables α_k as shown below:

$$\sum_{k=1}^{p} \alpha_k \sum_{n=-\infty}^{\infty} s_w(n-i)s_w(n-k) = \sum_{n=-\infty}^{\infty} s_w(n-i)s_w(n), \quad 1 \le i \le p. \tag{2.16}$$

In Equation (2.16), the windowed speech signal $s_w(n) = 0$ outside the window $w(n)$. The linear equations can be expressed in terms of the autocorrelation function. This is because the autocorrelation function of the windowed segment $s_w(n)$ is defined as

$$R(i) = \sum_{n=i}^{N_w-1} s_w(n)s_w(n-i), \quad 1 \le i \le p, \tag{2.17}$$

where N_w is the length of the window. The autocorrelation function is an even function, where $R(i) = R(-i)$. By substituting the values from Equation (2.17) in Equation (2.16), we get

$$\sum_{k=1}^{p} R(\mid i-k \mid)\alpha_k = R(i), \quad 1 \le i \le p. \tag{2.18}$$

The set of linear equations can be represented in the following matrix form:

$$\begin{bmatrix} R(0) & R(1) & \cdots & R(p-1) \\ R(1) & R(0) & \cdots & R(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ R(p-1) & R(p-2) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(p) \end{bmatrix}. \tag{2.19}$$

Equation (2.19) can be expressed as

$$\mathbf{R}\boldsymbol{\alpha} = \mathbf{r}. \tag{2.20}$$

The resulting matrix has the property that it is symmetric and all elements along a given diagonal are equal, i.e. it is a Toeplitz matrix. This allows the linear equations to be solved by some very efficient recursive procedures. The most widely used is perhaps Durbin's algorithm [88]. Because of the Toeplitz structure of $\mathbf{R}$, $A(z)$ is minimum phase. At the synthesis filter $H(z) = 1/A(z)$, the zeros of $A(z)$ become the poles of $H(z)$. Thus, the minimum phase of $A(z)$ guarantees the stability of $H(z)$.

Covariance Method

The covariance method is very similar to the autocorrelation method. The basic difference is the placement of the analysis window. The covariance method windows the error signal instead of the original speech signal. The energy E of the windowed error signal is

$$E = \sum_{n=-\infty}^{\infty} e_w^2(n) = \sum_{n=-\infty}^{\infty} e^2(n)w(n). \tag{2.21}$$

If we assign the partial derivatives $\frac{\partial E}{\partial \alpha_k}$ to zero, for $1 \le k \le p$, we have the following p linear equations:

$$\sum_{k=1}^{p} \phi(i,k)\alpha_k = \phi(i,0), \quad 1 \le i \le p, \tag{2.22}$$

where $\phi(i,k)$ is the covariance function of $s(n)$ which is defined as

$$\phi(i,k) = \sum_{n=-\infty}^{\infty} w(n)s(n-i)s(n-k). \tag{2.23}$$

The above equation can be expressed in the following matrix form:

$$\begin{bmatrix} \phi(1,1) & \phi(1,2) & \cdots & \phi(1,p) \\ \phi(2,1) & \phi(2,2) & \cdots & \phi(2,p) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(p,1) & \phi(p,2) & \cdots & \phi(p,p) \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \varphi(1) \\ \varphi(2) \\ \vdots \\ \varphi(p) \end{bmatrix}. \tag{2.24}$$

where $\varphi(i) = \phi(i,0)$ for $i = 1, 2, \ldots p$. Equation (2.24) can be written as

$$\boldsymbol{\phi}\mathbf{a} = \boldsymbol{\varphi}. \tag{2.25}$$

The resulting covariance matrix is symmetric but not Toeplitz, and is therefore not guaranteed to be invertible. It is possible that the above system of equations does not have a solution, in which case the LPC filter is unstable.

2.1.3 Repsentations of LPC Parameters

Linear predictive coding coefficients (LPC coefficients) have other representations: line spectral frequencies (LSF), reflection coefficients (RC), autocorrelations (AC), log area ratios (LAR), arcsine of reflection coefficients (ASRC), impulse responses of LP synthesis filter (IR), etc. They effectively have a one-to-one relationship with the LPC coefficients, and they preserve all the information from the LPC coefficients. Among them, some are computationally efficient. Some of them have special features which make them attractive for different purposes. That is why a good understanding of those representations and their features is needed prior to further processing.

Line Spectral Frequency (LSF)

Line spectral frequencies (LSFs) are an alternative representation to the LPC parameters. It was found that the LPC parameters have a large dynamic range of values, so they are not good for quantization. The LSFs, on the other hand, have a well behaved dynamic range. Also, it is well-known that the LSF parameters are changed smoothly along with time as shown in Fig. 2.3, so they are good for interpolation. If interpolation is done in the LSF domain, it is easier to guarantee the stability of the resulting synthesis filter thanks to the so-called ordering property of LSFs (or the LSF ordering property). This property is that the LSFs are ordered in a bounded range from 0 to π in radians, which is a necessary and efficient condition for the stability of the corresponding LPC synthesis filter. If the LPC coefficients are encoded as LSFs, we do not need to spend the same number of bits for each LSF. This is because higher LSFs correspond to the high frequency components and high frequency components have less effect in speech perception. So higher LSFs can be quantized using fewer bits than lower LSFs. This reduces the bit rate while keeping the speech quality almost the same. LSFs have a frequency domain interpretation. Usually the LSFs are more concentrated around formants. The bandwidth of a given formant is dependent on the closeness of corresponding LSFs [109] as illustrated in Fig. 2.4. Moreover, spectral sensitivity of each LSF is localized. A change in a LSF causes changes in power spectrum near its neighborhood as shown in Fig. 2.5. Another property of LSFs is that the LSFs of order p are interlaced with those of order $p-1$. Proof of this property can be found in [96]. This inter-model interlacing theorem provides a tight bound on the formant frequency region [68].

Calculation of Line Spectral Frequencies

It has been previously mentioned that the prediction error filter or the LPC analysis filter $A(z)$ can be expressed in terms of the LPC coefficients (direct form predictor coefficients) a_k in the following form:

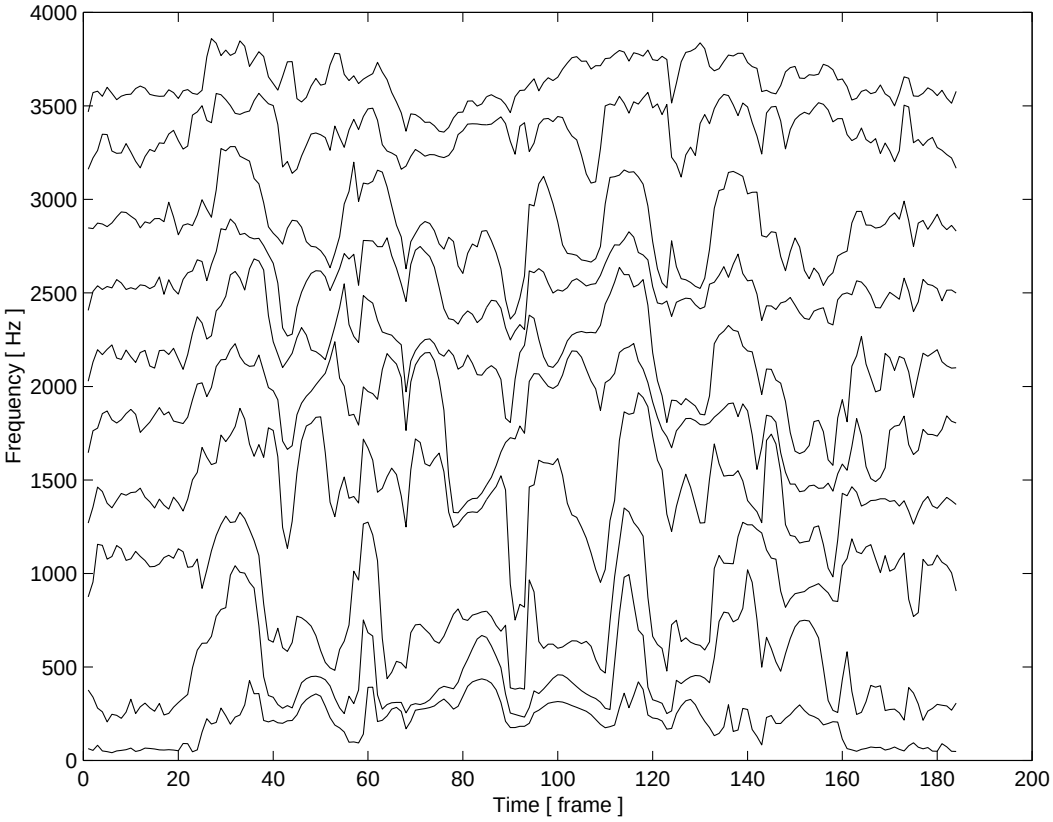


Figure 2.3: Example of a LSF parameter vector trajectory.

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k}. \tag{2.26}$$

Clearly the order of $A(z)$ is p. The $(1+p)$th order symmetric and antisymmetric polynomial $P(z)$ and $Q(z)$ can be obtained from $A(z)$:

$$\begin{aligned} P(z) &= A(z) + z^{-(p+1)} A(z-1), & (2.27) \\ Q(z) &= A(z) - z^{-(p+1)} A(z-1), & (2.28) \end{aligned}$$

where,

$$A(z) = \frac{1}{2}[P(z) + Q(z)]. \tag{2.29}$$

There are three important properties of $P(z)$ and $Q(z)$ [128]:

- All the roots of $P(z)$ and $Q(z)$ polynomials are on the unit circle.
- Roots of $P(z)$ and $Q(z)$ are interlaced.
- The minimum phase property of $A(z)$ can be preserved, if the first two properties are intact after quantization or interpolation.

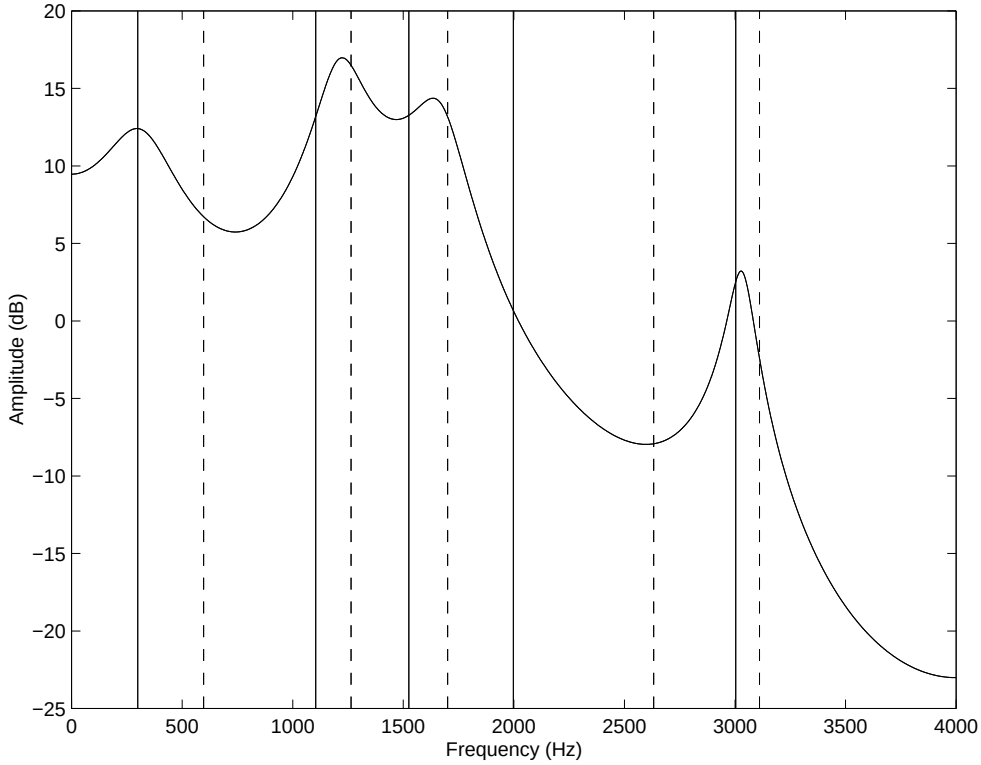


Figure 2.4: Positions of LSFs in log power spectra: the vertical lines indicate the positions of LSFs. LSFs are at 299Hz, 1104Hz, 1998Hz, 3003Hz for $P(z)$ and at 597Hz, 1264Hz, 1701Hz, 2632Hz, 3111Hz for $Q(z)$.

From the first property, we see that the roots of $P(z)$ and $Q(z)$ can be expressed in terms of w_i (as e^{jw_i}). These w_i are called the LSFs. The polynomials $P(z)$ and $Q(z)$ have two roots at $z = 1, z = -1$. Let us define two new polynomials $N_1(z)$ and $N_2(z)$ which have the same roots as $P(z)$ and $Q(z)$, except they do not have roots at $z = 1, z = -1$.

$$N_1(z) = \begin{cases} \frac{P(z)}{1+z^{-1}}, & \text{for } p \text{ even}, \\ P(z), & \text{for } p \text{ odd}. \end{cases} \tag{2.30}$$

$$N_2(z) = \begin{cases} \frac{P(z)}{1-z^{-1}}, & \text{for } p \text{ even}, \\ \frac{P(z)}{1-z^{-2}}, & \text{for } p \text{ odd}. \end{cases} \tag{2.31}$$

From Equation (2.30) and Equation (2.31), it is obvious that both $N_1(z)$ and $N_2(z)$ have even order, and they are symmetric. The roots occur as complex conjugate pairs, so only the roots on the upper semi-circle are to be calculated. Let the order of $N_1(z)$ and $N_2(z)$ be $2m$ and $2n$, respectively. Then

$$m = \begin{cases} \frac{p}{2}, & \text{for } p \text{ even}, \\ \frac{p+1}{2}, & \text{for } p \text{ odd}. \end{cases} \tag{2.32}$$

$$n = \begin{cases} \frac{p}{2}, & \text{for } p \text{ even}, \\ \frac{p-1}{2}, & \text{for } p \text{ odd}. \end{cases} \tag{2.33}$$

Which implies,

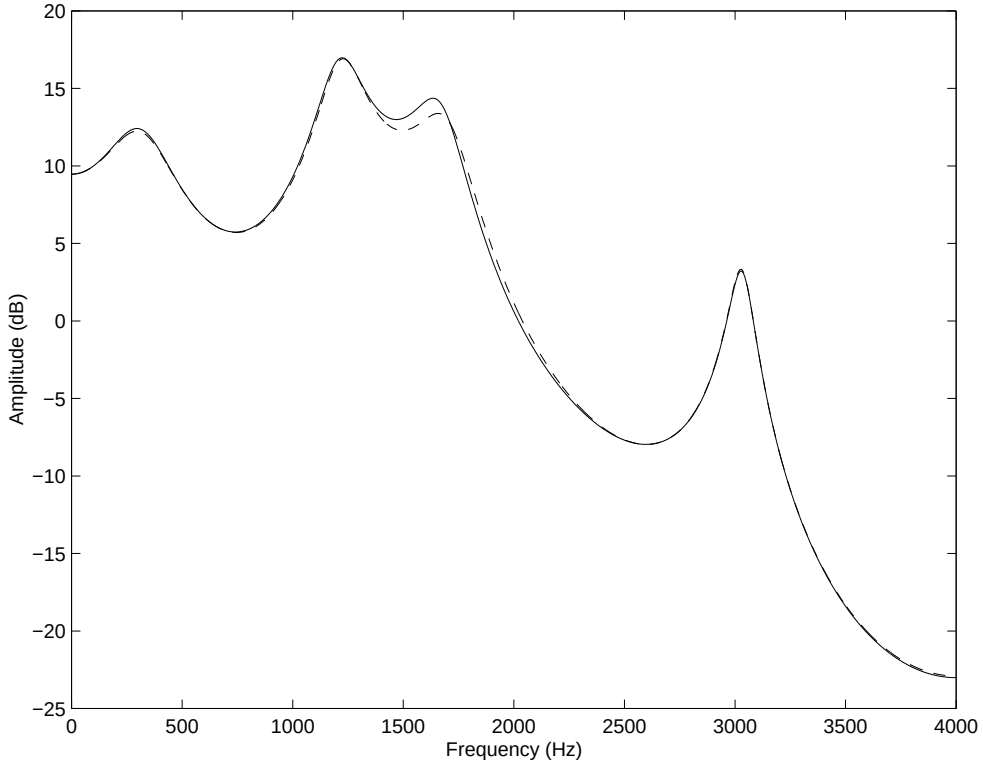


Figure 2.5: Frequency localization property of LSFs. The solid line indicates the original spectrum as given in Fig. 2.4, and the dashed line indicates the effect of changing the LSF at 1701Hz to 1720Hz.

$$N_1(z) = 1 + N_1(1)z^{-1} + \cdots + N_1(m)z^{-m} + \cdots + N_1(1)z^{-(2m-1)} + z^{-2m}, \quad (2.34)$$
$$N_2(z) = 1 + N_2(1)z^{-1} + \cdots + N_2(n)z^{-n} + \cdots + N_2(1)z^{-(2n-1)} + z^{-2n}. \quad (2.35)$$

From Equation (2.34) and Equation (2.35)

$$N_1(e^{jw}) = e^{-jwm}N_1'(w), \quad (2.36)$$
$$N_2(e^{jw}) = e^{-jwn}N_2'(w), \quad (2.37)$$

where,

$$N_1'(z) = 2\cos mw + 2N_1(1)\cos(m-1)w + \cdots + N_1(m), \quad (2.38)$$
$$N_2'(z) = 2\cos nw + 2N_2(1)\cos(n-1)w + \cdots + N_2(n). \quad (2.39)$$

Soong and Juang [128, 130] proposed a numerical method with a direct calculation of the discrete cosine transform to find the roots of $N_1'(w)$ and $N_2'(w)$. The roots of $N_1'(w)$ and $N_2'(w)$ are the LSFs. Kabal and Ramachandran [66] use an expansion of the mth order Chebyshev polynomial in x:

$$T_m(x) = \cos(mw), \quad (2.40)$$

where $T_m(x) = 2xT_{m-1}(x) + T_{m-2}(x)$. Now, $N_1'(w)$ and $N_2'(w)$ become

$$N_1'(x) = 2T_m(x) + 2N_1(1)T_{m-1}(x) + \cdots + N_1(m), \quad (2.41)$$
$$N_2'(x) = 2T_n(x) + 2N_2(1)T_{n-1}(x) + \cdots + N_2(n). \quad (2.42)$$

The roots of the expanded polynomials are determined iteratively by looking at the sign changes in the range $[-1, 1]$ and then the LSFs are found by using $w = \cos^{-1}(x)$.

Reflection Coefficients (RC)

The reflection coefficients can be computed from the filter coefficients using the following recursive formulae.

$$a_j^p = \alpha_j, \quad 1 \le j \le p \quad (2.43)$$

For i = p, p-1, ..., 1

$$a_j^{i-1} = \frac{a_j^i + a_i^i a_{i-j}^i}{1 - k_i^2}, \quad 1 \le j \le i-1 \quad (2.44)$$
$$k_{i-1} = a_{i-1}^{i-1} \quad (2.45)$$

These coefficients are physically realized in the cylindrical acoustic tube model of speech – a system of connected, hard-walled, lossless tubes through which a wave travels in one dimension. Because of this representation, they are commonly referred to as the reflection coefficients. In addition, they are also called the PARCOR (partial correlation) coefficients since they are intermediate products in determining the filter coefficients.

For a stable filter, the reflection coefficients fall in the range $-1 < k_i < 1$ for $i = 1, \ldots, p$. This is a very important condition because, by making sure that the k_i are between ± 1, even after quantization, the stability of the filter is guaranteed.

Log-Area Ratios (LAR)

It can be shown that when the reflection coefficients take values close to the boundaries ± 1, the log power spectrum becomes very sensitive to quantization error. One way to reduce the sensitivity is to warp the scale of the parameters with the transformation

$$LAR_i = \log \frac{1 + k_i}{1 - k_i}, \quad 1 \le i \le p. \quad (2.46)$$

Uniform quantization of the log-area ratios corresponds to a non-uniform quantization of the reflection coefficients.

The log-area ratios are so called because they represents the ratio of the cross-sectional area of adjacent sections in the cylindrical acoustic tube model of speech.

Log-Area Coefficients (LA)

The log-area coefficients represent the logarithm of the area of the vocal tract. They can be obtained by the following formulae:

$$A_{p+1} = 1, \tag{2.47}$$

$$A_i = A_{i+1}\left(\frac{1+k_i}{1-k_i}\right), \quad 1 \leq i \leq p. \tag{2.48}$$

$$LA_i = \log(A_i) \tag{2.49}$$

Cepstrum Coefficients

The cepstrum coefficients, which are the coefficients of the Fourier transform representation of the log magnitude spectrum, have been shown to be a more robust, reliable parameter set than the others.
A smoothed representation of the cepstrum coefficients, c_i, can be directly derived from the filter coefficients using the following formulae:

$$c_0 = \log(\sigma^2), \tag{2.50}$$

$$c_i = \alpha_i + \sum_{j=1}^{i-1}\left(\frac{j}{i}\right)c_j\alpha_{i-j}, \quad 1 \leq i \leq p, \tag{2.51}$$

$$c_i = \sum_{j=1}^{i-1}\left(\frac{j}{i}\right)c_j\alpha_{i-j}, \tag{2.52}$$

where σ^2 is the gain term of the LPC model. The coefficients are the cepstrum of impulse response of the linear predictive model, rather than the cepstrum of the original speech.

2.1.4 Limitations of the LPC Model

LPC analysis provides a simple and efficient method of representing the short-term spectral envelope of speech by a small number of parameters. However, the method is heavily based on an idealized model of the human speech production system.

The excitation source $u(n)$ in the LPC model of Fig. 2.1 relies on accurate classification of speech into either voiced or unvoiced segments. This is a task which is generally difficult to achieve in practice. There are more than two modes in which the vocal tract is excited and often these modes are mixed. Additionally, there are regions where it is not clear what the pitch period is given the fact that the speech is voiced.

Improving the excitation source has been successful, however, in most cases they have come at the expense of bit-rate. However, in most methods any simplification of the excitation model performs poorly in adverse conditions. In these circumstances, determining periodic behavior is extremely difficult.

2.2 STRAIGHT

2.2.1 Outline of STRAIGHT

The source-filter model has been found to produce synthetic speech of very high quality. However, for such systems to attain this high quality the parameters need careful attention

and hand-editing [151]. The STRAIGHT system [67], based on the source-filter model, allows flexible control of speech parameters and its conceptual simplicity has made this system a powerful tool for speech perception research as well as other speech research applications.

STRAIGHT (stands for "Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum") is a very high-quality vocoder [67]. By vocoder (voice coder) we mean a speech-specific coder that relies on speech models and is focussed upon producing perceptually intelligible speech without necessarily matching the waveform [131]. STRAIGHT has been developed to enable the real-time manipulation of speech parameters. This method uses a pitch-adaptive spectral analysis combined with a surface reconstruction method in the time-frequency region. The method also consists of a fundamental frequency (F0) extraction using instantaneous frequency calculation based on a new concept called "fundamentalness." STRAIGHT can preserve the details of time-frequency surfaces while almost perfectly removing fine structures due to signal periodicity. The method independently allows for over 600 % manipulation of such speech parameters as pitch, vocal tract length, and speaking rate, without introducing further degradation due to parameter manipulation.

STRAIGHT consists of three key concepts. The first is a time-frequency surface reconstruction method, which is based on the cardinal B-spline smoothing kernel from which the name "STRAIGHT" is derived. The second is a reliable and accurate F0 extraction method for guiding the pitch-adaptive analysis on which the STRAIGHT is based: TEMPO. TEMPO stands for "Time-domain Excitation extraction based on a Minimum Perturbation Operator." It extracts F0 as the instantaneous frequency of the fundamental component of complex sounds like voiced speech by using a new concept of "fundamentalness." Fundamentalness is defined as the negative log of the total AM (amplitude modulation) and FM (frequency modulation) magnitude for a wavelet transform using an auditory-like analyzing wavelet. However, it should be noted that the concept of "fundamentalness" has been replaced by a fixed-point analysis in the recent versions of STRAIGHT [151]. The third is an excitation source design based on the phase manipulation of all-pass filters, and it is called SPIKES (Synthetic Phase Impulse for Keeping Equivalent Sound). This new design procedure of all-pass filters reduces the characteristic degradation caused by usual pulse excitation, which can be annoying, especially when using headphones. For a detailed mathematical treatment, the reader is referred to [67].

There are two main reasons for using STRAIGHT in this thesis research. First, STRAIGHT is an extremely high-quality speech analysis/modification/synthesis method [67]. This method has been successfully applied in many speech application areas. However, there have been very few studies investigating its usefulness in speech coding. Second, the method is said to be characterized by smoothed spectrum representation and fine flexible controlling of excitation. The spectrum derived from STRAIGHT is very smooth thanks to a time-frequency interpolation procedure. It follows that the LSF parameters extracted from the spectrogram are correlated among frames, and thus the corresponding LSF contours are smooth also. It is not the case of a normal LP analysis method, where LSF parameters are extracted independently on a frame-by-frame basis.

It is worth noting that there are four versions of the STRAIGHT system that have been developed so far as follows: versions 14, 17, 23, and 30. Many researchers and developers using this system have their own preferred versions of the system which they believe produce the best quality speech [151]. In this thesis research, the STRAIGHT version 17

is employed because of its ease of use and the high-quality speech it can produce.

2.2.2 Derivation of LSF Parameters after STRAIGHT Analysis

The amplitude spectrum $X[m]$, where $0 \leq m \leq \frac{M}{2}$ with M is the number of samples in the frequency domain, obtained from STRAIGHT analysis is transformed to the power spectrum using Equation (2.53).

$$S[m] = \mid X[m] \mid^2, \quad 0 \leq m \leq \frac{M}{2} \tag{2.53}$$

The i^{th} autocorrelation coefficient, R[i], is then calculated using the inverse Fourier transform of the power spectrum as follows:

$$R[i] = \frac{1}{M} \sum_{m=0}^{M-1} S[m] \exp\{j\frac{2\pi mi}{M}\} \tag{2.54}$$

where $S[m] = S[M-m]$ and $0 \leq i \leq M-1$. Assume that the speech samples can be estimated by a P-th order all-pole model, where $0 < P < M$, the reconstruction error is calculated as given in Equation (2.55).

$$P_L = R[0] - \sum_{l=1}^{P} a_l^P R[l] \tag{2.55}$$

where $\{a_l^P\}$, $l = 1, 2 \cdots P$, are the corresponding linear predictive coding (LPC) coefficients. P_L hereafter is referred to as gain. By minimizing P_L with respect to a_l^P, where $l = 1, 2 \cdots P$, a_l^Ps could be evaluated. They are then transformed to the LSF parameters.

2.3 Brief History of Temporal Decomposition

In 1983, a new technique for efficient coding of linear predictive coding (LPC) parameters, i.e. spectral parameters, was introduced by Atal [8]. Considering that speech events do not occur at uniformly spaced time intervals and that articulatory movements are sometimes fast, sometimes slow, he concluded that uniform time sampling of speech parameters is not efficient. Thus, he proposed the temporal decomposition (TD) to represent the continuous variation of these parameters as a linear-weighted sum of a number of discrete elementary components. In other words, the observed spectral parameter vectors are approximated by a linear combination of a number of vectors of the same dimension called event targets. The interpolation functions used in this approximation are later referred to as event functions. The computational procedure of Atal's method includes first expressing the event functions as a linear combination of orthogonal functions using singular value decomposition (SVD) of the spectral parameter matrix. Then the event functions are computed based on the minimization of a compactness measure of event functions. The event targets are then calculated by minimizing the mean squared error between the original and reconstructed spectral parameters. Although this method was based on articulatory considerations, Atal did not attempt to interpret the possibly phonetic meaning of the so-determined units. Efficient speech coding was his primary

objective. Following Atal's initial paper, a number of extensions and applications have been explored.

While the original algorithm for TD by Atal works quite well with various speech samples, it has two major drawbacks which could make it inapplicable in most applications. First, Atal's method imposes an expensive computational load on the analyzing process. Second, it is very sensitive to some trivial changes in the analysis parameters [91, 141]. To alleviate the complexity problem with the original method, a number of modifications have been proposed in the literature [3, 11, 15, 21, 34, 48, 52, 71, 103, 106, 127, 141]. The modified algorithms mostly yield more efficient solutions to the problems of finding the event locations, but almost use the same technique as that in the original algorithm to refine the event functions. In addition, many modifications still employ the SVD for orthogonalizing the matrix of spectral parameters, which is a consuming task, e.g., [141]. To alleviate the sensitivity problem with Atal's method, some solutions have also been proposed, such as [103, 141].

In [3, 11, 52, 71, 103, 106], events are located across the speech block without using the SVD. The technique proposed in [3] uses the original spectral parameters for the decomposition, hence no orthogonalization is performed. This results in highly overlapped, temporally wide event functions, which are to be modified through a decorrelation process [106]. Event localization are determined in [106] based on a spectral steadiness measure applied to the matrix of the reflection coefficients in the LPC model. Event locations are detected in [103, 71] through analysis of the time trajectories of the spectral parameters within each block of the signal. It is assumed that the event locations coincide with most-steady instants detected using a spectral transition measure. This problem is solved through dynamic programming [11, 127], as the optimal solution in the sense of distance between the approximated and the original parameter sets. These methods for TD all use the same event refinement technique used in the original method, to compute the final event function shapes. Meanwhile, the event functions in [52] have been approximated by a fixed function, e.g., Gaussian or rectangular. In the following, we give a more detailed summary of some specific investigations on TD.

Marcus and Van Lieshout [91] suggested that temporal decomposition is promising as a means of segmenting speech into a sequence of overlapping events closely related to the phonetic structure of the speech signal. They also pointed out that the original TD method by Atal [8] is very sensitive to some trivial changes in the analysis parameters. Although this had no serious consequences in using the method for economical coding of speech, it caused the question of the possible validity of TD as a method for determining phonetically plausible events in speech.

Van Dijk-Kappers and Marcus [141] extended and modified the method of temporal decomposition to improve the determination of the number and locations of the events so as to overcome the sensitivity problems of the original TD method. The major modification proposed was a simple rectangular weighting factor instead of Atal's quadratic weighting factor in the compactness measure for event functions. The modification gave better results with respect to the stability of the number and locations of the events. Despite of the modifications, the computational cost has more or less remained the same because the time consuming SVD was still involved.

Van Dijk-Kappers [142] compared the performance of different speech parameter sets for temporal decomposition. Nine different parameter sets including log area, log-area ratios, reflection coefficients, and filter bank parameters have been considered as candidates

for temporal decomposition. The results revealed that log area parameters perform best in terms of reconstruction accuracy followed by log-area ratio and reflection coefficient parameters. The phonetic relevance of events has also been investigated with positive results. One major result shown was the fact that speech parameters that yield a phonetically relevant decomposition, also give the best reconstruction accuracy, even compared with decompositions which yielded much more event functions. With respect to phonetic relevance of events, filter bank parameters have been found to be the best. This has been attributed to the fact that filter bank parameters have amplitude information integrated in the parameters.

Niranjan and Fallside [106] gave a geometric interpretation of the temporal decomposition results. They viewed TD as a breakpoint analysis procedure in a multi-dimensional spaces, where the basic "units" between breakpoints are two dimensional subspace (or planes). This interpretation assumes that only two event functions, which are adjacent in time, overlap at any instant of time. This assumption resulted in a simplified model of TD which was later referred to as the second order TD model by Athaudage et al. [11]. The second order TD model was employed in combination of a dynamic programming-based optimization strategy by Shiraki and Honda in [127], and by Athaudage et al. in the so-called Optimized Temporal Decomposition (OTD) method [11]. Dix and Bloothooft [32, 34] imposed an additional constraint on the event functions in the second order TD model that all event functions at any instant of time sum up to one. A geometric argument on this restriction was provided in [32, 34], which replaced the planes by straight line segments as a means of connecting between breakpoints. This restricted second order TD model was later used in [70, 71] by Kim and Oh, however, without any explicit explanations.

The method proposed in [52] for TD, Hierarchical Temporal Decomposition (HTD), by Ghaemmaghami et al. is basically different from the others in both stages of TD: detection of event locations and refinement of event functions. HTD relies on the idea of the event approximation, introduced in [48, 50], which uses a fixed function for all functions. This eliminates both computationally expensive tasks, the singular value decomposition (SVD) as in [8, 141] and the event refinement presented in most algorithms for TD. The reduction in the spectral distortion is achieved by adjusting the event locations based on minimization of parametric distance. Seven different "popular" parameter sets (excluding LSFs) have been evaluated for TD-based speech coding [49] from the view point of reconstructed speech quality. Although the a-priori assumptions on event function shapes and event locations, provides a significant advantage in terms of speech coding efficiency (i.e. compression ratio) and computational cost, however, it seems that the HTD technique sacrifices a lot in terms of parameter reconstruction accuracy, thus resulting in poor quality of synthesized speech.

Nandasena and Akagi [103] introduced a novel approach to TD, called Spectral Stability Based Event Localizing Temporal Decomposition (S^2BEL-TD). In this method, the event localization is performed based on a maximum spectral stability criterion. This overcomes the high parameter sensitivity of Atal's method. Also, S^2BEL-TD avoids the use of the time consuming SVD routine involved in Atal's method, thus resulting in a computationally simpler algorithm for TD. However, S^2BEL-TD cannot ensure the stability of the LPC synthesis filter when applied to decomposing LSF parameters. In addition, still there is no evidence that it can be applied to any speech condition.

Kim and Oh [70, 71] proposed the Restricted Temporal Decomposition (RTD) method

for LSF parameters. To this end, the LSF ordering property is considered for the event targets. This TD method employs a restricted second order TD model introduced in [32, 34]. The RTD method can achieve results comparable to S^2BEL-TD in terms of reconstruction accuracy with lower computational complexity and higher stability. Unfortunately, RTD still has not completely guaranteed the LSF ordering property for the event targets, and thus, the stability of the corresponding LPC synthesis filter has not completely been ensured [107].

On the application side, the concept of temporal decomposition of speech has attracted many researchers in recent years, specially in application areas such as speech coding, speech recognition, speech segmentation, and speech synthesis. The fact that temporal decomposition decomposes the speech parameters into two elementary components, which occur at a lower rate than the original speech parameters, gives a means of coding speech efficiently at a lower bit-rate. TD has been employed as a technique for economical speech coding by many researchers, such as Atal [8], Cheng and O'Shaughnessy [21, 22], Shiraki and Honda [127], Ghaemmaghami et al. [48]-[55], Lemma et al. [85], Kim and Oh [70, 71], Nandasena and Akagi [103]. In [122], Ritz and Burnett have found that TD is a promising approach to low-rate wide-band speech coding also. The usefulness of TD in voice storage applications has also been investigated by Athaudage et al. [11]-[14]. The strong relationship between the TD representation of speech and the speech production mechanism has provided the necessary motivation to investigate its applications in speech segmentation and speech recognition. Bailly et al. [15] have combined a TD-like algorithm and a dynamic programming technique for the alignment of the speech signal with the phonetic transcription. In [33, 34], Dix and Bloothooft have interpreted TD as a breakpoints analysis procedure in a multidimensional parameter space, where breakpoints are connected by straight line segments. The TD interpolation scheme can be viewed as a generalization of segmentation, allowing for a gradual transition from one segment towards the next. Niranjan and Fallside [106] have suggested how to use the TD model in speech recognition systems. Bimbot et al. [17] and Marteau et al. [92] did some pilot experiments on speech recognition, obtaining promising results. The use of TD in speech recognition has also been investigated by Van Dijk-Kappers and Marcus [141], Van Dijk-Kappers [142, 143], and Kim [73]. The application of TD in speech synthesis was first advocated by Chollet et al. [27] and Ahlbom et al. [3]. In [16], Bimbot et al. have used TD to encode a set of segments called polysons, which were considered as basic units in a speech synthesis system. Bimbot et al. [18] have explored the use of TD for rule-based speech synthesis, combining TD with a glottal excitation model.

2.4 Problems in TD of LSF Parameters

As presented earlier, line spectral frequency (LSF) parameters have some properties that make them more suitable for interpolation and quantization than other representations of LPC parameters. For example, the high intra-frame correlation of LSFs implied by the LSF ordering property gives the LSF parameters advantages while vector quantized. Meanwhile, the high inter-frame correlation of LSF vectors makes them vary slowly along with time axis, which is ideal for interpolation. Therefore, most of low-bit-rate speech coders employ LSF parameters to represent short-term spectral information of speech. Moreover, the LSFs have some other properties that make them suitable for many appli-

cations, in addition to speech coding. The fact that spectral sensitivity of each LSF is localized and the relation of LSFs to formants give necessary motivation to use LSFs in other applications, e.g., voice modification. Therefore, TD of LSF parameters has many advantages: (i) high reconstruction accuracy; (ii) desirable properties to be applied in voice modification; and (iii) it can be beneficially integrated into most of current speech coding systems. However, no TD algorithm reported in the literature has completely ensured that the reconstructed LSF vectors after TD analysis are valid and thus, TD of LSF parameters often causes undesirable results.

In a conventional TD method, the event targets are determined in the least mean squared error sense only. Consequently, the conventional TD methods do not consider any constraint on the input spectral parameter vectors. This results in the fact that the distribution of determined event targets are different from that of the original spectral parameter vectors, although both are in the same dimensional vector space. On the one hand, this phenomenon causes no problem for the spectral parameter sets with no restriction on the validity, e.g., LA parameters, filter bank coefficients or cepstrum parameters, and it can be considered as the idealization of the observed spectral parameter vectors as described in [142]. On the other hand, TD of LSF parameters produces some event targets which are no longer valid LSF parameters [71]. Some of them do not satisfy the ordering property of LSFs and the others lie on the unrealistic region of the parameter vector space, e.g., some of their components do not belong to the range from 0 to π.

Invalid-LSF event targets estimated from an LSF parameter vector trajectory cause three serious problems as follows: Firstly, they cause the corresponding LPC synthesis filter to be unstable because some of the reconstructed spectral parameter vectors are not LSF vectors. Secondly, they do not have their own spectra as valid LSF vectors do. Consequently, those event targets are regarded just as the numerical results, but not the idealized targets for the given vector trajectory. They also prohibit us from matching the determined events with meaningful phonetic knowledge as considered in [141]. Finally, it is impossible to utilize the advantages of the LSF parameters during quantization. In order to avoid these problems, a minimum differential LSF (dLSF) constraint can be enforced during the reconstruction of LSF vectors. However, this is an ad-hoc method and does not guarantee an optimal result with respect to the given constraint. In this context, a modified TD algorithm with some additional constraint on the event targets should be exploited.

The reason why TD of LSF causes the above mentioned problems mainly comes from the different characteristics of LSF parameters from the others. If consider reflection coefficients (RC), log area (LA), log-area ratios (LAR), or cepstrum parameters, they are representing some sorts of intensity. A reflection coefficient k_i represents the degree of reflection of the air flow at the ith section. A log-area coefficient LA_i represents the degree of wideness of the ith cross-sectional area. Log-area ratio is the ratio of LA to its neighboring section. Cepstrum c_i also represents the magnitude of the ith frequency component. Consequently, each component of these spectral parameter vectors takes no value restriction with respect to other components. Therefore, it is intuitively acceptable to add/subtract two or more intensity vectors and to decompose them into appropriately superposed component vectors. On the contrary, a LSF parameter represents a location/position, but not an intensity. Thus, it is neither addable nor subtractable, which makes them impossible for the conventional TD method.

Chapter 3

Review of Past Algorithms for Temporal Decomposition of Speech

3.1 Introduction

Temporal decomposition of speech was first proposed by Atal [8] as a method for efficient coding of LPC parameters. Suppose that a given utterance has been produced by a sequence of K movements aimed at realizing K acoustic targets. Let us denote the speech parameters corresponding to the kth target by $\mathbf{a}_k$, and the temporal evolution of this event by a function, $\phi_k(n)$. The frame number n varies between 1 and N. In temporal decomposition of speech, the observed speech parameters, $\mathbf{y}(n)$, are approximated by $\hat{\mathbf{y}}(n)$, a linear combination of event targets as follows:

$$\hat{\mathbf{y}}(n) = \sum_{k=1}^{K} \mathbf{a}_k \phi_k(n), \quad 1 \leq n \leq N \tag{3.1}$$

where,

$$\begin{aligned} \mathbf{a}_k &= [a_{1k}\ a_{2k}\ \cdots\ a_{Pk}]^T \\ \mathbf{y}(n) &= [y_1(n)\ y_2(n)\ \cdots\ y_P(n)]^T \\ \hat{\mathbf{y}}(n) &= [\hat{y}_1(n)\ \hat{y}_2(n)\ \cdots\ \hat{y}_P(n)]^T \end{aligned}$$

The superscript T on a vector or matrix means its transpose. In matrix notation Equation (3.1) can be written as

$$\hat{\mathbf{Y}} = \mathbf{A}\mathbf{\Phi} \tag{3.2}$$

where P is the dimension of the spectral parameters, $\hat{\mathbf{Y}}$ is a $P \times N$ matrix whose nth column is $\hat{\mathbf{y}}(n)$, $\mathbf{A}$ is a $P \times K$ matrix whose kth column is $\mathbf{a}_k$, and $\mathbf{\Phi}$ is a $K \times N$ matrix whose kth row is $\boldsymbol{\phi}_k$. In Equation (3.1), both the event targets and event functions are unknown and the temporal decomposition analysis involves the determination of them once the speech parameter sequence of an utterance is given.

Equations (3.1) and (3.2) are overdetermined, as only $\mathbf{Y}$ is known. To find $\mathbf{A}$ and $\mathbf{\Phi}$, $\mathbf{Y}$ is to be decomposed through orthogonalization [8, 141]. The analysis procedure, in the original TD method by Atal [8], is performed in two stages: (i) the event locations are detected using singular value decomposition (SVD); and (ii) the event functions and targets are refined using an iterative method to minimize the distance (or error) between

the reconstructed and original spectral parameters. Although the original implementation of temporal decomposition of speech [8] was mathematically solid, it is known to have the following two major drawbacks: (i) The method is computationally costly, making it impractical; and (ii) High parameter sensitivity of the number and locations of the events. In other words, they are very sensitive to some trivial changes in the analysis parameters.

The so-called Spectral Stability Based Event Localizing Temporal Decomposition (S^2BEL-TD) [103] intends to overcome the drawbacks of Atal's method by implementing TD in a simpler way, i.e. by avoiding SVD, while adopting a maximum spectral stability criterion to determine the number and locations of the events, which avoids the necessity of redundant evaluation of event functions. However, S^2BEL-TD cannot always ensure the stability of the LPC synthesis filter after spectral transformation performed on LSF parameters by using this method.

Meanwhile, the Restricted Temporal Decomposition (RTD) method [71] focuses on the problem of applying TD to analyzing LSF parameters. The LSF representation of speech is known to provide the best interpolation as well as quantization performance over other LPC-related parameters. However, due to the stability problems in the LPC synthesis filter, LSF parameters cannot be used for TD. In order to overcome this deficiency, the RTD method has been proposed so that every event target preserves the ordering property of LSFs. Unfortunately, RTD in the current form cannot always ensure the LSF ordering property of event targets and therefore, cannot always be applied to analyzing LSF parameters.

The three algorithms for TD mentioned above have a direct impact on the thesis and this chapter is devoted to their descriptions. The rest of this chapter is organized as follows: Section 3.2 gives a mathematical summary of Atal's method. This is followed by Section 3.3 which describes the computational steps of the S^2BEL-TD in detail together with an additional well-defined evaluation procedure. Moreover, Section 3.3 also reports on the use of S^2BEL-TD for the analysis of speech excitation parameters. In Section 3.4, the RTD method for line spectral frequency (LSF) parameters is presented. Finally, these three methods of TD are summarized, and their advantages as well as shortcomings are discussed in the last section.

3.2 Atal's Method of Temporal Decomposition

Atal's temporal decomposition method involves the following procedure. For a detailed mathematical treatment, the reader is referred to [8]. First, the spectral parameter matrix of a windowed speech segment of about 200-300 ms is decomposed into two orthogonal matrices and a diagonal matrix of eigenvalues, using the so-called singular value decomposition.

$$Y^T = UDV^T$$

where Y^T is the $N \times P$ spectral parameter matrix, U is a $N \times P$ orthogonal matrix, V is a $P \times P$ orthogonal matrix, and D is a diagonal matrix of eigenvalues. N and P are the number of frames in the windowed speech segment and the order of the spectral parameters, respectively. This allows the event functions to be expressed as a linear combination of a set of orthogonal functions, and also allows the number of events, M, to

be fixed in the windowed speech segment under analysis, by taking into account only the number of significant eigenvalues. Normally, a window of about 200-300 ms gives $M = 5$.

$$\phi_k(n) = \sum_{i=1}^{M} b_{ki} u_i(n)$$

where $u_i(n)$ is the element (n, i) of the matrix U and b_{ki} are a set of coefficients. Next, the nearest event function, $\phi(n)$, to the center of the windowed speech segment, $n = n_c$, is evaluated by considering the minimization of a distance measure, $\theta(n_c)$.

$$\theta(n_c) = \sqrt{\frac{\sum_{n=1}^{N}(n - n_c)^2 \phi^2(n)}{\sum_{n=1}^{N} \phi^2(n)}}$$

Minimization of $\ln(\theta(n_c))$, with respect to the coefficients b_i leads to an eigenvector problem of a matrix $R \in R^{K \times K}$.

$$R\mathbf{b} = \lambda \mathbf{b}$$

where the element (i, r) of the matrix R is given by,

$$R_{ir} = \sum_{n=1}^{N} (n - n_c)^2 u_i(n) u_r(n),$$

and $\mathbf{b}$ is the vector of coefficients b_i. The solution corresponding to the smallest eigenvalue λ provides the optimum $\mathbf{b}$.

In order to analyze a complete utterance the above procedure should be repeated with windows located at intervals through out the utterance. Atal's method requires the window to be shifted by a small interval, i.e. by a frame interval, to ensure that no event function is missed. Therefore, if the total number of windows is L, SVD and eigenvector solving should be performed L times. SVD is a highly involved computational procedure and this is known to be the major reason for the high computational complexity of the Atal's method.

Since the window is shifted at each time by a small interval, the same event function is generally found for several adjacent windows. In order to find the locations of the event functions, and to reduce the total set of event functions, a reduction algorithm based on a zero crossing criterion of a timing function, $\nu(l)$, is incorporated.

$$\nu(l) = \frac{\sum_{n=1}^{N}(n - l)\phi^2(n)}{\sum_{n=1}^{N} \phi^2(n)}$$

The function $\nu(l)$ crosses the $\nu(l) = 0$ axis from positive to negative at each location l which equals the location of one of the $\phi_k(n)$ for some k.

The spectral targets, $\mathbf{a}_k$, are determined by considering the minimization of the squared error between reconstructed and original spectral parameters, E_i, with respect to a_{ik}'s.

$$E_i = \sum_{n=1}^{N} \left(y_i(n) - \sum_{k=1}^{K} a_{ik} \phi_k(n) \right)^2, \quad 1 \leq i \leq P$$

where N and K are the total number of frames and events in the entire utterance. Finally, an iterative refinement procedure is used to improve the event function shapes and to reduce the reconstruction error. The refined set of event functions are evaluated by minimizing the reconstruction error, E_n, of spectral vectors.

$$E_n = \sum_{i=1}^{P} \left(y_i(n) - \sum_{k=1}^{K} a_{ik}\phi_k(n) \right)^2, \quad 1 \leq n \leq N$$

The resultant $\phi_k(n)$'s are used to obtain an even better estimates of the targets, $\mathbf{a}_k$'s. The procedure is repeated until both $\phi_k(n)$'s and $\mathbf{a}_k$'s converge to a set of stable values.

As described earlier, the high computational cost of Atal's method [8] has be mainly attributed to the use of the computationally costly singular value decomposition (SVD) routine, and the repeated evaluation of the event functions at short time intervals before screening out the redundant event functions using a reduction algorithm. Marcus and Van Lieshout [91] investigated the possible validity of TD as a method of determining phonetically plausible events in speech, but came out with the parameter sensitivity problem of the original method with respect to the number and locations of the event functions. In other words, they are very sensitive to some trivial changes in analysis parameters, i.e analysis window size, number of parameters retained after singular value decomposition, etc. Van Dijk-Kappers and Marcus [141] improved the TD method to make events more stable, i.e. less parameter sensitive, but the computational cost has more or less remained the same because the time consuming SVD was still involved.

3.3 Spectral Stability Based Event Localizing Temporal Decomposition

The Spectral Stability Based Event Localizing Temporal Decomposition (S^2BEL-TD) approach intends to overcome the drawbacks of the original method of Atal by implementing it in a mathematically simpler way, i.e. by avoiding SVD, while adopting a spectral stability criterion to determine the number and locations of the events. Given these number and locations, the subsequent computation of refined event targets and event functions is much less demanding than the traditional TD method. Also, this makes the number and locations of the events more parameter independent.

The S^2BEL-TD of speech involves the following three computational steps.

STEP 1: Determination of the *event targets* (first approximation).

$$\mathbf{A}^{(0)} = \left[\ \mathbf{a}_k^{(0)} \ \right]_{1 \leq k \leq K}$$

STEP 2: Determination of the *event functions* (first approximation).

$$\mathbf{\Phi}^{(0)} = \left[\ \phi_k(n)^{(0)} \ \right]_{1 \leq k \leq K,\ 1 \leq n \leq N}$$

STEP 3: Iterative refinement of *event targets* & *event functions*.

$$(\mathbf{A}^{(0)}, \mathbf{\Phi}^{(0)}) \Rightarrow (\mathbf{A}^{(1)}, \mathbf{\Phi}^{(1)}) \Rightarrow \cdots (\mathbf{A}^{(S)}, \mathbf{\Phi}^{(S)})$$

The superscript notation indicates the iteration step number. The details of the Steps 1, 2, and 3 are given in Sections 3.3.1, 3.3.2, and 3.3.3, respectively.

3.3.1 Determination of Event Targets

The determination of the first approximation of the event targets is based on a maximum spectral stability criterion. The spectrally stable points in speech are used as a hint for the locations where speech events exist. It is assumed that each acoustic event that exists in speech gives rise to a spectrally stable point in its neighborhood. Therefore, the locations of the spectrally stable points and the corresponding spectral parameter sets can be used as a good approximation to the event locations and event targets, respectively. Because of this use of points of maximum spectral stability for event detection, this approach is termed *spectral stability based event localizing* temporal decomposition.

The transition rate of the ith spectral parameter, $y_i(n)$, at the time point n is calculated as the gradient of the best fitting straight line, i.e. regression line, within the time window $[n-M, n+M]$, as given in Equation (3.3). The squared sum of these transition rates of individual spectral parameters, $y_i(n)$, where $1 \leq i \leq P$, is defined as the Spectral Feature Transition Rate (SFTR) at the time point n, and is given by Equation (3.4).

$$c_i(n) = \frac{\sum_{m=-M}^{M} m y_i(n+m)}{\sum_{m=-M}^{M} m^2}, \quad 1 \leq i \leq P \tag{3.3}$$

$$\text{SFTR}: \quad s(n) = \sum_{i=1}^{P} c_i(n)^2, \quad 1 \leq n \leq N \tag{3.4}$$

The local minima of $s(n)$ indicate the frames with maximum local spectral stability in speech, and these points are considered as the approximate locations of the events, and the corresponding spectral parameter vectors as the initial approximation of the event targets. Therefore, if the local minima of $s(n)$ are at $n_1, n_2, .., n_K$, where $n_1 < n_2 < ... < n_K$, the initial approximation of the event target matrix, $\mathbf{A}^{(0)}$, can be formed as

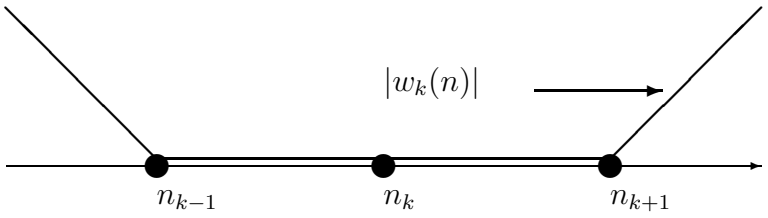
$$\begin{aligned} \mathbf{A}^{(0)} &= \left[\begin{array}{cccc} \mathbf{a}_1^{(0)} & \mathbf{a}_2^{(0)} & \cdots & \mathbf{a}_K^{(0)} \end{array} \right] \\ &= \left[\begin{array}{cccc} \mathbf{y}(n_1) & \mathbf{y}(n_2) & \cdots & \mathbf{y}(n_K) \end{array} \right] \end{aligned} \tag{3.5}$$

The number of events, K, and their locations, $n_1 < n_2 < ... < n_K$, are determined through the SFTR analysis. Therefore, the window size, $2M$, of SFTR analysis is the only parameter that effects the number and locations of the events in the S^{2}BEL-TD algorithm.

3.3.2 Determination of Event Functions

Since the speech events exist only for a limited time duration in continuous speech, event functions should be time limited. This makes it necessary to add a constraint to this effect, when evaluating them. This is achieved using a weighting function, $w_k(n)$, corresponding to each event function, $\phi_k(n)$. The weighting function $w_k(n)$ for the kth event function is defined as follows:

$$w_k(n) = \begin{cases} n_{k-1} - n, & \text{if } 1 \leq n < n_{k-1} \\ 0, & \text{if } n_{k-1} \leq n \leq n_{k+1} \\ n - n_{k+1}, & \text{if } n_{k+1} < n \leq N \end{cases}$$



$$\mathbf{w}_k = \left[\begin{array}{cccc} w_k(1) & w_k(2) & \cdots & w_k(N) \end{array} \right]$$

$$\mathbf{W} = \left(\begin{array}{c} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \cdots \\ \mathbf{w}_K \end{array} \right) \in R^{K \times N}$$

where, $\mathbf{W}$ is called the weighting function matrix. The reason for the above definition of the weighting function can be justified as follows. It is known that event function $\phi_k(n)$ exists around its center n_k. But, little is known about its length, i.e. the duration of its existence, at this stage. Since spectral properties would be less governed by the kth event in the region beyond the centers of the adjacent events, n_{k-1} and n_{k+1}, $\phi_k(n)$ should be fairly small in amplitude and gradually decreasing in this region. Therefore, $w_k(n)$ is set to zero in between the adjacent event centers, and is linearly increased beyond those points. This provides the event function total freedom to show its temporal behavior between the points of adjacent event centers, n_{k-1} and n_{k+1}, but a decreasing degree of freedom beyond those points.

Although, n_{k-1} and n_{k+1} may not be the best limits for the event function $\phi_k(n)$, they are used at this stage to evaluate the first approximation of the event functions. In Section 3.3.3 the use of adaptive weighting functions with adaptive limits for the events is described as a part of the refinement process. By considering the columns of the matrix $\mathbf{W}$, diagonal matrices are formed as

$$\mathbf{W}_n = \text{diag} \left[\begin{array}{cccc} w_1(n) & w_2(n) \cdots & w_K(n) \end{array} \right] \in R^{K \times K}$$

The functional $J\left(\boldsymbol{\phi}_n, \lambda\right)$ is formulated by taking into account the sum of the squared error between the original and the reconstructed spectral parameters, and a constraint to limit the spreading of event functions in time, as given in Equation (3.6).

$$J\left(\boldsymbol{\phi}(n), \lambda\right) = \sum_{i=1}^{P} \left(y_i(n) - \hat{y}_i(n)\right)^2 + \lambda \sum_{k=1}^{K} w_k(n)^2 \phi_k(n)^2, \quad 1 \leq n \leq N \tag{3.6}$$

where λ is a constant weighting factor and,

$$\boldsymbol{\phi}(n) = \left[\begin{array}{cccc} \phi_1(n) & \phi_2(n) & \cdots & \phi_K(n) \end{array} \right]^T, \quad 1 \leq n \leq N$$

$y_i(n)$ and $\hat{y}_i(n)$ are the i^{th} element of the spectral vectors $\mathbf{y}(n)$ and $\hat{\mathbf{y}}(n)$, respectively.

$\boldsymbol{\phi}(n)$, where $1 \leq n \leq N$, is determined by considering the minimization of the functional $J\left(\boldsymbol{\phi}(n), \lambda\right)$ with respect to $\boldsymbol{\phi}(n)$ as follows:

$$\frac{\partial J\left(\boldsymbol{\phi}(n),\lambda\right)}{\partial \phi_r(n)} = \sum_{i=1}^{P} 2\left(\sum_{k=1}^{K} a_{ik}\phi_k(n) - y_i(n)\right) a_{ir} + 2\lambda w_r(n)^2 \phi_r(n)$$
$$= 0$$
$$\sum_{i=1}^{P} a_{ir}\left(\sum_{k=1}^{K} a_{ik}\phi_k(n)\right) + \lambda w_r(n)^2 \phi_r(n) = \sum_{i=1}^{P} a_{ir} y_i(n), \qquad 1 \le r \le K \tag{3.7}$$

Conversion of Equation (3.7) into matrix notation results in

$$\mathbf{A}^T\mathbf{A}\boldsymbol{\phi}(n) + \lambda \mathbf{W}_n^T \mathbf{W}_n \boldsymbol{\phi}(n) = \mathbf{A}^T \mathbf{y}(n)$$
$$\boldsymbol{\phi}(n) = \left(\mathbf{A}^T\mathbf{A} + \lambda {\mathbf{W}_n}^T \mathbf{W}_n\right)^{-1} \mathbf{A}^T \mathbf{y}_n, \quad 1 \le n \le N \tag{3.8}$$

Therefore, the first approximation of the event function matrix, $\boldsymbol{\Phi}^{(0)}$, can be formed as

$$\boldsymbol{\Phi}^{(0)} = \begin{pmatrix} \boldsymbol{\phi}(1) & \boldsymbol{\phi}(2) & \cdots & \boldsymbol{\phi}(N) \end{pmatrix} \tag{3.9}$$

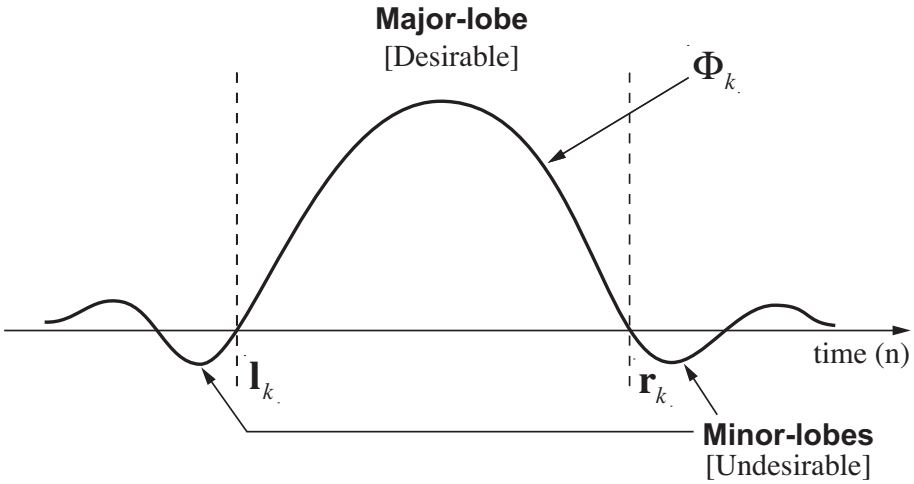


Figure 3.1: Typical shape of an initial event function. Note the presence of undesirable minor lobes, i.e. negative ripples, in addition to the desirable major lobe.

The weighting factor λ in the functional $J\left(\boldsymbol{\phi}(n),\lambda\right)$ determines the relative weighting between the two error terms involved. A suitable value for λ is to be selected, based on simulation results. This value of λ used to determine the first approximation of the event functions is referred to as $\lambda^{(0)}$ in the sections followed.

3.3.3 Iterative Refinement Procedure

An iterative refinement procedure is adopted to improve the shapes of the event functions and the reconstruction accuracy of TD, and to refine the event targets. The initial event functions show undesirable minor lobes, i.e. negative ripples, apart from the desirable major lobes as shown in Fig. 3.1. This violates the non-negativity property imposed on the event functions. The iterative refinement procedure effectively smooths-out the minor lobes while allowing the major lobes to evolve freely. It also improves the reconstruction accuracy of TD and refines the event targets. This involves the recursive performance of the procedures described in the sections followed. Generally, 4 to 5 iterations are required to shape up the event functions.

Refinement of Event Functions

Event functions are recalculated using the procedure of Section 3.3.2, but with an adaptive weighting function and the quantitative balancing of the two error-terms of the functional $J(\boldsymbol{\phi}(n), \lambda)$, as described below.

$$(\mathbf{A}^{(l-1)}, \boldsymbol{\Phi}^{(l-1)}) \rightarrow \boldsymbol{\Phi}^{(l)}, \quad 1 \leq l \leq S$$

where, l and S are the iteration step number and total number of iterations, respectively.

Adaptive Weighting function:
An adaptive weighting function is defined as given in Equation (3.10). It is adaptive to the major-lobe limits of the event functions.

$$w_k^{(l)}(n) = \begin{cases} l_k^{(l-1)} - n, & \text{if } 1 \leq n < l_k^{(l-1)} \\ 0, & \text{if } l_k^{(l-1)} \leq n \leq r_k^{(l-1)} \\ n - r_k^{(l-1)}, & \text{if } r_k^{(l-1)} < n \leq N \end{cases} \tag{3.10}$$

Where, $l_k^{(l-1)}$ and $r_k^{(l-1)}$ are the left and right limits of the major lobe of the event function $\phi_k(n)^{(l-1)}$. This definition of adaptive weighting function restricts the minor-lobes while allowing the major-lobe to evolve freely. Therefore, it gives rise to major-lobe expansion and contraction, with a simultaneous minor-lobe reduction, when the iterations are performed.

Quantitative Balancing of the functional $J(\boldsymbol{\phi}(n), \lambda)$:
Weighting factor $\lambda^{(l)}$ at the iteration step l is selected so as to balance the two error terms of the functional $J(\boldsymbol{\phi}(n), \lambda)$ using the results obtained at the iteration step $(l-1)$, i.e. $\boldsymbol{\Phi}^{(l-1)}$ and $\mathbf{A}^{(l-1)}$, as given below.

$$\lambda^{(l)} = \sigma \times \left(\frac{\sum_{n=1}^{N} \sum_{i=1}^{P} \left(y_i(n) - \hat{y}_i{}^{(l-1)}(n) \right)^2}{\sum_{n=1}^{N} \sum_{k=1}^{K} w_k^{(l)}(n)^2 \phi_k^{(l-1)}(n)^2} \right)$$

where, $\hat{y}_i{}^{(l-1)}(n) = \sum_{k=1}^{K} a_{ik}^{(l-1)} \phi_k^{(l-1)}(n)$, and σ is the constant balancing ratio.
The event functions matrix, $\boldsymbol{\Phi}^{(l)}$, at the iteration step l is calculated as follows, similarly to Equations (3.8) and (3.9).

$$\boldsymbol{\phi}(n)^{(l)} = \left(\mathbf{A}^{(l-1)^T} \mathbf{A}^{(l-1)} + \lambda^{(l)} \mathbf{W}_n^{(l)^T} \mathbf{W}_n^{(l)} \right)^{-1} \mathbf{A}^{(l-1)^T} \mathbf{y}_n, \quad 1 \leq n \leq N$$

where,

$$\mathbf{W}_n^{(l)} = \text{diag} \left[\begin{array}{cccc} w_1^{(l)}(n) & w_2^{(l)}(n) & \cdots & w_K^{(l)}(n) \end{array} \right]$$

Hence,

$$\boldsymbol{\Phi}^{(l)} = \left(\begin{array}{cccc} \boldsymbol{\phi}(1)^{(l)} & \boldsymbol{\phi}(2)^{(l)} & \cdots & \boldsymbol{\phi}(N)^{(l)} \end{array} \right)$$

Refinement of Event Targets

Refinement of event targets involves the recalculation of them by minimizing the squared error between the original and the reconstructed spectral parameters, with respect to the

target vectors. Event targets at the lth iteration are calculated from the event functions at the lth iteration, as described below.

$$\mathbf{\Phi}^{(l)} \rightarrow \mathbf{A}^{(l)}, \quad 1 \leq l \leq S$$

The squared error between the original and reconstructed ith spectral parameter at the iteration step l can be expressed as follows:

$$E_i^{(l)} = \sum_{n=1}^{N} \left(y_i(n) - \sum_{k=1}^{K} a_{ik}^{(l)} \phi_k^{(l)}(n) \right)^2, \quad 1 \leq i \leq P$$

By setting the partial derivative of $E_i^{(l)}$ with respect to a_{ir}, to zero:

$$\begin{aligned} \frac{\partial E_i^{(l)}}{\partial a_{ir}} &= \sum_{n=1}^{N} \left(y_i(n) - \sum_{k=1}^{K} a_{ik}^{(l)} \phi_k^{(l)}(n) \right) \left(-2\phi_r(n)^{(l)} \right) \\ &= 0 \end{aligned}$$

$$\sum_{k=1}^{K} a_{ik}^{(l)} \sum_{n=1}^{N} \phi_k^{(l)}(n) \phi_r^{(l)}(n) = \sum_{n=1}^{N} y_i(n) \phi_r^{(l)}(n) \tag{3.11}$$

where, $1 \leq r \leq K, \quad 1 \leq i \leq P$

Equation (3.11) gives P sets of K variable simultaneous equations, using which $a_{ik}^{(l)}$, where $1 \leq k \leq K$ and $1 \leq i \leq P$, could be evaluated. Therefore, the event target matrix at the iteration step l can be formed as follows:

$$\mathbf{A}^{(l)} = \left[\; a_{ik}^{(l)} \; \right]_{1 \leq i \leq P, \quad 1 \leq k \leq K}$$

Termination and Convergence of Iterations

The two above steps, refinement of event targets and refinement of event functions, are repeatedly performed until the minor lobe content, $MLC^{(l)}$, drops below a certain predetermined threshold level, e.g.1%. Minor lobe content, $MLC^{(l)}$,at the lth iteration step is defined as follows:

$$MLC^{(l)} = \sqrt{\frac{\sum_{k=1}^{K} \sum_{n=1}^{N} \phi_k^{(l)}(n)^2 c_k^{(l)}(n)}{\sum_{k=1}^{K} \sum_{n=1}^{N} \phi_k^{(l)}(n)^2}} \times 100\%$$

where,

$$c_k^{(l)}(n) = \begin{cases} 0, & \text{if } l_k^{(l)} \leq n \leq r_k^{(l)} \\ 1, & \text{otherwise} \end{cases}$$

where, $l_k^{(l)}$ and $r_k^{(l)}$ are the left and right limits of the major lobe of the kth event function, at the lth iteration step. Also, the root-mean-squared (RMS) error between the original and reconstructed spectral parameters, at the lth iteration step, is defined as follows:

$$E_{rms}^{(l)} = \sqrt{\frac{1}{NP} \sum_{n=1}^{N} \sum_{i=1}^{P} \left(y_i(n) - \hat{y}_i{}^{(l)}(n) \right)^2}$$

Convergence of $MLC^{(l)}$ and $E_{rms}^{(l)}$ with the iteration step number l is an important property for the iterative refinement procedure. Simulation results show that good convergence can be achieved by properly selecting the parameters $\lambda^{(0)}$ and σ.

3.3.4 Segmental S^{2}BEL-TD

The present algorithm of S^{2}BEL-TD analysis takes the total length of the input speech as a block for the TD analysis. Although there is no problem with this for word utterances and short sentence utterances, for relatively long utterances with more than about 500 frames, taking the whole utterance as a single segment for TD analysis proves time consuming. This can be simply attributed to the large dimension of the matrices involved in the computational procedure. This makes it necessary to develop the TD analysis algorithm so that it will work on short speech blocks, or segments, when analyzing a long utterance of input speech. This is termed segmental S^{2}BEL-TD analysis. On the other hand, if S^{2}BEL-TD is to be used in any kind of real time analysis, segmental analysis becomes inevitable.

The implementation of segmental analysis is based on a mutually non-interacting events criterion. Let E_i and E_j be two events with event functions $\phi_i(n)$ and $\phi_j(n)$. The indices i and j describe the chronological order of the two events E_i and E_j. The two event E_i and E_j are called mutually non-interacting if the following condition is satisfied.

$$\sum_{n=1}^{N} \phi_i(n)\phi_j(n) = 0$$

$$\text{i.e} \quad \phi_i(n)\phi_j(n) = 0, \quad 1 \leq n \leq N$$

This means that either $\phi_i(n)$ or $\phi_j(n)$ is zero at all time points n. This situation can be easily visualized as two non-overlapping event functions. Obviously, if the events E_i and E_j are separated in time by a sufficient number of intermediate events they would be mutually non-interacting. The authors were interested in the minimum l, let this be L, such that,

$$\sum_{n=1}^{N} \phi_i(n)\phi_j(n) = 0, \quad \text{if } |i-j| > L$$

By simple observation of TD results over a large set of speech data it was confirmed that $L = 3$. This means that two event functions with at least 3 intermediate events, do not overlap. Therefore, an event could be accurately evaluated without any unaccounted mutual effects, if the speech segment contains at least 3 adjacent events to both sides. In speech production point of view this may mean that the feed-forward and feed-back co-articulation do not occur over more than 3 acoustic events.

Using the above result an algorithm for the segmental TD analysis is developed as follows. Input speech is segmented with at least $2L$ events in the overlapping region between two adjacent segments. In each segment, the authors suggested to neglect the first and last L events as inaccurate due to unaccounted mutual effects, except for the first and last segment of the input speech. In the first segment, only the last L events are neglected, and in the last segment, only the first L events are neglected. The segment size is kept fixed around 100 frames.

3.3.5 Simulation Results

The ATR Japanese [2] and the TIMIT English speech database [44] were used for the speech data. Both Log Area Ratio (LAR) parameters and Line Spectral Frequency (LSF) parameters were considered as a candidate spectral parameter for the S^{2}BEL-TD. LAR

parameters have given better results, i.e. better reconstruction accuracy, in temporal decomposition [142] over the other LPC related spectral parameters. LSF parameters have been known to have the best interpolation properties [110, 26], i.e. linear combinationability. S^{2}BEL-TD was implemented on both LAR and LSF parameters and their reconstruction accuracies are compared as a part of the performance evaluation of the method. 10th order LAR and LSF parameters were calculated using a LPC analysis window of $2M = 40$ ms at 10 ms frame intervals, from 8 kHz sampling speech files.

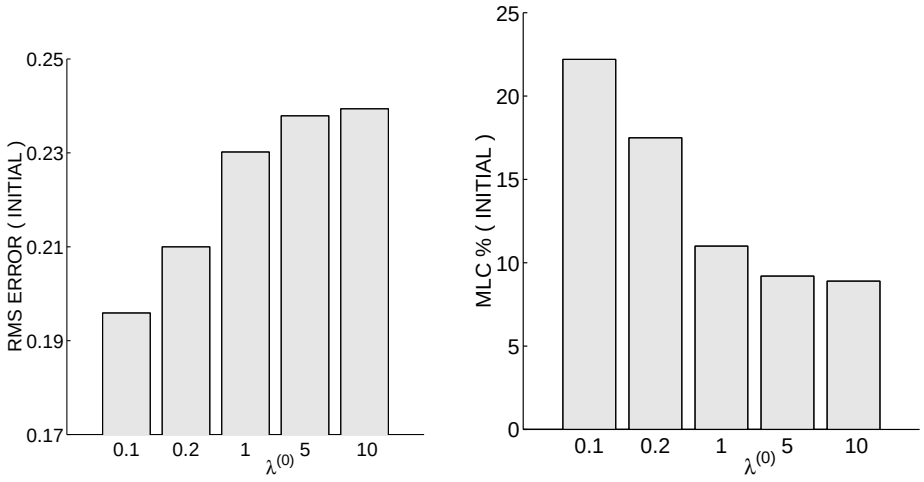


Figure 3.2: Initial RMS Error between original and reconstructed spectral parameters, $E^{(0)}_{rms}$ (left), and initial minor lobe content, $MLC^{(0)}$ (right), for different values of $\lambda^{(0)}$, as bar plots. Note that $MLC^{(0)}$ decreases, but $E^{(0)}_{rms}$ increases with increasing $\lambda^{(0)}$.

The male Japanese word utterance "*aikawarazu*" was used with a SFTR analysis window size of 40 ms to investigate the convergence properties of S^{2}BEL-TD algorithm. The spectral parameter is LAR and simulations were performed for $\lambda^{(0)}$ values of 10, 5, 1, 0.2 and 0.1. The initial minor lobe content, $MLC^{(0)}$, and the initial RMS error between reconstructed and original spectral parameters, $E^{(0)}_{rms}$, obtained for different values of $\lambda^{(0)}$ are shown in the Fig. 3.2. A high value for $\lambda^{(0)}$ causes a high reconstruction error and a relatively low $MLC^{(0)}$, while a low value for $\lambda^{(0)}$ causes a relatively low reconstruction error and a high $MLC^{(0)}$. Fig. 3.3 shows the typical shape of initial event functions, $\phi_k(n)^{(0)}$ for some k, for different values of the initial weighting factor $\lambda^{(0)}$.

The iterative refinement of the event functions and the targets was performed according to the procedure described in Section 3.3.3. The initial weighting factor $\lambda^{(0)}$ and the balancing ratio σ are constant to be set appropriately according to the simulation results. Simulation was performed for $\lambda^{(0)}$ values of 10, 1, 0.2 and for σ values of 5, 1, 0.2 while maintaining $\sigma = 1$ and $\lambda^{(0)} = 0.2$, respectively. The convergence patterns of the reconstruction error ($E^{(l)}_{rms}$ against l) are shown in the Fig. 3.4 and Fig. 3.5. Reconstruction error decreases and reaches a certain minimum after a few iterations. Fig. 3.6 shows the effect of the iterative refinement on the event function shapes. Minor lobe content decreases and becomes almost negligible after a few iterations. The minor lobe smoothing and major lobe reshaping can be observed as desirable effects of the refinement procedure.

In Fig. 3.7, a plot of SFTR and the final event functions are shown for the female English sentence utterance "*we always thought we would die with our boots on.*" The

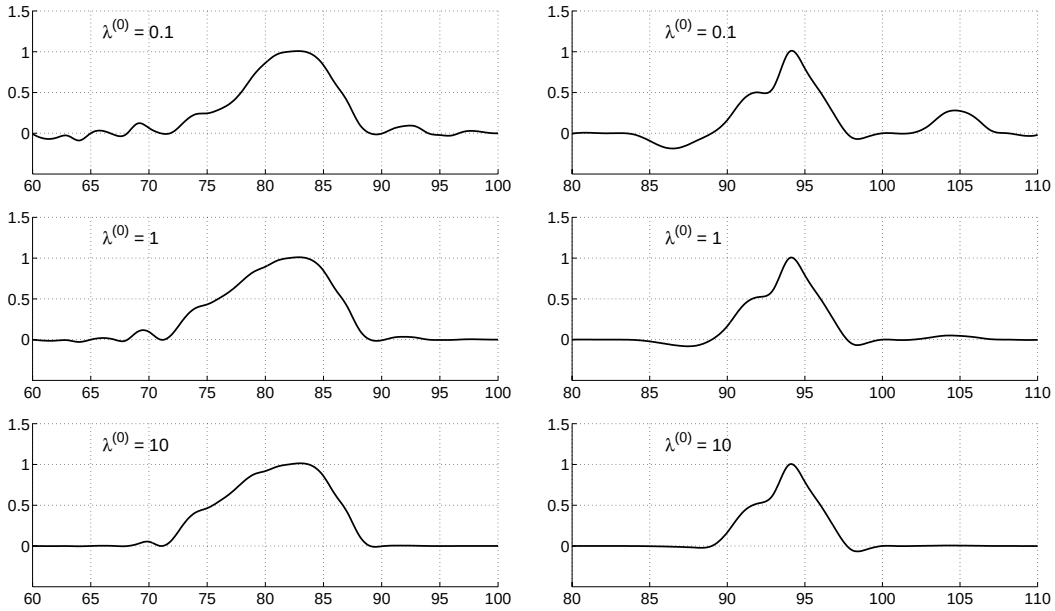


Figure 3.3: Typical shape of the Initial event functions, $\phi_k(n)^{(0)}$, for some k. Note that $MLC^{(0)}$ increases as $\lambda^{(0)}$ decreases.

spectral parameter is LSF, which has the same tendency as LAR but the magnitude of $\lambda^{(0)}$ is different. Here $\lambda^{(0)} = 0.005$, $\sigma = 1$ were selected as appropriate values for the initial weighting factor and balancing ratio, respectively. SFTR window size of $2M = 40$ ms was selected resulting in an average event rate of about 20 events/sec. The speech waveform of the utterance is also shown together with the phonetic transcription for reference. The window size, $2M$, of SFTR analysis is the only parameter that effects the number and locations of the events in the S^{2}BEL-TD algorithm. It controls the event rate, and can be appropriately selected to achieve the optimal performance of S^{2}BEL-TD for different applications. In speech coding point of view, window size, $2M$, can be selected so as to obtain a certain optimal tradeoff between reconstruction accuracy of the spectral parameters (spectral distortion) and the bit rate. In speech decoding, it can be selected to optimize the correlation between phonemes/sub-phonemes and events.

3.3.6 Performance Evaluation

In this section, the performance of S^{2}BEL-TD in terms of interpolation property, computational complexity, and stability of the number and locations of the events were evaluated. To evaluate the interpolation performance of the S^{2}BEL-TD algorithm, log spectral distortion (LSD) is employed, where LSD is a commonly used objective measure in evaluating the performance of LPC quantization [109] and interpolation [110]. LSD measure was also used for evaluating the interpolation performance of several TD algorithms, e.g., [127, 11]. This criterion is a function of the distortion introduced in the spectral density of speech in each particular frame. Log spectral distortion, D_n, for the nth frame is defined (in dB) as follows:

$$D_n = \sqrt{\frac{1}{F_s}\int_0^{F_s} [10log_{10}(P_n(f)) - 10log_{10}(\hat{P}_n(f))]^2 df}$$

where F_s is the sampling frequency, and $P_n(f)$ and $\hat{P}_n(f)$ are the LPC power spectra

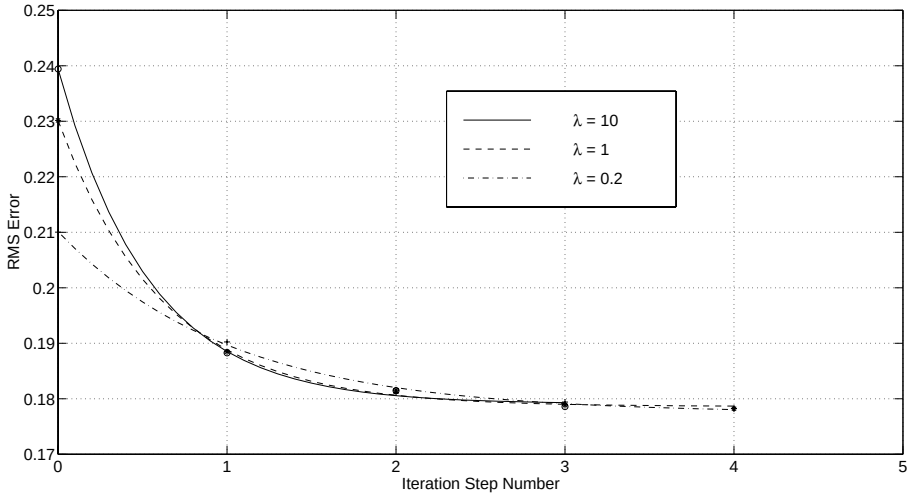


Figure 3.4: Convergence patterns of the reconstruction error, $E_{rms}^{(l)}$, with iteration step l, for different values of $\lambda^{(0)}$. Balancing ratio is $\sigma = 1$. Note that after few iterations $E_{rms}^{(0)}$ reaches a minimum.

Table 3.1: Average log spectral distortion and percentage number of outlier frames for LARs and LSFs. The speech data set consists of 250 sentence utterances spoken by 10 speakers (5 males & 5 females) of the ATR Japanese speech database

Parameter	Avg. LSD (dB)	$\leq$ 2 dB	2-4 dB	$>$ 4 dB
LAR	1.7831	69.0%	26.8%	4.2%
LSF	1.4643	80.6%	18.5%	0.9%

corresponding to the nth frame of the original spectral parameters, $\mathbf{y}(n)$, and the reconstructed spectral parameters, $\hat{\mathbf{y}}(n)$, respectively. The results are provided in terms of log spectral distortion histograms, average log spectral distortion and percentage outliers having log spectral distortion greater than 2 dB. The outliers are divided into the following two types. Type 1: consists of outliers in the range 2-4 dB, and Type 2: consists of outliers having spectral distortion greater than 4 dB.

A set of 250 sentence utterances of the ATR Japanese speech database [2] and another set of 192 sentence utterances of the TIMIT English speech database [44] were selected for spectral distortion evaluation. The Japanese speech data set consists of about 20 minutes of speech from 10 speakers (5 males & 5 females). Meanwhile, the English speech data set contains 24 speakers, 2 males and 1 female from each of 8 dialect regions. Each speaker read a different set of 5 phonetically-compact sentences (the SX sentences) and 3 phonetically-diverse sentences (the SI sentences). Both LAR and LSF parameters were calculated, and S^2BEL-TD analyzed. LSD was calculated on a frame-by-frame basis.

Table 3.1 & Table 3.2 give the summary of the log spectral distortion results obtained for the above sets of utterances with LAR and LSF as the spectral parameter. The distribution of the log spectral distortion in the form of histograms are shown in Fig. 3.8 and Fig. 3.9, each for both cases of LAR and LSF parameters concerning with one speech data set. Results indicate slightly better performance in the case of LSF parameters over LAR parameters.

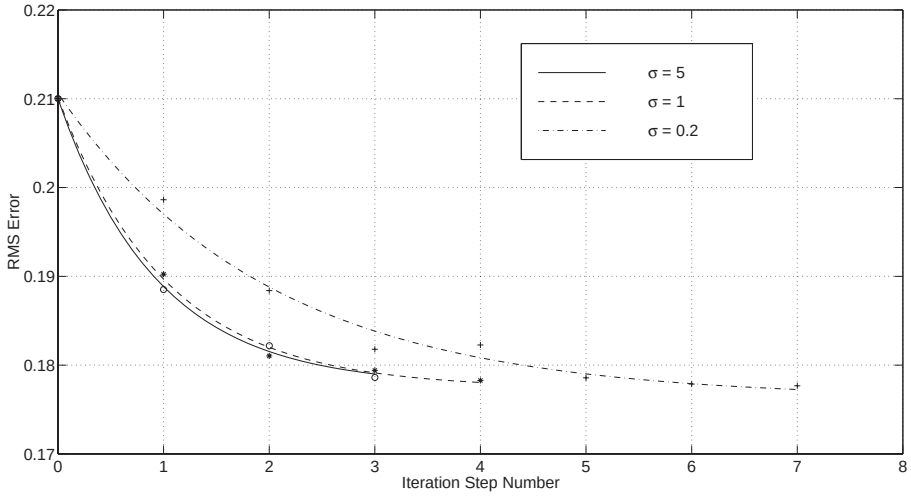


Figure 3.5: Convergence patterns of the reconstruction error, $E_{rms}^{(l)}$, with iteration step l, for different σ. Initial weighting factor is $\lambda^{(0)} = 0.2$. Note that after few iterations $E_{rms}^{(0)}$ reaches a minimum, and σ acts as an accelerating factor for convergence.

Table 3.2: Average log spectral distortion and percentage number of outlier frames for LARs and LSFs. The speech data set consists of 192 sentence utterances spoken by 24 speakers (2 males & 1 female from each of 8 dialect regions) of the TIMIT English speech database

Parameter	Avg. LSD (dB)	$\leq$ 2 dB	2-4 dB	$>$ 4 dB
LAR	1.6863	72.7%	23.7%	3.6%
LSF	1.4778	79.9%	19.0%	1.1%

Since the S^{2}BEL-TD aims at overcoming the two drawbacks of high computational cost, and the high parameter sensitivity of the number and locations of the events imposed on the Atal's method, it is necessary to evaluate the performance of S^{2}BEL-TD on these aspects. With respect to computational complexity the S^{2}BEL-TD shows a significant improvement over the original method by Atal. This can be mainly attributed to the fact that the SVD is the most time consuming part of the Atal's method [141] and the SVD is not required for S^{2}BEL-TD. Moreover, the S^{2}BEL-TD was implemented in a mathematically simpler way than that of Atal's method. The instability problem of the number and locations of the events with respect to TD analysis window size and the number of parameters retained after SVD, has also been overcome in S^{2}BEL-TD. It has been emphasized in Section 3.3.1 that the window size of SFTR analysis is the only parameter that effects the number and locations of the events in the S^{2}BEL-TD method. Since SFTR is a local measure, the TD analysis window size makes no difference in the number and locations of the event functions found. But this is not the case in the original method by Atal, where even a trivial change in window size or number of parameters retained after SVD leads to a dramatic changes in the number and locations of the event functions. Investigation of Atal's method by [91] has revealed this fact.

In addition, the S^{2}BEL-TD was used for analyzing a considerable number of speech utterances spoken by different speakers (males & females) in different speech conditions and worked satisfactorily. All speech utterances were well S^{2}BEL-TD analyzed using the

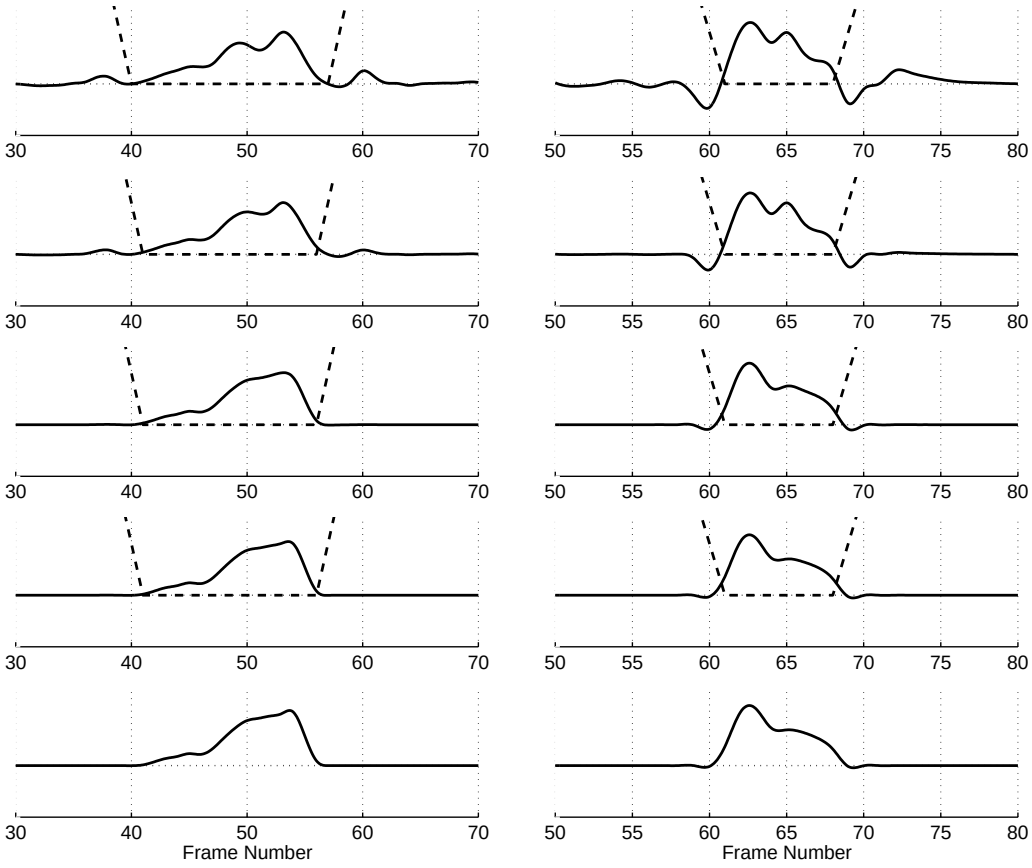


Figure 3.6: Effect of iterative refinement on event function shapes for some k (Top: initial event functions, $\phi_k(n)^{(0)}$'s, Bottom: final event functions, $\phi_k(n)^{(S)}$'s). Weighting functions, $w_k(n)^{(l)}$'s, are also shown for reference. Note the minor lobe smoothing and major lobe reshaping property which finally results in well-shaped and non-negative event functions.

same parameters, i.e. the initial weighting factor $\lambda^{(0)}$ and the balancing ratio σ.

3.3.7 S^{2}BEL-TD of Excitation Parameters

In this section, the application of S^{2}BEL-TD technique to describing speech excitation parameters and some simulation results are presented.

Determination of Excitation Targets

The S^{2}BEL-TD technique is employed to describe the temporal characteristics of the speech excitation parameters, i.e gain, pitch and voicing. The same event functions evaluated for the spectral parameters are used to describe the temporal pattern of the gain, pitch and voicing parameters also. The speech production mechanism is assumed to be a synchronously controlled process with respect to the movement of different articulators, i.e. jaws, tongue, larynx, glottis etc., and therefore the temporal evolutionary patterns of different properties of speech, i.e. spectrum, pitch, gain and voicing, can be described by

a common set of event functions.

Let $b(n)$ be an excitation parameter, i.e. gain, pitch or voicing. Then $b(n)$ is approximated by $\hat{b}(n)$, the reconstructed excitation parameter for the nth frame, as follows in terms of excitation targets, b_k's, and the event functions, $\phi_k(n)$'s.

$$\hat{b}(n) = \sum_{k=1}^{K} b_k \phi_k(n), \quad 1 \leq n \leq N \tag{3.12}$$

In matrix notation, Equation (3.12) can be written as

$$\hat{B} = A_b \mathbf{\Phi}$$

where $\hat{B}$ and A_b are the reconstructed excitation parameter vector and excitation target vector, respectively.

In Equation (3.12), the event functions, $\phi_k(n)$'s, are known and therefore the excitation targets, b_k's, are determined by minimizing the squared error between the original excitation parameters and the reconstructed excitation parameters as follows:

$$E_b = \sum_{n=1}^{N} \left(b(n) - \sum_{k=1}^{K} b_k \phi_k(n) \right)^2$$

By setting the partial derivative of E_b with respect to b_r, to zero:

$$\begin{aligned} \frac{\partial E_b}{\partial b_r} &= \sum_{n=1}^{N} \left(b(n) - \sum_{k=1}^{K} b_k \phi_k(n) \right) (-2\phi_r(n)) \\ &= 0, \end{aligned}$$

$$\sum_{k=1}^{K} b_k \sum_{n=1}^{N} \phi_k(n)\phi_r(n) = \sum_{n=1}^{N} b(n)\phi_r(n), \quad 1 \leq r \leq K \tag{3.13}$$

Equation (3.13) gives a set of K variable simultaneous equations, using which b_k, where $1 \leq k \leq K$, could be evaluated.

In the case of pitch parameters, linear interpolation was used within the unvoiced segments to form a continuous pitch contour. In the case of voicing parameters, a hard limiter with a threshold value of 0.5 was used to determine the reconstructed binary voicing parameters and binary voicing targets, from the non-binary results of Equations (3.12) and (3.13), respectively.

Simulation Results

The gain, pitch and voicing parameters, hereafter indicated by $g(n)$, $p(n)$, and $v(n)$, respectively, were calculated at 10 ms frame intervals with a 40 ms analysis window, for the sentence utterance "*kantan na shiryou wo ookuri shimasu node, shibaraku omachi kudasai* ," of the ATR Japanese speech database [2]. Each parameter contour was S^{2}BEL-TD analyzed according to the procedure described above with the event functions obtained from S^{2}BEL-TD analysis of LSF parameters.

Fig. 3.10 shows the plots of original and reconstructed gain parameters and the plot of frame-wise gain error, $e_g(n)$, where $e_g(n) = \hat{g}(n) - g(n)$. The RMS gain error, $\sqrt{E_g}$, where $E_g = \frac{1}{N} \sum_{n=1}^{N} {e_g}^2(n)$, was found to be about 4 dB.

Fig. 3.11 shows the plots of original and reconstructed pitch frequency parameters and the plot of frame-wise pitch frequency error, $e_p(n)$, where $e_p(n) = \hat{p}(n) - p(n)$. The RMS pitch error, $\sqrt{E_p}$, where $E_p = \frac{1}{N}\sum_{n=1}^{N} {e_p}^2(n)$, was found to be about 2.3 Hz. In the case of binary voicing parameters, the voicing error, $e_v(n)$, where $e_v(n) = \hat{v}(n) - v(n)$, appeared only at, but not all, voiced/unvoiced boundaries as error spikes of mostly 1 frame. The percentage number of frames with voicing errors was found to be about 4%.

Moreover, the performance of S^{2}BEL-TD in terms of excitation parameters has also been evaluated over the set of 250 Japanese sentence utterances and the set of 192 English sentence utterances used in Section 3.3.6. The RMS gain error, RMS pitch error and percentage number of frames with voicing errors were found about 4 dB, 6 Hz and 5%, respectively. It was observed that the RMS gain error and RMS pitch error can be mainly attributed to some discrete time points, where the corresponding frame-wise gain error and pitch error obtained very high values. Meanwhile, no voicing errors were observed during continuous voiced and unvoiced segments, except for the points of voicing transitions.

The significant match between the original and reconstructed excitation parameters results in the fact that a common set of event functions can be used to describe the temporal patterns of both spectral and excitation parameters.

3.4 Restricted Temporal Decomposition of LSF Parameters

To estimate proper events for speech, some restrictions on event functions $\phi_k(n)$s should be enforced as follows: (i) $\phi_k(n)$ values are restricted to the range $[0; 1]$; (ii) $\phi_k(n)$ is restricted to having maximum value at its corresponding event location n_k; and (iii) events are ordered with respect to their central positions, i.e., event locations. For Restricted Temporal Decomposition (RTD), the authors proposed the use of the restricted second order TD model, however, without any explicit explanation. By the restricted second order TD model we mean that only two event functions can overlap at any instant of time and they sum up to one. With these restrictions, Equation (3.1) can be simplified as follows:

$$\begin{aligned} \hat{\mathbf{y}}(n) &= \mathbf{a}_k\phi_k(n) + \mathbf{a}_{k+1}\left(1 - \phi_k(n)\right) \\ &= \phi_k(n)(\mathbf{a}_k - \mathbf{a}_{k+1}) + \mathbf{a}_{k+1}, \quad n_k \le n < n_{k+1} \end{aligned} \tag{3.14}$$

where n_k and n_{k+1} are the locations of event k and event $k+1$, respectively.
Therefore, the total reconstruction error for the whole speech segment can be written as

$$\begin{aligned} E &= \| Y - A\Phi \|^2 = \sum_{n=1}^{N} \| \mathbf{y}(n) - \hat{\mathbf{y}}(n) \|^2 \\ &= \sum_{k=1}^{K-1} \sum_{n=n_k}^{n_{k+1}-1} \| \left(\mathbf{y}(n) - \mathbf{a}_{k+1}\right) - \left(\mathbf{a}_k - \mathbf{a}_{k+1}\right)\phi_k(n) \|^2 \end{aligned} \tag{3.15}$$

To minimize the error E, $\phi_k(n)$ should be estimated using Equation (3.16), which is obtained from setting the partial derivatives of Equation (3.15) with respect to $\phi_k(n)$

equal to zero.

$$\hat{\phi}_k(n) = \frac{\langle (\mathbf{y}(n) - \mathbf{a}_{k+1}), (\mathbf{a}_k - \mathbf{a}_{k+1}) \rangle}{\| \mathbf{a}_k - \mathbf{a}_{k+1} \|^2} \tag{3.16}$$

where, $< ., . >$ and $\| . \|$ denote the inner product of two vectors and the Euclid norm of a vector, respectively.

Finally, considering the above restriction on the event functions, the RTD determines $\phi_k(n)$ as

$$\phi_k(n) = \begin{cases} 1 - \phi_{k-1}(n), & \text{if } n_{k-1} < n < n_k \\ 1, & \text{if } n = n_k \\ \min(1,\ \max(0, \hat{\phi}_k(n))), & \text{if } n_k < n < n_{k+1} \\ 0, & \text{otherwise} \end{cases} \tag{3.17}$$

Turning to the event targets, the RTD method estimates them corresponding to the determined event functions in the least mean squared error sense by using the following formula [8]. Note that $(\mathbf{\Phi}\mathbf{\Phi}^T)$ here is tridiagonal and can be easily inverted.

$$\mathbf{A} = \mathbf{Y}\mathbf{\Phi}(\mathbf{\Phi}\mathbf{\Phi}^T)^{-1} \tag{3.18}$$

The estimated event targets may violate the ordering property of LSFs since this property is not taken into account in the error minimization criterion. Assuming that the minimum differential LSF (dLSF) between two consecutive parameters is ε, RTD re-estimates the event targets from the lowest to the highest order, replaces $a_{i-1,k}$ and $a_{i,k}$ by $\hat{a}_{i-1,k}$ and $\hat{a}_{i,k} = \hat{a}_{i-1,k} + \varepsilon$, respectively, whenever $a_{i-1,k} + \varepsilon > a_{i,k}$. Taking into account the increment of error E caused by this change:

$$\mathbf{\Delta} = \sum_n (a_{i-1,k} - \hat{a}_{i-1,k})^2 \phi_k(n)^2 + \sum_n (a_{i,k} - \hat{a}_{i,k})^2 \phi_k(n)^2 \tag{3.19}$$

$\hat{a}_{i-1,k}$ should be set as follows to minimize $\mathbf{\Delta}$.

$$\hat{a}_{i-1,k} = \frac{a_{i-1,k} + a_{i,k} - \varepsilon}{2} \tag{3.20}$$

In the result, when the event locations n_k, $k = 1, ..., K$ are known, and the corresponding event targets are initialized with the samples of the vector trajectory $\mathbf{y}(n_k)$, we can calculate proper event functions and event targets iteratively by using Equations (3.17), (3.18), and (3.20). RTD determines the initial event locations as the local minimum points of a measure of spectral dynamics called Spectral Transition Measure (STM), which is similar to the spectral feature transition rate (SFTR) in the S^{2}BEL-TD method [103].

$$STM_{LSF}(n) = \| \sum_{t=-M}^{M} t.y(n+t) \|^2 \tag{3.21}$$

where $M = 2$. After that, RTD inserts a new event where the initial interpolation error $\mathbf{e}(n) = \| \mathbf{y}(n) - \hat{\mathbf{y}}(n) \|^2$ has a local maximum and is larger than the certain threshold θ. For online analysis, RTD segments the input vector trajectory and performs TD of each segment sequentially. A segment can be bounded by two event found with $STM_{LSF}(n)$. Since an event function is overlapped by its adjacent ones, the last event of each segment should be re-estimated with the following segment. The whole RTD is summarized as follows [70, 71].

1. Initialize $n_1 \leftarrow 1$, $a_1 \leftarrow y(1)$, and $K = 2$.

2. Find n_K, the next local minimum point of $STM_{LSF}(n)$, which will be the end of the current segment; set $a_K \leftarrow y(n_K)$.

3. Estimate the initial event functions ϕ_k corresponding to the current set of event targets using Equation (3.17); note that no iteration is needed for this step.

4. If $\max_n(e(n)) > \theta$, insert an event as $K \leftarrow K + 1$, $n_K \leftarrow \text{argmax}_n(e(n))$, $a_K \leftarrow y(n_K)$, and reorder the events by their central positions, then go back to step 3.

5. Re-estimate the event targets a_k using Equations (3.18) and (3.20), but do not update a_1 if it is from the previous segment.

6. Re-estimate the event functions using Equation (3.17); if the results have converged or have been re-estimated a certain number of times, go to step 7; if not, go back to step 5.

7. Store the events for the current segment; to analyze the next segment, set $n_1 \leftarrow n_{K-1}$, $n_2 \leftarrow n_K$, $a_1 \leftarrow a_{K-1}$, $a_2 \leftarrow a_K$, and $K \leftarrow 3$, then go back to step 2.

Kim and Oh [71] evaluated the performance of the RTD method over a speech data set collected from the TIMIT speech corpus [44]. They chose 1890 phonetically-diverse sentences (SI set) from the TIMIT database and used 1386 sentences for training and 504 sentences for testing. A 10th order LPC analysis was performed using the autocorrelation method with a 30 ms Hamming window which was shifted by 20 ms. Finally, the LPC coefficients were converted to LSF parameters and then TD analyzed. They reported that they were able to interpolate the LSF parameter vector trajectory using RTD with approximately 18 events/sec while the loss of prediction gain was only 0.15 dB. They then quantized the events obtained from RTD analysis using scalar quantization of event targets and vector quantization of event function shapes. Here, 33 bits were used for quantizing an event target while 6 bits were used for quantizing the shape of an event function. It is of interest to note that each event function was length-normalized before quantization by taking 10 equidistant samples and an event function can be represented by its length and shape. Since the maximum event function length was 11 frames, 4 bits were used for encoding the length of an event function. The average log spectral distortion (LSD) caused by both interpolation and quantization was found to be about 1.74 dB at the bit rate of 753 bps.

3.5 Summary and Discussion

The original TD method by Atal [8] involves the description of speech parameter vector trajectory in terms of a sequence of overlapping event functions and an associated sequence of event targets. Atal's method provided a means of efficient coding of log-area parameters with good spectral parameter reconstruction accuracy. The computational procedure of the method includes two stages. First, the event locations are detected using the singular value decomposition (SVD) and, second, the event functions and targets are refined using an iterative procedure to minimize the distance (or error) between the reconstructed and original spectral parameter vectors. Although Atal's method of temporal decomposition

demonstrated clear potential benefits in terms of data compression, its high computational complexity, which can be mainly attributed to the computationally costly SVD routine, and the high parameter sensitivity of the number and locations of the events should be noted as the major drawbacks, specially in practical application point of view.

The S^2BEL-TD [103] approach to temporal decomposition of speech aims at overcoming these two drawbacks imposed on the original TD method. The spectral stability criterion used in event localizing, and the use of adaptive weighting functions in determining the event functions, can be highlighted as the main features of the S^2BEL algorithm for TD. The former makes the event localization more parameter independent eventually overcoming the instability problem of the Atal's method. The latter gives a great degree of freedom to the event functions to evolve through iterations. Also, the S^2BEL algorithm which makes no use of SVD algorithm and the redundant calculation of event functions, can be considered as a significant improvement in terms of computational cost compared to the original method by Atal. On continuous speech S^2BEL-TD can be performed on a segmental basis. The representation of speech excitation parameters also in terms of excitation targets and event functions makes S^2BEL-TD a complete higher-level parametric model of speech. However, the stability of the LPC synthesis filter after spectral transformation of LSF parameters performed by S^2BEL-TD has not been taken into account. Therefore, S^2BEL-TD cannot always be applied to decomposing LSF parameters. In addition, still there is no evidence that S^2BEL-TD can be applied to analyzing speech in any condition. That is, there is no evidence for the convergence property of the iterative refinement procedure imposed on S^2BEL-TD.

The RTD method [71] is presented for LSF parameters. LSF parameters have been known to have better interpolation and quantization properties over the other LPC related spectral parameters. Therefore, most of low bit rate speech coders adopt LSF parameters to represent the spectral information of speech. LSF parameters had rarely been considered as input for TD in the literature due to the stability problems in the corresponding LPC synthesis filter. This is because there is no guarantee that the reconstructed LSF parameters are valid after spectral transformation performed by TD. The RTD method considers the LSF ordering property to make LSF parameters possible for TD. This method employs a restricted second order TD model, where only two adjacent event functions can overlap and all event functions at any moment of time sum up to one. The RTD method can achieve results comparable to S^2BEL-TD in terms of reconstruction accuracy with lower computational complexity and higher stability. However, RTD has not completely guaranteed the LSF ordering property for the event targets. The reason is that RTD re-estimates the event targets from the lowest to the highest order, replacing each pair of components if their values are not in ascending order, which may result in the fact that some components of event targets may be re-estimated twice and, after the second re-estimation, they violate the ordering property with the previous ones. Thus, the stability of the corresponding LPC synthesis filter has not completely been ensured. Also, event functions obtained from RTD analysis may have more than one peak, which is undesirable from speech coding point of view. Specially, still there are several event functions having more than one lobe, which is not acceptable in the conventional TD method.

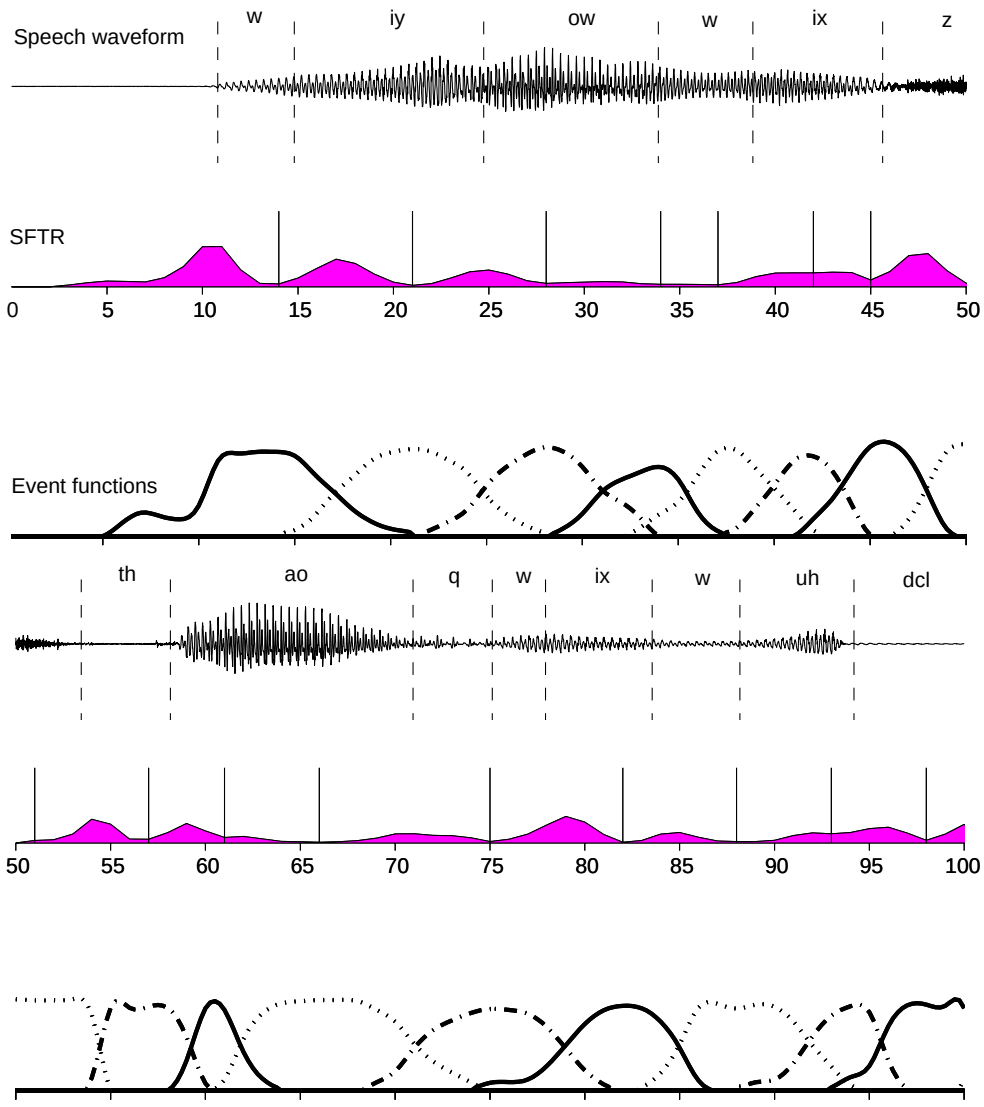


Figure 3.7: Plot of SFTR and the final event functions for the utterance "*we always thought we would die with our boots on.*" S^{2}BEL-TD analysis has been performed on the utterance on a segmental basis. The speech waveform is also shown together with the phonetic transcription for reference. Broken lines in the speech plot show the phoneme boundaries, while the solid lines in the SFTR plot show the spectrally stable frame locations, i.e. local minima of SFTR.

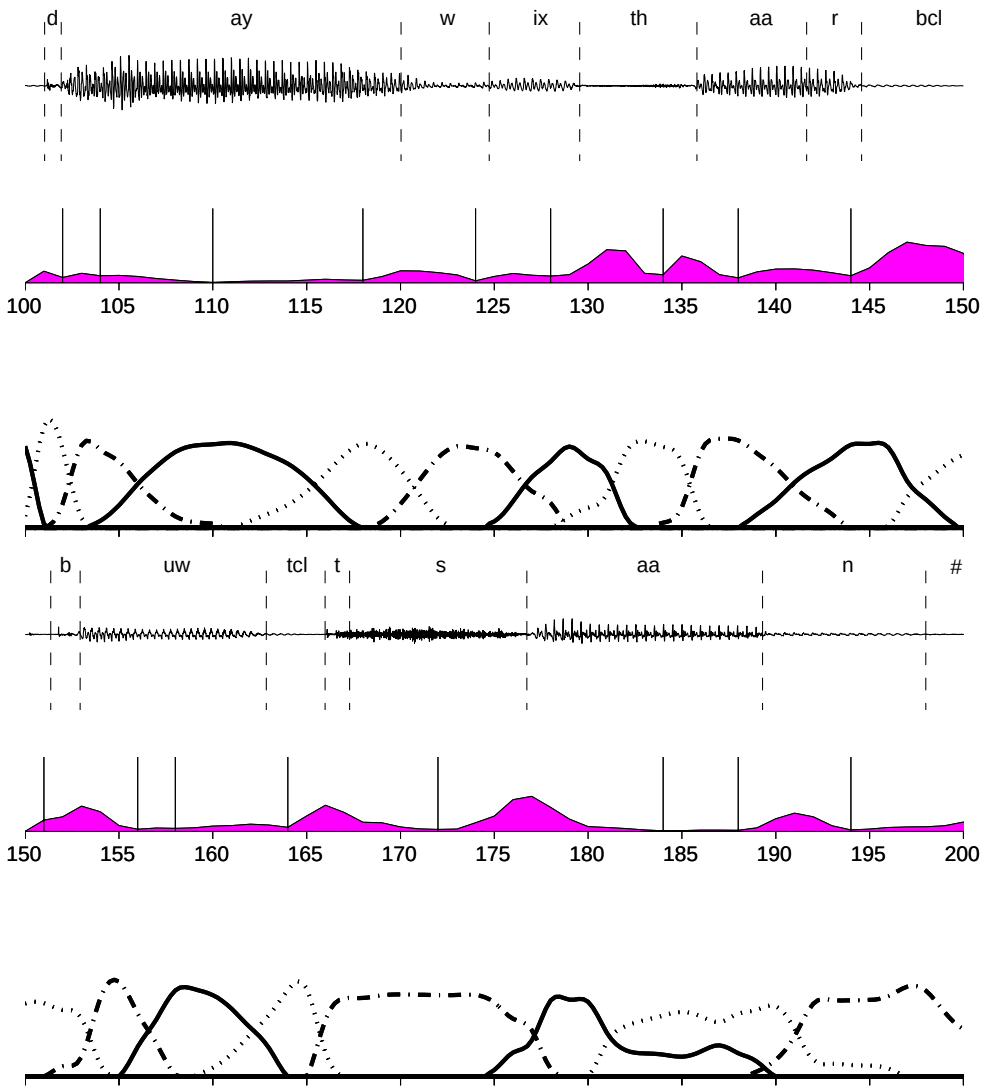


Figure 3.7: *Continued.*

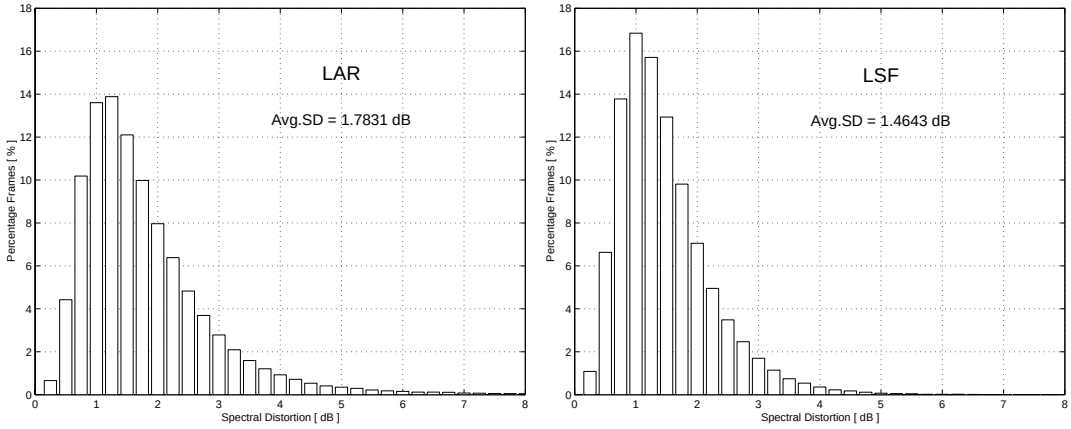


Figure 3.8: Distribution of the log spectral distortion (LSD) between original and reconstructed spectral parameters in the form of histograms. Left: LSD histogram for the LAR parameters, Right: LSD histogram for the LSF parameters. Speech data set consists of 250 sentence utterances spoken by 10 speakers (5 males & 5 females) of the ATR Japanese speech database. LSFs show slightly better reconstruction accuracy than LARs.

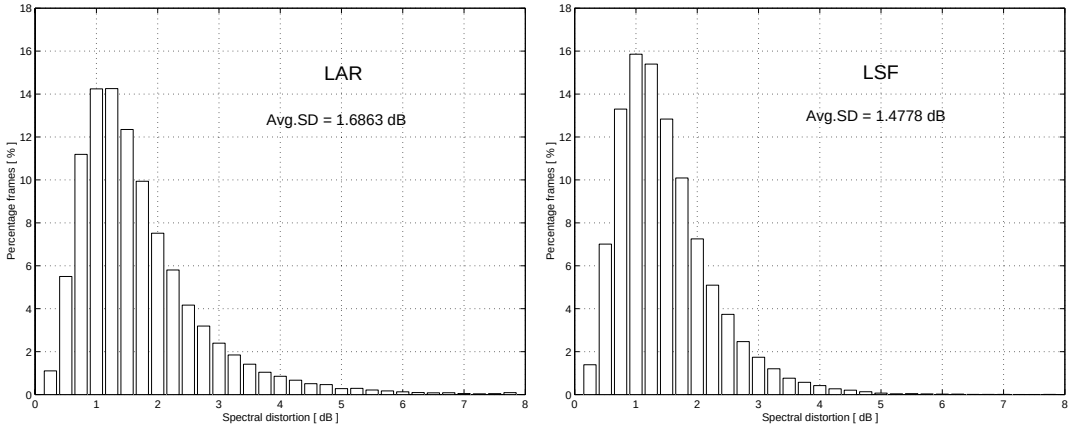


Figure 3.9: Distribution of the log spectral distortion (LSD) between original and reconstructed spectral parameters in the form of histograms. Left: LSD histogram for the LAR parameters, Right: LSD histogram for the LSF parameters. Speech data set consists of 192 sentence utterances spoken by 24 speakers (2 males & 1 female from each of 8 dialect regions) of the TIMIT English speech database. LSFs also show slightly better reconstruction accuracy than LARs.

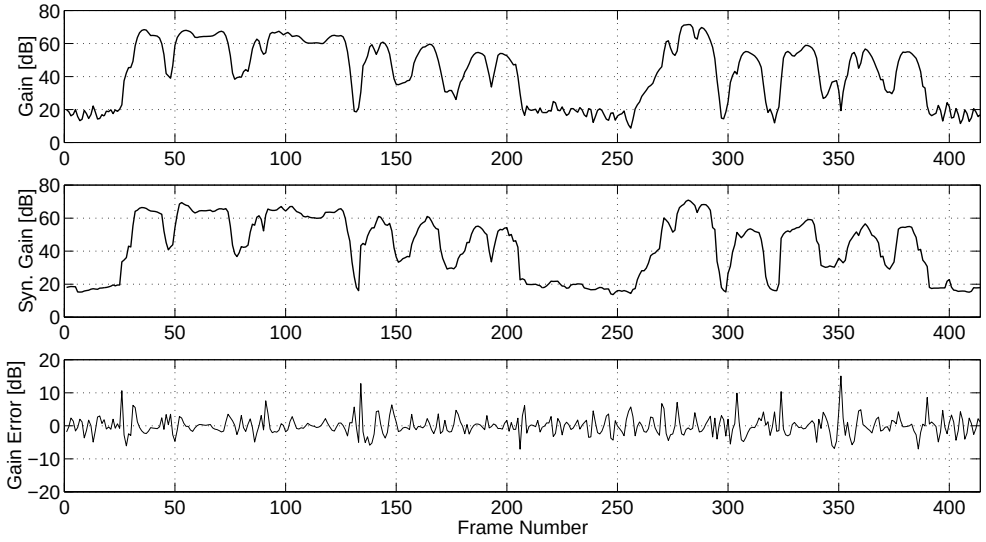


Figure 3.10: Original gain parameters, $g(n)$, reconstructed gain parameters, $\hat{g}(n)$, and frame-wise gain error, $e_g(n) = \hat{g}(n) - g(n)$, for the utterance "*kantan na shiryou wo ookuri shimasu node, shibaraku omachi kudasai*," of the ATR Japanese speech database. The root-mean-squared (RMS) gain error is 4.051 dB.

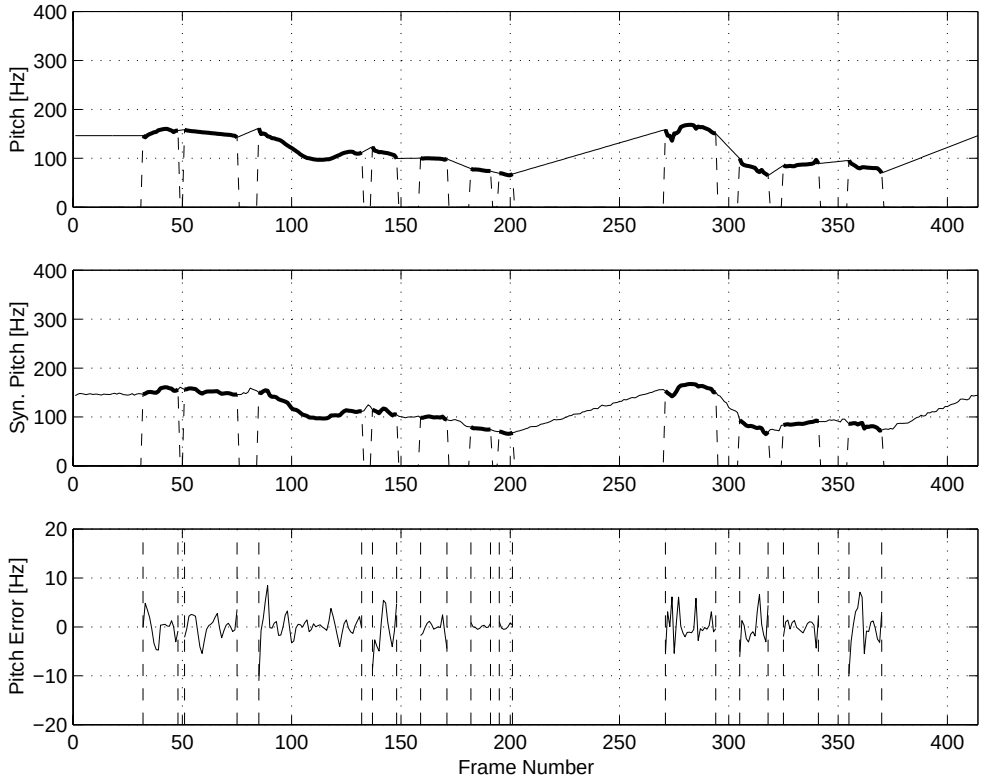


Figure 3.11: Original pitch parameters, $p(n)$, reconstructed pitch parameters, $\hat{p}(n)$, and frame-wise pitch error, $e_p(n) = \hat{p}(n) - p(n)$, for the utterance "*kantan na shiryou wo ookuri shimasu node, shibaraku omachi kudasai*," of the ATR Japanese speech database. Pitch error is shown only for the voiced segments of the utterance. The RMS pitch error is 2.2984 Hz.

Chapter 4

Improving the Restricted Temporal Decomposition Method for LSF Parameters

4.1 Introduction

As already presented, temporal decomposition (TD) [8], which is an analysis procedure based on a linear model of the effects of co-articulation, yields a linear approximation of a time sequence of spectral parameters in terms of a series of time-overlapping event functions and an associated series of event targets as given in Equation (4.1).

$$\hat{\mathbf{y}}(n) = \sum_{k=1}^{K} \mathbf{a}_k \phi_k(n), \quad 1 \leq n \leq N \tag{4.1}$$

where $\mathbf{a}_k$ and $\phi_k(n)$ are the k^{th} event target and k^{th} event function, respectively. $\hat{\mathbf{y}}(n)$ is the approximation of $\mathbf{y}$(n), the n^{th} spectral parameter vector, produced by the TD model. In matrix notation, Equation (4.1) can be written as

$$\hat{\mathbf{Y}} = \mathbf{A}\mathbf{\Phi} \qquad \hat{\mathbf{Y}} \in \mathbf{R}^{P \times N}, \mathbf{A} \in \mathbf{R}^{P \times K}, \mathbf{\Phi} \in \mathbf{R}^{K \times N}$$

where P, N, and K are the order of the spectral parameters, the number of frames in the speech segment, and the number of event functions, respectively.

Assume that the co-articulation in speech production described by the TD model in terms of overlapping event functions is limited to adjacent events, the second order TD model [106, 127, 11], where only two adjacent event functions can overlap as depicted in Fig. 4.1, is given in Equation (4.2).

$$\hat{\mathbf{y}}(n) = \mathbf{a}_k \phi_k(n) + \mathbf{a}_{k+1} \phi_{k+1}(n), \; n_k \leq n < n_{k+1} \tag{4.2}$$

where n_k and n_{k+1} are the locations of event k and event $k+1$, respectively.

The restricted second order TD model was utilized in [34, 71] with an additional restriction to the event functions in the second order TD model is that all event functions at any time sum up to one. The argument for imposing this constraint on the event functions has not been explicitly stated in [71]. But, it has been shown in [34] that this constraint is needed to describe TD as a breakpoint analysis procedure in a multidimensional vector

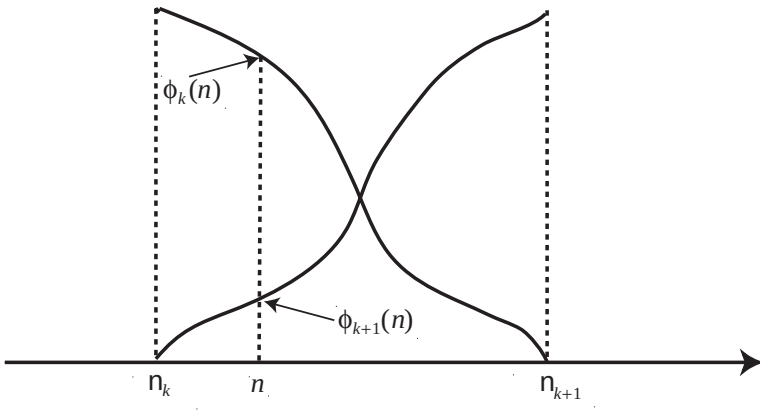


Figure 4.1: Example of two adjacent event functions in the second order TD model.

space, where breakpoints are connected by straight line segments. Equation (4.2) can be rewritten as

$$\hat{\mathbf{y}}(n) = \mathbf{a}_k \phi_k(n) + \mathbf{a}_{k+1} \left(1 - \phi_k(n)\right), \; n_k \leq n < n_{k+1} \tag{4.3}$$

The spectral parameters used in the original TD method by Atal [8] were the log-area parameters. Some other spectral parameter sets such as log area ratios, cepstrum, and so forth have also been considered as input for TD [49]. Due to the stability problems in the linear predictive coding (LPC) model, not all types of parametric representations of speech can be used. This is because there is no guarantee that the selected spectral parameters are valid after spectral transformation performed by TD. Thus, the line spectral frequency (LSF) parameters [62] have not been used for the conventional TD method although they have several properties that make them more suitable for interpolation [110] and quantization [109].

An important property of LSFs $\{\omega_i\}$ is that they are ordered in $(0, \pi)$ as follows:

$$0 < \omega_1 < \omega_2 < \cdots < \omega_P < \pi \tag{4.4}$$

Also, (4.4) means that the difference of two consecutive LSFs (dLSF) $\{d_i = \omega_i - \omega_{i-1}\}$ with $d_1 = \omega_1$ and $d_{P+1} = \pi - \omega_P$ are always greater than zero. This ordering property is a necessary and sufficient condition for the stability of the corresponding LPC synthesis filter. It implies that TD can be applied to analyzing the LSF parameters if the ordering property of LSFs is guaranteed for the event targets.

Kim and Oh [71] have introduced a method of temporal decomposition for the LSF parameters, called "Restricted Temporal Decomposition" (RTD), based on the restricted second order TD model. The RTD method enforces a minimum dLSF constraint on the event targets in order to preserve their LSF ordering property. Originally, RTD was proposed in narrowband speech coding for significantly reducing the bit rate for spectral parameters [71]. Subsequent research [122] investigated its application to wideband speech coding and found that RTD is a promising approach to low rate wideband speech coding also. However, both have not reported any drawback, from which RTD is being suffered.

In this chapter, we claim that the RTD method, however, has not completely guaranteed the LSF ordering property for the event targets; instead, we propose an improved algorithm, namely modified RTD (MRTD), to solve this problem. Additionally, we impose a new property, the well-shapedness property, on the event functions to model the

temporal structure of speech more effectively and reduce the quantization error when vector quantized. Also, we have conducted an experiment to evaluate the performance of MRTD in modeling and quantizing speech excitation parameters. Results show that excitation information of speech can also be well described and quantized using the MRTD technique.

The chapter is organized as follows: In the next section, we review the description of TD as a breakpoint analysis procedure. Section 4.3 is devoted to describing the MRTD algorithm. Section 4.4 presents a method of vector quantization of LSF parameters based on MRTD. Meanwhile, the use of MRTD in modeling and quantizing speech excitation parameters is provided in Section 4.5. Finally, some conclusions are drawn in Section 4.6.

4.2 TD as a Breakpoint Analysis Procedure

Temporal decomposition makes little use of any specific phonetic knowledge. It yields an approximation of a sequence of spectral parameters by a linear combination of event targets. It only uses information on the time scale at which events take place and makes a presumption on the general shape of event functions. In this section, we briefly summarize the geometric interpretation of TD which was initiated by Niranjan and Fallside [106], and then further developed by Dix and Bloothooft [34].

Since TD's underlying distance metric is Euclidean, a natural requirement is to have this approximation be invariant with respect to a translation or rotation of the spectral parameters [34]. Invariance with respect to these operations can be given a geometric interpretation if the consecutive sequence of spectral parameter vectors $\mathbf{y}(n)$ is seen as a path in a multidimensional parameter space. The relative positions of the actual path $\mathbf{y}(n)$ and the approximated path $\hat{\mathbf{y}}(n)$ are the same if the parameter space is rotated or if a different origin is used (see Fig. 4.2). If the event targets $\mathbf{a}_k$ transform similarly to the spectral parameters while the event functions remain the same as given in Table 4.1, the requirement of rotation and translation invariance is met. These transformation properties are in accordance with an interpretation of event functions as a description of the temporal evolution along a path and the event targets as a description of the locations in parameter space; rotation and translation only change the location in parameter space and have no effect on the temporal evolution.

Table 4.1: Desired transformation properties of TD if invariance with respect to translations $\mathbf{T}$ and rotations $\mathbf{R}$ is required.

Spectral parameters	Approximation	Event functions	Event targets
$\mathbf{y}(n)$	$\hat{\mathbf{y}}(n)$	$\phi_k(n)$	$\mathbf{a}_k$
$\mathbf{T}\mathbf{y}(n)$	$\mathbf{T}\hat{\mathbf{y}}(n)$	$\phi_k(n)$	$\mathbf{T}\mathbf{a}_k$
$\mathbf{R}\mathbf{y}(n)$	$\mathbf{R}\hat{\mathbf{y}}(n)$	$\phi_k(n)$	$\mathbf{R}\mathbf{a}_k$

The transformation properties can be used to derive a constraint on the possible event functions. It has been shown in [32] that TD is rotation and scale invariant, but it is not translation invariant, basically because it uses the singular value decomposition (SVD), which is not invariant for this transformation. The demand of translation invariance

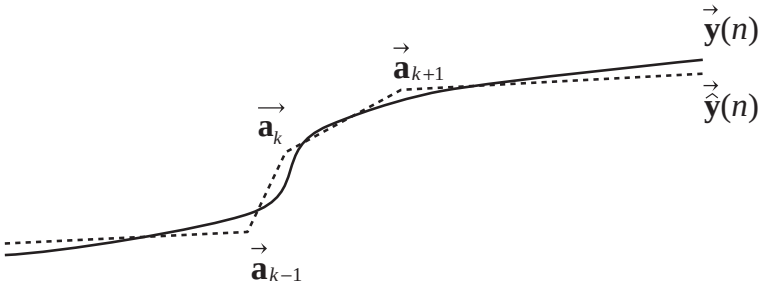


Figure 4.2: The path in parameter space described by the sequence of spectral parameters $\mathbf{y}(n)$ is approximated by means of straight line segments between breakpoints. Note that the breakpoints do not lie on the path describing the sequence of spectral parameters since the event targets are different from the original spectral parameter vectors at the event locations, i.e., $\mathbf{a}_k \neq \mathbf{y}(n_k)$ for every k, due to the refinement of event targets.

implies that if an arbitrary vector $\mathbf{b}$ is added to the spectral parameters $\mathbf{y}(n)$, the following must hold:

$$\hat{\mathbf{y}}(n) + \mathbf{b} = \sum_{k=1}^{K} (\mathbf{a}_k + \mathbf{b})\phi_k(n) = \sum_{k=1}^{K} \mathbf{a}_k \phi_k(n) + \mathbf{b} \sum_{k=1}^{K} \phi_k(n). \tag{4.5}$$

By substituting the definition of $\hat{\mathbf{y}}(n)$ from Equation (4.1) to Equation (4.5), we have

$$\sum_{k=1}^{K} \phi_k(n) = 1. \tag{4.6}$$

Equation (4.6) sets a constraint on the event functions yielded by TD. It also resolves an undetermined degree of freedom in the linear approximation in terms of event functions and targets. This degree of freedom is associated with Equation (4.7), which holds for any set of non-zero constants $\mathbf{c}_k$:

$$\sum_{k=1}^{K} \mathbf{a}_k \phi_k(n) = \sum_{k=1}^{K} (\mathbf{c_k} \mathbf{a}_k)(\phi_k(n)/\mathbf{c}_k). \tag{4.7}$$

Usually, this freedom of choice is resolved by normalizing each event function to have a maximum value of 1.0. The rationale for this practice is that one expects the speech signals to be maximally steady at the instant of time at which a event function reach its maximum value and expects the corresponding event target to be close to the spectral parameter vector at this particular moment. In case of strongly overlapping event function, this choice is somewhat arbitrary, however. One can use Equation (4.6) to determine the scaling in this case.

The relation between event functions expressed by Equation (4.6) can be given a geometrical interpretation if we make the additional assumption that at any moment in time, only two event function can overlap. This assumption yields the most restricted form of overlap, but in practice, it is not a real limitation because one can always use an extra event function in order to get better approximation. Making this assumption is mathematically somewhat sound too since given a number of event targets $< \mathbf{a}_1, \cdots, \mathbf{a}_K >$ with each couple $< \mathbf{a}_k, \mathbf{a}_{k+1} >$ being independent (corresponding to different events), the

projection $P(\mathbf{y}_n) = \hat{\mathbf{y}}_n = \mathbf{a}_1\phi_1(n) + \cdots + \mathbf{a}_K\phi_K(n)$ gives the smallest reconstruction error, and expansion in terms of the event targets $\mathbf{a}_k$ may not be unique, whereas for $K = 2$, the expansion is one to one. If we view the spectral parameters $\mathbf{y}(n)$ as a path in a multidimensional parameter space, then Equation (4.6) can be interpreted geometrically by stating that the approximated path $\hat{\mathbf{y}}_n$ is built up from straight line segments. Using the assumption that only two adjacent event functions can overlap, the movement from event k towards the next $k+1$ is approximated by a weighted sum of two adjacent event targets as already given in Equation (4.2), i.e., the formulation of the second order TD model. In addition, using the fact that $\phi_k(n)$ and $\phi_{k+1}(n)$ must sum up to one according to Equation (4.6), we can have a parametrization of a straight line through the points $\mathbf{a}_k$ and $\mathbf{a}_{k+1}$ as given earlier in Equation (4.3), i.e., the formulation of the restricted second order TD model.

If we furthermore require $\phi_k(n)$ to be in the interval $[0, 1]$, we can see the event targets $\mathbf{a}_k$ as breakpoints marking the start of a new line segment and the end of a previous line segment. The term "breakpoint" was coined by Niranjan and Fallside [106], who showed that if only adjacent event functions may overlap, then TD models the sequence of parameter vectors as a path in a sequence of planes (since linear combination of two event targets generate a plane), "breaking" the path from one plane to the next at the positions of the event targets.

In summary, on the basis of the discussion above, we may conclude that, for reasonable assumptions, TD is equivalent to a breakpoint analysis procedure that yields an approximation of a sequence of spectral parameters by means of straight line segments.

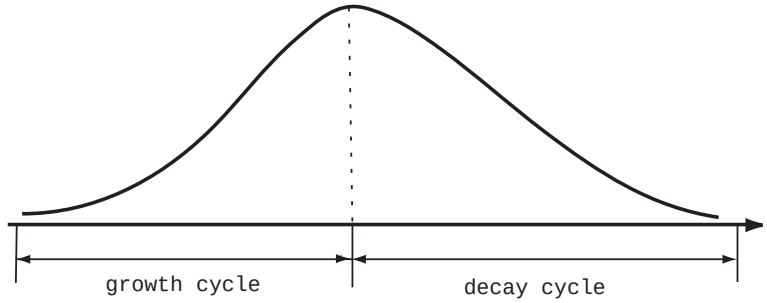
4.3 Modified RTD of LSF Parameters

4.3.1 Additional Constraints on Event Functions

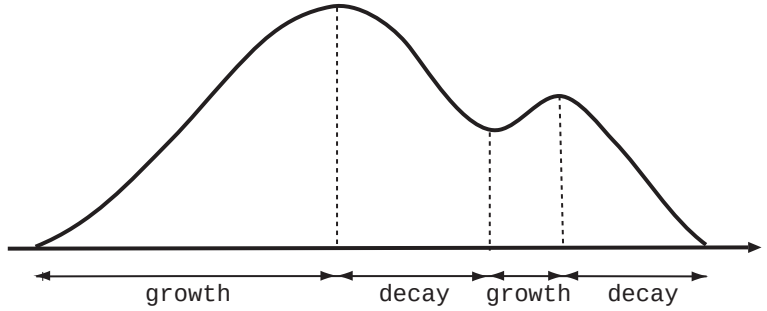
Based on the geometric interpretation of TD described in [34], we impose a new property, namely, the well-shapedness property on the event functions. Here, by a well-shaped event function we mean an event function having only one peak, as depicted in Fig. 4.3(a). Those event functions having more than one peak are called ill-shaped event functions (see, e.g., Fig. 4.3(b)). Well-shaped event functions are desirable from speech coding point of view. Further, the well-shapedness property helps to describe the temporal structure of speech by means of straight line segments between breakpoints more effectively.

As described in Section 4.2, TD yields an approximation of a sequence of spectral parameters by a linear combination of event targets. Since TD's underlying distance metric is Euclidean, a natural requirement is to have this approximation be invariant with respect to a translation or rotation of the spectral parameters. Dix and Bloothooft [34] considered the geometric interpretation of TD results and found that TD is rotation and scale invariant, but it is not translation invariant.

In order to overcome this shortcoming and describe TD as a breakpoint analysis procedure in a multidimensional vector space, Dix and Bloothooft enforced two constraints, which are identical to those in the RTD method [71], on the event functions: (i) at any moment of time only two event functions, which are adjacent in time, are non-zero; and (ii) all event functions at any time sum up to one. In other words, the restricted second order TD model was utilized in both [34] and [71]. Geometrically speaking, the two event targets $\mathbf{a}_k$ and $\mathbf{a}_{k+1}$ define a plane in P-dimensional vector space. The determination of



(a)



(b)

Figure 4.3: Examples of well-shaped (a) and ill-shaped event functions (b).

event functions $\phi_k(n)$ and $\phi_{k+1}(n)$ in the interval $[n_k, n_{k+1}]$ is now depicted in Fig. 4.4 as the projection of vector $y(n)$ onto this plane and is also equivalent to that in [71]. Clearly the following holds: $\phi_k(n_k) = 1$, $\phi_k(n_{k+1}) = 0$, and $0 \leq \phi_k(n) \leq 1$ for $n_k \leq n \leq n_{k+1}$.

The TD model is based on the hypothesis of articulatory movements towards and away from targets. An appealing result of the above properties of event functions is that one can interpret the values $\phi_k(n)$ as a kind of activation values of the corresponding event. During the transition from one event towards the next the activation value of the left event decreases from one to zero, whilst the right event increases its activation value from zero to the value of one. Note that to model the temporal structure of speech more effectively no backwards transitions are allowed. Therefore, each event function should have a growth cycle; during which the event function grows from zero to one and a decay cycle; during which the event function decays from one to zero. In other words, each event function should have the well-shapedness property. In contrast, an ill-shaped event function can be viewed as an event function which has several growth and decay cycles.

However, the determination of event functions in [34, 71, 122] has not guaranteed the well-shapedness property for them since their changes during the transition from one event towards the next may not be monotonic, which results in ill-shaped event functions. In particular, one may wonder that if an event function has some values of one interlaced by other values, it will cause the next event function to have more than one lobe, which is not acceptable in the conventional TD method. Ill-shaped event functions are undesirable from speech coding point of view also. They increase the quantization error when vector quantized because the uncharacteristic valleys and secondary peaks are not normally

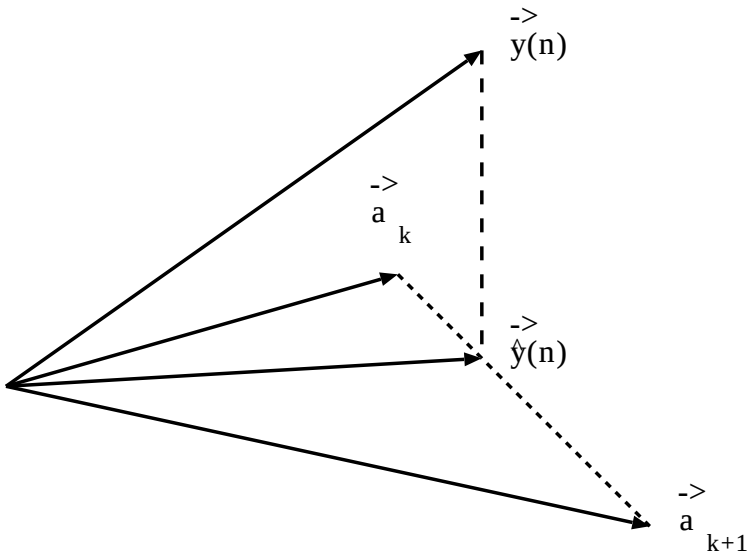


Figure 4.4: Determination of event functions in the transition interval $[n_k, n_{k+1}]$ in the original RTD method. The point of the line segment between $\mathbf{a}_k$ and $\mathbf{a}_{k+1}$ with minimum distance from $\mathbf{y}(n)$ is taken as the best approximation.

captured by the codebook functions.

Taking into account the above considerations, we have determined the event functions corresponding to the point of the line segment between $\hat{\mathbf{y}}(n-1)$ and $\mathbf{a}_{k+1}$ instead of $\mathbf{a}_k$ and $\mathbf{a}_{k+1}$ as considered in [34, 71, 122], with minimum distance from $\mathbf{y}(n)$ (see Fig. 4.5). This determination of event functions can be written in mathematical form as

$$\phi_k(n) = \begin{cases} 1 - \phi_{k-1}(n), & \text{if } n_{k-1} < n < n_k \\ 1, & \text{if } n = n_k \\ \min(\phi_k(n-1), & \\ \ \max(0, \hat{\phi}_k(n))), & \text{if } n_k < n < n_{k+1} \\ 0, & \text{otherwise} \end{cases} \tag{4.8}$$

where

$$\hat{\phi}_k(n) = \frac{\langle (\mathbf{y}(n) - \mathbf{a}_{k+1}), (\mathbf{a}_k - \mathbf{a}_{k+1}) \rangle}{\| \mathbf{a}_k - \mathbf{a}_{k+1} \|^2}$$

Here, $< ., . >$ and $\| . \|$ denote the inner product of two vectors and the norm of a vector, respectively.

4.3.2 Refinement of Event Targets

The event targets are estimated corresponding to the determined event functions in the least mean squared error sense using the following formula [8].

$$\mathbf{A} = \mathbf{Y}\mathbf{\Phi}^T(\mathbf{\Phi}\mathbf{\Phi}^T)^{-1} \tag{4.9}$$

The estimated event targets may violate the ordering property of LSFs since the error minimization criterion does not consider this property. As presented in Section 3.4, given

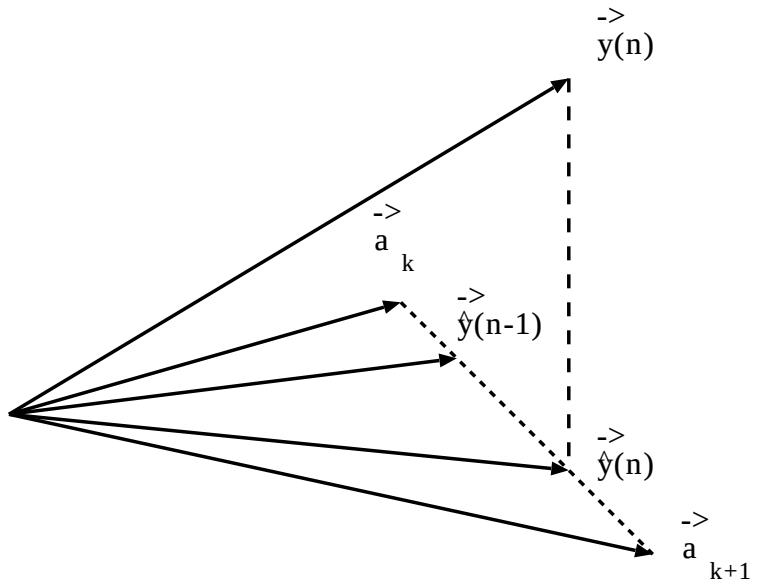


Figure 4.5: Determination of event functions in the transition interval $[n_k, n_{k+1}]$ in the modified method. The point of the line segment between $\hat{\mathbf{y}}(n-1)$ and $\mathbf{a}_{k+1}$ with minimum distance from $\mathbf{y}(n)$ is taken as the best approximation.

the minimum value of dLSFs as ε, the RTD method [71] re-estimates the event target $\mathbf{a}_k$ from the lowest to the highest order, replaces $\mathbf{a}_{i-1,k}$ and $\mathbf{a}_{i,k}$ by $\hat{\mathbf{a}}_{i-1,k}$ and $\hat{\mathbf{a}}_{i,k} = \hat{\mathbf{a}}_{i-1,k} + \varepsilon$, respectively, whenever $\mathbf{a}_{i-1,k} + \varepsilon > \mathbf{a}_{i,k}$. Considering the increment of the total reconstructed error E, where $E = \sum_{n=1}^{N} \parallel \mathbf{y}(n) - \hat{\mathbf{y}}(n) \parallel^2$ with $\hat{\mathbf{y}}(n)$ is the nth reconstructed parameter vector, caused by this change, RTD determines $\hat{\mathbf{a}}_{i-1,k}$ as

$$\hat{\mathbf{a}}_{i-1,k} = \frac{\mathbf{a}_{i-1,k} + \mathbf{a}_{i,k} - \varepsilon}{2}$$

However, this routine still does not assure the LSF ordering property for $\mathbf{a}_k$ since $\hat{\mathbf{a}}_{i-1,k} < \mathbf{a}_{i-1,k}$ and there is no guarantee that $\mathbf{a}_{1,k} > 0$ or $\mathbf{a}_{P,k} < \pi$. In this section, we propose an improved algorithm to deal with this problem.

Firstly, a more general routine for changing J components ($1 \leq J \leq P - i + 1$): $\mathbf{a}_{i,k}, \mathbf{a}_{i+1,k}, \cdots, \mathbf{a}_{i+J-1,k}$ to $\hat{\mathbf{a}}_{i,k}$, $\hat{\mathbf{a}}_{i+1,k}$=$\hat{\mathbf{a}}_{i,k}$+ε, $\cdots$, $\hat{\mathbf{a}}_{i+J-1,k}$=$\hat{\mathbf{a}}_{i,k} + (J-1)\varepsilon$, respectively, minimizing the increment of the total reconstructed error E is established. The increment of E caused by this change is calculated as follows. Here, $E' = \sum_{n=1}^{N} \parallel \mathbf{y}(n) - \tilde{\mathbf{y}}(n) \parallel^2$ is the newly total reconstructed error, where $\tilde{\mathbf{y}}(n)$ is the nth newly reconstructed parameter vector corresponding to the new value of the event target $\mathbf{a}_k$.

$$\begin{aligned}
\boldsymbol{\Delta} &= E' - E \\
&= \sum_{n=1}^{N} \sum_{j=1}^{P} \left[(y_j(n) - \tilde{y_j}(n))^2 - (y_j(n) - \hat{y_j}(n))^2 \right] \\
&= \sum_{n=1}^{N} \sum_{j=i}^{i+J-1} \left[2y_j(n) - \hat{y_j}(n) - \tilde{y_j}(n) \right] \left[\hat{y_j}(n) - \tilde{y_j}(n) \right]
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{n=1}^{N}\sum_{j=i}^{i+J-1}\left[2(y_j(n)-\hat{y}_j(n))+(\hat{y}_j(n)-\tilde{y}_j(n))\right]\left[\hat{y}_j(n)-\tilde{y}_j(n)\right] \\
&= 2\sum_{n=1}^{N}\sum_{j=i}^{i+J-1}\left(y_j(n)-\hat{y}_j(n)\right)\left(y_j(n)-\tilde{y}_j(n)\right)+\sum_{n=1}^{N}\sum_{j=i}^{i+J-1}\left(\hat{y}_j(n)-\tilde{y}_j(n)\right)^2 \\
&= 2\sum_{j=i}^{i+J-1}\left(a_{j,k}-\hat{a}_{j,k}\right)\sum_{n=1}^{N}\left(y_j(n)-\hat{y}_j(n)\right)\phi_k(n) \\
&+ \sum_{j=i}^{i+J-1}\left(a_{j,k}-\hat{a}_{j,k}\right)^2\sum_{n=1}^{N}\phi_k(n)^2 \qquad (4.10)
\end{aligned}
$$

It is somewhat complicated, but can be further simplified by considering that $a_{j,k}$, where $i \le j \le i+J-1$, minimize the error E. That is,

$$
\frac{\partial E}{\partial a_{j,k}} = 2\sum_{n=1}^{N}(y_j(n)-\hat{y}_j(n))\phi_k(n) = 0, \quad i \le j \le i+J-1.
$$

Hence,

$$
\begin{aligned}
\boldsymbol{\Delta} &= \sum_{j=i}^{i+J-1}\left(a_{j,k}-\hat{a}_{j,k}\right)^2\sum_{n=1}^{N}\phi_k(n)^2 \\
&= \sum_{l=0}^{J-1}\left[\mathbf{a}_{i+l,k}-(\hat{\mathbf{a}}_{i,k}+l\varepsilon)\right]^2\sum_{n}\phi_k(n)^2. \qquad (4.11)
\end{aligned}
$$

On the other hand,

$$
\hat{\mathbf{a}}_{i,k} \ge \mathbf{a}_{i-1,k}+\varepsilon.
$$

Therefore, $\hat{\mathbf{a}}_{i,k}$ should be determined as follows to minimize $\boldsymbol{\Delta}$:

$$
\hat{\mathbf{a}}_{i,k} = \begin{cases} \mathbf{a}_{i-1,k}+\varepsilon, & \text{if } \tilde{\mathbf{a}}_{i,k} < \mathbf{a}_{i-1,k}+\varepsilon \\ \tilde{\mathbf{a}}_{i,k}, & \text{otherwise} \end{cases} \qquad (4.12)
$$

where

$$
\tilde{\mathbf{a}}_{i,k} = \frac{\sum_{l=0}^{J-1}\mathbf{a}_{i+l,k}}{J} - \frac{(J-1)\varepsilon}{2}
$$

In the sequel, an algorithm for normalizing an event target $\mathbf{a}_k$ is developed. In order to assure that $\mathbf{a}_{1,k} > 0$ and $\mathbf{a}_{P,k} < \pi$, we add zero and π to $\mathbf{a}_k$ so that $\mathbf{a}_k = [0, \mathbf{a}_{1,k}, ..., \mathbf{a}_{P,k}, \pi]^T$. Zero and π are denoted as $\mathbf{a}_{0,k}$ and $\mathbf{a}_{P+1,k}$ for simplicity. Note that $\mathbf{a}_{0,k}$ and $\mathbf{a}_{P+1,k}$ cannot be changed during the normalization. The whole algorithm is depicted in Fig. 4.6 and described as follows:

1. Initialize $i \leftarrow 0$.
2. If $i < P$ and $\mathbf{a}_{i,k}+\varepsilon \le \mathbf{a}_{i+1,k}$, set $i \leftarrow i+1$. Repeat this step until $i = P$ or $\mathbf{a}_{i,k}+\varepsilon > \mathbf{a}_{i+1,k}$. If $i = P$, go to step 6.
3. If $i = 0$, set $i \leftarrow 1$ and $j \leftarrow 1$ since $\mathbf{a}_{0,k}$ could not be changed; if not, set $j \leftarrow 2$.

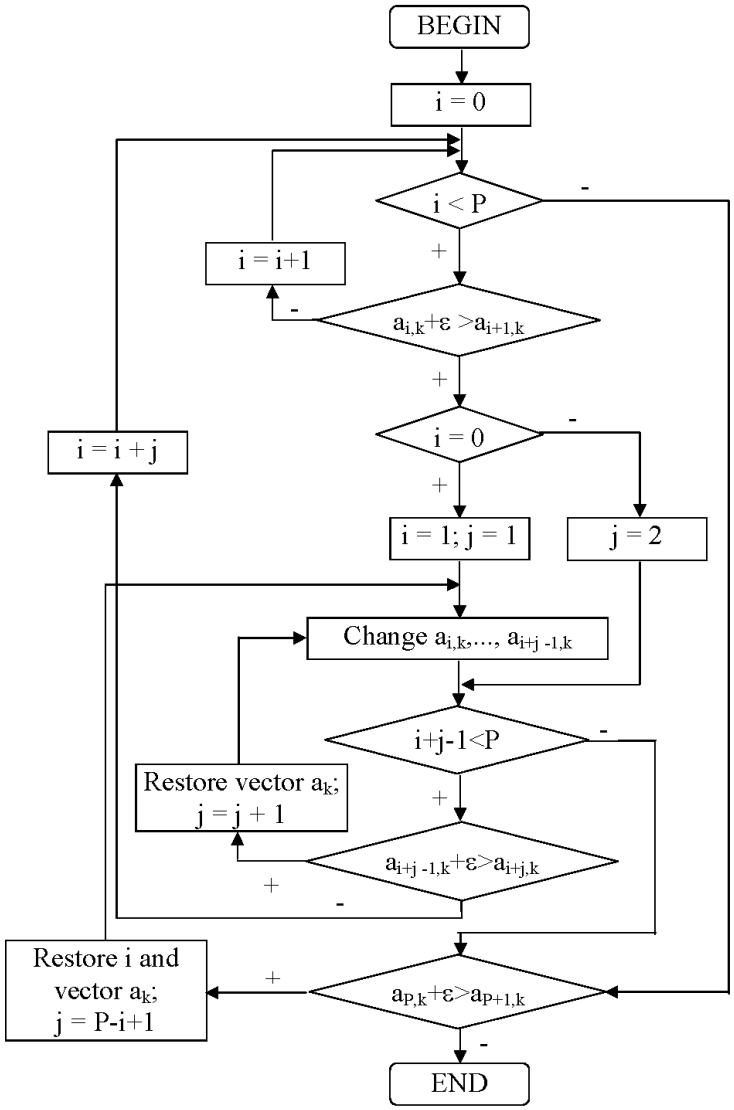


Figure 4.6: Block diagram of the improved algorithm for normalizing event targets.

4. Change $\mathbf{a}_{i,k}, ..., \mathbf{a}_{i+j-1,k}$ to $\hat{\mathbf{a}}_{i,k}, ..., \hat{\mathbf{a}}_{i+j-1,k}$ using Equation (4.12). If $i + j - 1 = P$, go to step 6.

5. If $\mathbf{a}_{i+j-1,k} + \varepsilon > \mathbf{a}_{i+j,k}$, restore $\mathbf{a}_k$ from the previous step, set $j \leftarrow j + 1$, and go back to step 4; if not, set $i \leftarrow i + j$. Go back to step 2 if $i < P$.

6. If $\mathbf{a}_{P,k} + \varepsilon \leq \mathbf{a}_{P+1,k}$, $\mathbf{a}_k$ has been normalized; if not, restore i and the corresponding value of vector $\mathbf{a}_k$ from the previous step, set $j \leftarrow P - i + 1$ and go back to step 4.

At step 6, it is of interest to notice that if i is the last component of a modified segment, i is then set to the beginning of that segment. In particular, if $i = 0$, vector $\mathbf{a}_k$ is set as $[0, \pi - P\varepsilon, \pi - (P-1)\varepsilon, ..., \pi]^T$. However, in practice this case almost never occurs.

In the result, when the locations of events n_k, where $k = 1, ..., K$, are known and the corresponding event targets are initialized with the samples of the LSF parameter vector

trajectory $\mathbf{y}(n_k)$, we can calculate proper event functions and event targets iteratively using Equations (4.8), (4.9), and (4.12). Here, we suggest using the local minima of the following spectral feature transition rate (SFTR) based on LSF parameters as the initial locations of events [71, 103].

$$\text{SFTR}: \quad s(n) = \sum_{i=1}^{P} c_i(n)^2, \quad 1 \le n \le N \tag{4.13}$$

where

$$c_i(n) = \frac{\sum_{m=-M}^{M} m\mathbf{y}_i(n+m)}{\sum_{m=-M}^{M} m^2}, \quad 1 \le i \le P \tag{4.14}$$

The window size, $2M$, of SFTR analysis is the only parameter that effects the initial number and locations of events. In addition, a new event is inserted where the initial reconstruction error $e(n) = \| \mathbf{y}(n) - \hat{\mathbf{y}}(n) \|^2$ has a local maximum larger than a certain threshold θ as considered in [71]. The way of segmenting input vectors for online analysis presented in [71] is also adopted in the MRTD method. The block diagram of the MRTD algorithm is given in Fig. 4.7.

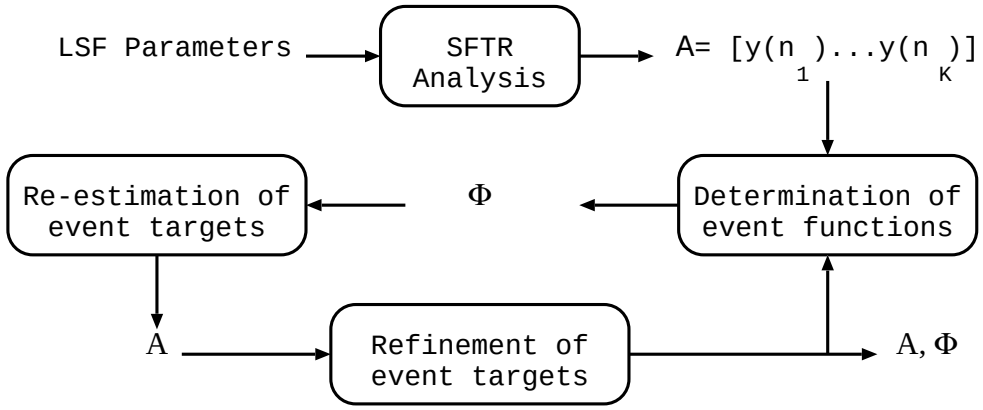


Figure 4.7: Block diagram of the MRTD algorithm.

4.3.3 Performance Evaluation

A set of 250 sentences of the ATR Japanese speech database [2] were selected as the speech data. This speech data set consists of about 20 minutes of speech spoken by 10 speakers (5 males & 5 females) re-sampled at 8 kHz sampling frequency. 10^{th} order LSF parameters were calculated using a LPC analysis window of 30 ms at 10 ms frame intervals, and TD analyzed using the original RTD and the MRTD methods in turn. Here, $2M = 4$, $\theta = 0.2$, and $\varepsilon = 0.01$ were empirically chosen as suitable values for the window size of SFTR analysis, the event insertion threshold, and the minimum dLSF, respectively.

Table 4.2 gives the summary of invalid-LSF event targets and well-shaped event functions obtained from the MRTD and RTD methods for the above speech data set. Results indicate that the drawbacks of RTD method described in Sections 4.3.1 and 4.3.2 have been overcome in the proposed MRTD method.

Log spectral distortion (LSD) measure [109, 110] was used to evaluate the interpolation performance of the proposed MRTD algorithm in comparison with that of the original

RTD. The LSD evaluated is that between the original LSF parameters, $\mathbf{y}(n)$, and the reconstructed LSF parameters after TD analysis, $\hat{\mathbf{y}}(n)$. Table 4.3 gives the summary of spectral distortion results obtained from the RTD and MRTD methods for the speech data set mentioned above. The distribution of the log spectral distortion in the form of histograms are shown in Fig. 4.8. Results indicate slightly better performance in the case of RTD over MRTD.

Shortly speaking, the drawbacks of RTD method in terms of invalid-LSF event targets and ill-shaped event functions can be solved with a negligible increase in spectral distortion. Note that LSD was calculated for the interpolation step only, i.e., before quantization.

Table 4.2: Percentage number of invalid-LSF event targets and well-shaped event functions for RTD and MRTD methods. The speech data set consists of 250 utterances spoken by 10 speakers (5 males and 5 females) of the ATR Japanese speech database.

Method	% invalid-LSF event targets	% well-shaped event functions
RTD	0.08%	88%
MRTD	0%	100%

Table 4.3: Event rate, average LSD, and percentage number of outlier frames for RTD and MRTD methods. The speech data set consists of 250 utterances spoken by 10 speakers (5 males and 5 females) of the ATR Japanese speech database.

Method	Event rate	Avg. LSD	2-4 dB	>4 dB
RTD	20.16 events/sec	1.563 dB	22.97%	0.96%
MRTD	20.16 events/sec	1.568 dB	23.15%	0.98%

Fig. 4.9 shows the plot of event functions obtained from the MRTD method for an example of a Female/Japanese speech utterance "*shimekiri ha geNshu desu ka.*" In Fig. 4.10, the plots of original and reconstructed LSF parameters after MRTD analysis are shown for the same speech utterance as utilized in Fig 4.9.

4.4 Vector Quantization of LSF Parameters Based on MRTD

We have conducted an experiment to evaluate the performance of MRTD based vector quantization (VQ) of LSF parameters. During the experiment, a set of 1890 phonetically-diverse sentences (SI set) from the TIMIT speech database [44], re-sampled at 8 kHz sampling frequency, were selected as the speech data. Among them, we chose 1386 sentences for training and 504 sentences for testing. 10^{th} order LSF parameters were calculated and then MRTD analyzed.

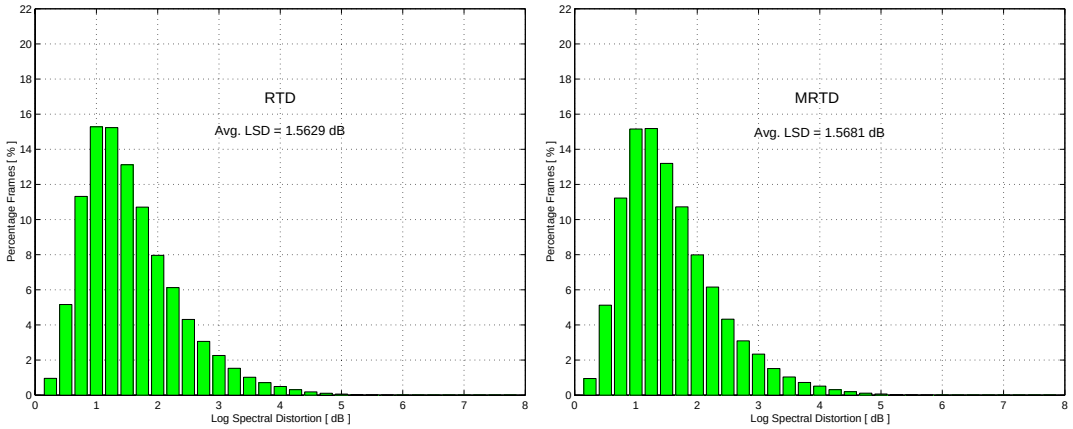


Figure 4.8: Distribution of the log spectral distortion (LSD) between the original and reconstructed LSF parameters in the form of histograms. Left: LSD histogram for RTD. Right: LSD histogram for MRTD. Speech data set consists of 250 sentence utterances spoken by 10 speakers (5 males & 5 females) of the ATR Japanese speech database.

Separate codebooks were used for the event targets and the event functions. Codebook generation was performed according to the LBG clustering algorithm [86]. Squared Euclidean distance was used as the distortion measure.

In the case of event functions, normalization of the event functions is necessary to fix the dimension of the event function vector space. Notice that only quantizing $\phi_k(n)$ in the interval $[n_k; n_{k+1}]$ is enough to reconstruct the whole event function $\phi_k(n)$. Moreover, $\phi_k(n)$ always starts from one and goes to zero in that interval and the type of decrease (after normalizing the length of $\phi_k(n)$ by simply truncating or zero-padding the original) can be vector quantized (see Fig. 4.11). Considering that most intervals between two consecutive central positions are less than 19 frames long, the normalized event function vector space thus has a dimension of 16.

Let R_1, R_2 and P be the bit allocation for coding each of event targets, normalized event functions, and event locations, respectively. R_1 and R_2 are the codebook sizes for event targets and event functions, and we select $P = 4$. Therefore, the final bit rate requirement to code spectral information of speech can be expressed as follows:

$$\text{Bit rate} = (R_1 + R_2 + P) \times \text{Event rate} \tag{4.15}$$

The average LSD between the original and the reconstructed LSF parameters (after quantization) were found to be 3.41 dB, 3.14 dB, 2.96 dB, and 2.81 dB for the codebook sizes of 8-5, 10-5, 11-6, and 12-7, respectively. By codebook size of *a*-*b* we mean that a is the codebook size for VQ of event targets, while b is that of normalized event functions. In comparison, we have also evaluated the VQ performance of LSF parameters over the same speech data for the codebook sizes of 6, 7, 8, and 9 bits. Fig. 4.12 shows the graph of average LSD against the bit rate requirement for spectral coding. As shown in the figure, an average LSD of about 2.8 dB can be achieved at the bit rate requirement of 460 bps with MRTD based VQ. Meanwhile, the bit rate requirement needed is 900 bps for VQ of LSF parameters (without MRTD) to achieve an equivalent result. This justifies the fact that spectral information of speech can be efficiently coded using MRTD based VQ of LSF parameters.

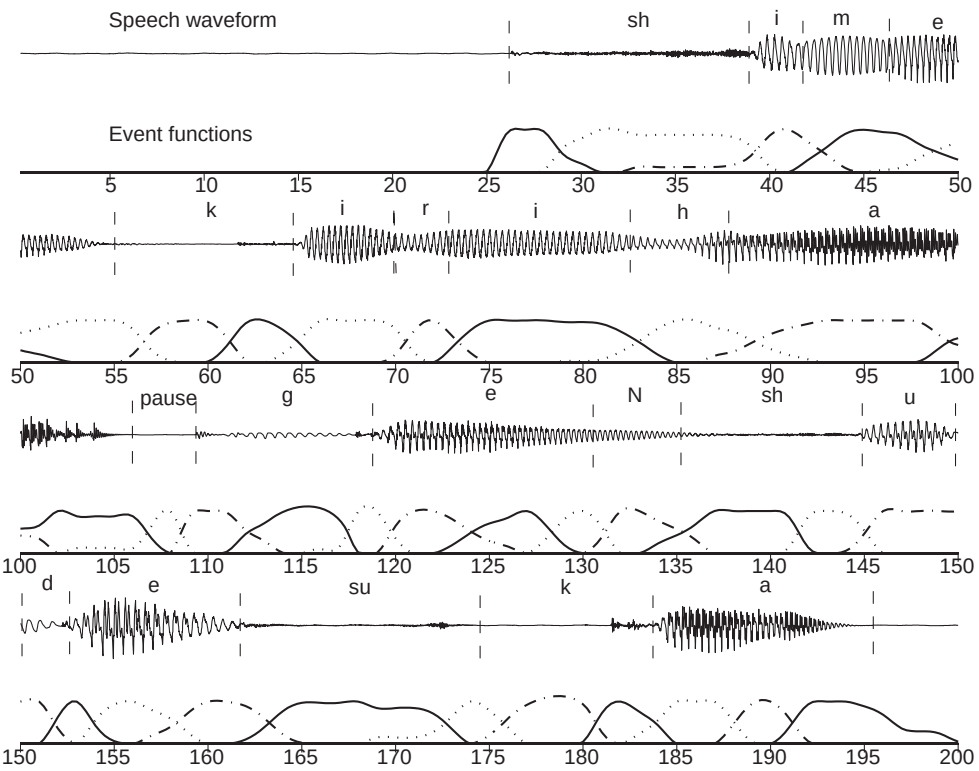


Figure 4.9: Plot of the event functions obtained from the MRTD method for the Female/Japanese speech utterance "*shimekiri ha geNshu desu ka*." The speech waveform is also shown together with the phonetic transcription for reference. The numerals indicate the frame numbers. Note that every event function is well-shaped.

Also, the event rate was found to be about 20 events/sec, which results in an average duration between central positions of two consecutive events of about 50 ms. Considering that the window size for the spectral feature transition rate (SFTR) calculation used in event localization is 40 ms and the LPC analysis window is 30 ms long, we have an average algorithmic delay of about 95 ms. Moreover, MRTD has significantly reduced the computational cost of TD because it uses neither the computationally costly singular value decomposition (SVD) routine nor the Gauss-Seidel iteration. These make MRTD suitable for real-time speech coder.

4.5 MRTD of Excitation Parameters

4.5.1 Determination of Excitation Targets

The MRTD technique is employed to describe the temporal characteristics of speech excitation parameters, i.e gain, pitch and voicing. The same event functions evaluated for LSF parameters are also used to describe the temporal pattern of the gain, pitch and voicing parameters. We are motivated by the fact that the speech production mechanism is assumed to be a synchronously controlled process with respect to the movement of different articulators, i.e. jaws, tongue, larynx, glottis etc. Therefore, we expect that the

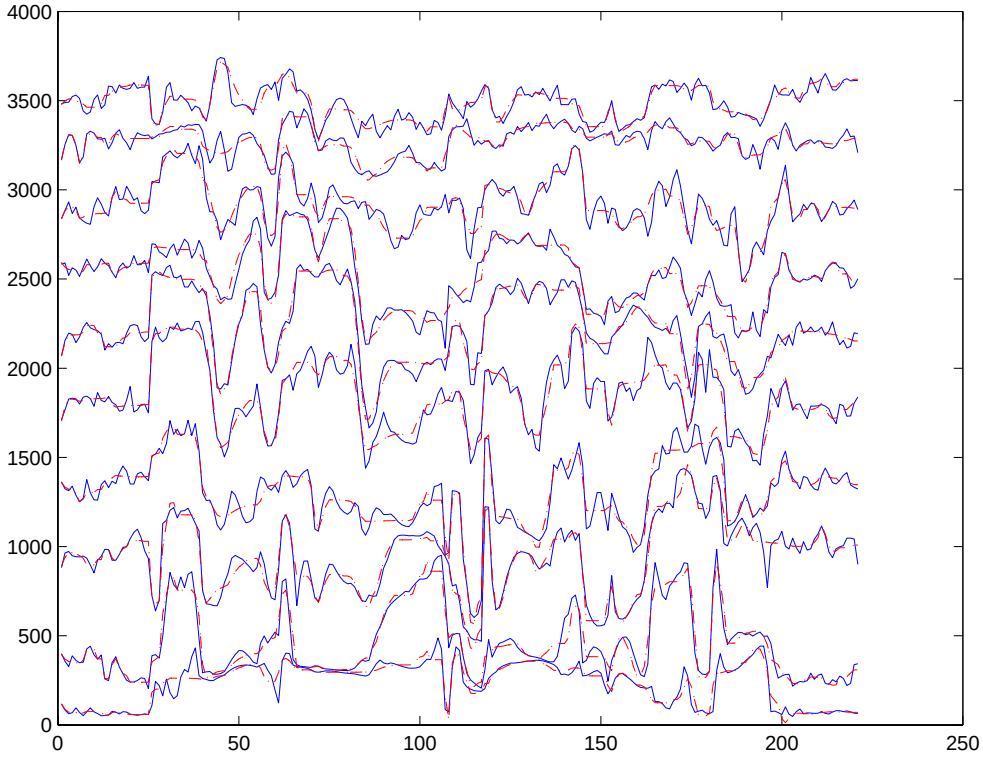


Figure 4.10: Plots of the original and reconstructed LSF parameters obtained from the MRTD method for the Female/Japanese speech utterance "*shimekiri ha geNshu desu ka.*" The solid line indicates the original LSF parameter vector trajectory and the dashed line indicates the reconstructed LSF parameter vector trajectory. The average log spectral distortion was found to be 1.5647 dB.

temporal evolutionary patterns of different properties of speech, i.e. spectrum, pitch, gain and voicing, can be described by a common set of event functions.

Let $b(n)$ be a excitation parameter, i.e. gain, pitch or voicing. Then $b(n)$ is approximated by $\hat{b}(n)$, the reconstructed excitation parameter for the nth frame, as follows in terms of excitation targets, b_ks, and event functions, $\phi_k(n)$s.

$$\hat{b}(n) = \sum_{k=1}^{K} b_k \phi_k(n), \quad 1 \leq n \leq N \tag{4.16}$$

In matrix notation, Equation (4.16) can be written as

$$\hat{\mathbf{B}} = \mathbf{A}_b \mathbf{\Phi} \tag{4.17}$$

where $\hat{\mathbf{B}}$ and $\mathbf{A}_b$ are the reconstructed excitation parameter vector and excitation target vector, respectively.

In Equation (4.17), the matrix of event functions, $\mathbf{\Phi}$, is known and therefore the excitation target vector, $\mathbf{A}_b$, should be determined so that the sum squared error between the original excitation parameters and the reconstructed excitation parameters is minimized. Consequently, $\mathbf{A}_b$ is calculated as follows:

$$\mathbf{A}_b = \hat{\mathbf{B}} \mathbf{\Phi}^T (\mathbf{\Phi} \mathbf{\Phi}^T)^{-1} \tag{4.18}$$

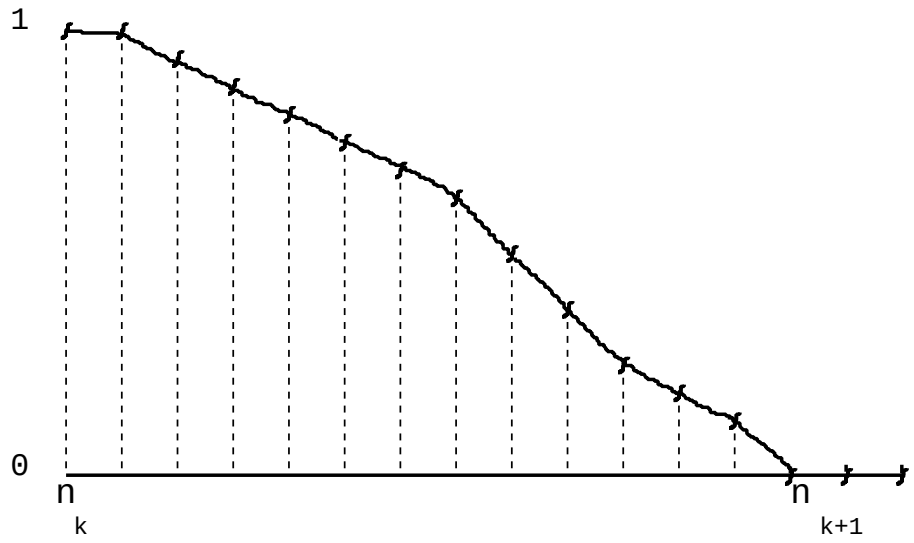


Figure 4.11: Example of zero-padding an event function.

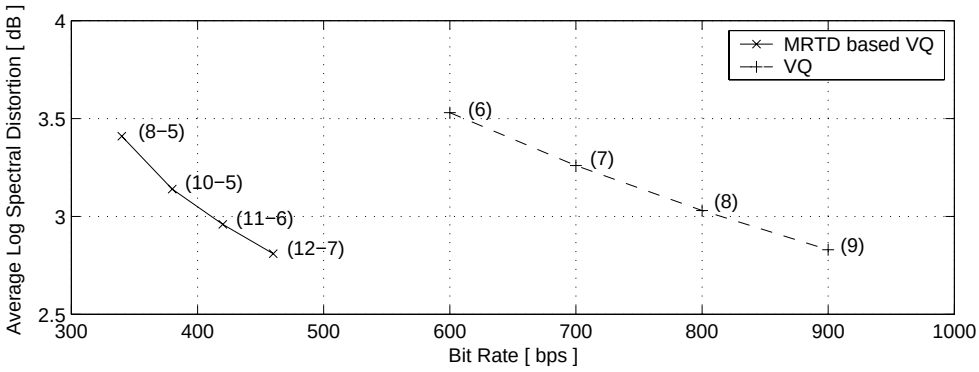


Figure 4.12: Spectral distortion against the bit rate requirement for spectral coding

In the case of pitch parameters, linear interpolation was used within the unvoiced segments to form a continuous pitch contour. In the case of voicing parameters, a hard limiter with a threshold value of 0.5 was used to determine the reconstructed binary voicing parameters and binary voicing targets, from the non-binary results of Equations (4.16) and (4.18), respectively.

4.5.2 Simulation Results

The gain, pitch and voicing parameters, hereafter indicated by $g(n)$, $p(n)$, and $v(n)$, respectively, were calculated at 10 ms frame intervals with a 40 ms analysis window, for the sentence utterance "*kantan na shiryou wo ookuri shimasu node, shibaraku omachi kudasai*," of the ATR Japanese speech database [2]. Each parameter contour was MRTD analyzed according to the procedure described above using the event functions obtained from MRTD analysis of LSF parameters.

Fig. 4.13 shows the plots of original and reconstructed gain parameters and the plot of frame-wise gain error, $e_g(n)$, where $e_g(n) = \hat{g}(n) - g(n)$. The root-mean-squared (RMS)

gain error, $\sqrt{E_g}$, where $E_g = \frac{1}{N}\sum_{n=1}^{N} {e_g}^2(n)$, was found to be about 4.37 dB.

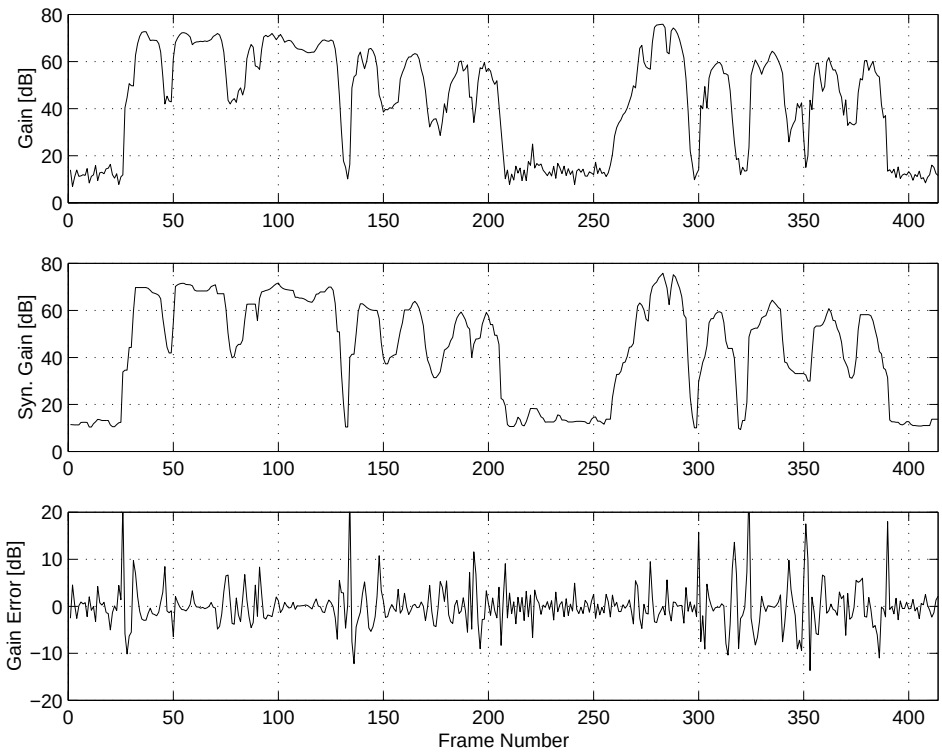


Figure 4.13: Original gain parameters, $g(n)$, reconstructed gain parameters, $\hat{g}(n)$, for the sentence utterance "*kantan na shiryou wo ookuri shimasu node, shibaraku omachi kudasai.*" The RMS gain error is 4.37 dB.

Fig. 4.14 shows the plots of original and reconstructed pitch frequency parameters and the plot of frame-wise pitch frequency error, $e_p(n)$, where $e_p(n) = \hat{p}(n) - p(n)$. The RMS pitch error, $\sqrt{E_p}$, where $E_p = \frac{1}{N}\sum_{n=1}^{N} {e_p}^2(n)$, was found to be about 2.09 Hz.

In the case of binary voicing parameters, the voicing error, $e_v(n)$, where $e_v(n) = \hat{v}(n) - v(n)$, appeared only at, but not all, voiced/unvoiced boundaries as error spikes of mostly 1 frame. The percentage number of frames with voicing errors was found to be about 4.59%. Fig. 4.15 shows the plots of original and reconstructed voicing parameters and the plot of frame-wise voicing error, $e_v(n)$.

Moreover, we have also evaluated the performance of MRTD in terms of excitation parameters over a set of 250 sentence utterances of the ATR Japanese speech database [2]. This speech data set consists of about 20 minutes of speech spoken by 10 speakers (5 males & 5 females) resampled at 8 kHz sampling frequency. The RMS gain error, RMS pitch error and percentage number of frames with voicing errors were found to be about 4.01 dB, 5.75 Hz and 4.74%, respectively. It was observed that the RMS gain error and RMS pitch error can be mainly attributed to some discrete time points, where the corresponding frame-wise gain error and pitch error obtained very high values. Meanwhile, no voicing errors were observed during continuous voiced and unvoiced segments, except for the points of voicing transitions.

The significant match between the original and reconstructed excitation parameters results in the fact that a common set of event functions can be used to describe the

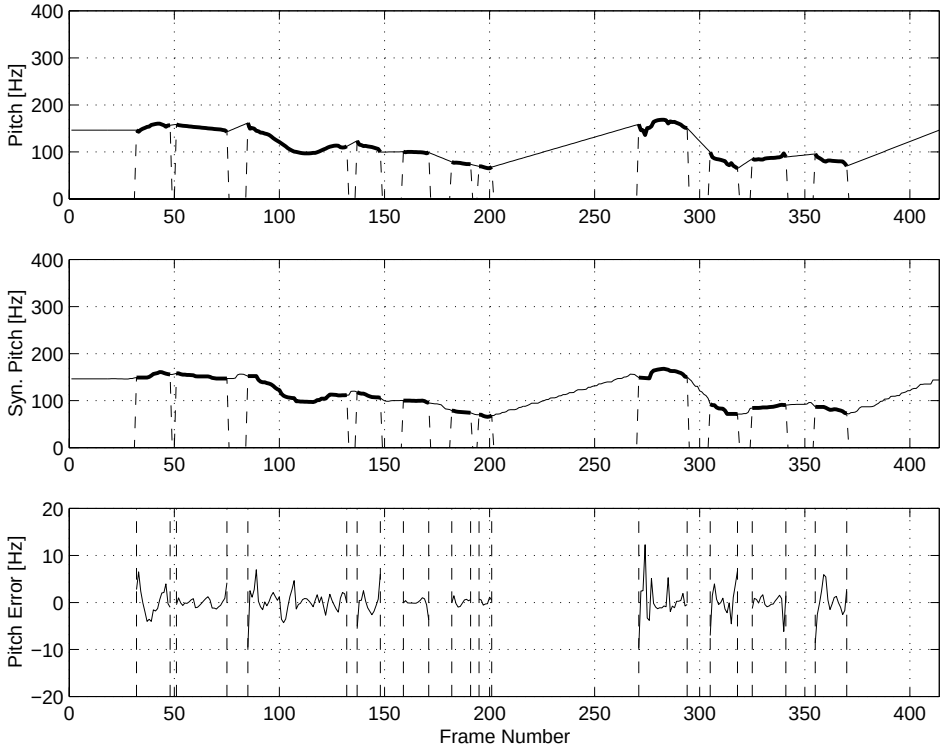


Figure 4.14: Original pitch parameters, $p(n)$, reconstructed pitch parameters, $\hat{p}(n)$, for the sentence utterance "*kantan na shiryou wo ookuri shimasu node, shibaraku omachi kudasai.*" The RMS pitch error is 2.09 Hz.

temporal patterns of both spectral and excitation parameters.

Also, we have evaluated the performance of the original restricted temporal decomposition (RTD) method [71] over the speech data set mentioned above. Experimental results obtained were about 4.03 dB, 5.8 Hz, and 4.75% for RMS gain error, RMS pitch error, and percentage of frames with voicing error, respectively, slightly higher than those obtained from the MRTD method.

4.5.3 Quantization of Excitation Targets

Since voicing targets can be quantized at 1 bit/target, in this section only the quantization of gain and pitch targets is presented.

Fig. 4.16 shows gain and pitch target contours for the sentence utterance "*kantan na shiryou wo ookuri shimasu node, shibaraku omachi kudasai.*" Note that the slow evolution of gain target and pitch target in voiced segments, interrupted by sudden jumps in unvoiced segments.

In this section, we propose a differential and logarithmic quantization scheme for gain and pitch targets. The logarithm of gain target as well as the logarithm of pitch target are quantized both in differential and a memoryless quantizer, and the best of two output values is transmitted to the receiver. The major advantage of this scheme is that the high correlation of consecutive gain and pitch targets during voiced segments can be exploited, without losing performance of unvoiced speech, where consecutive gain

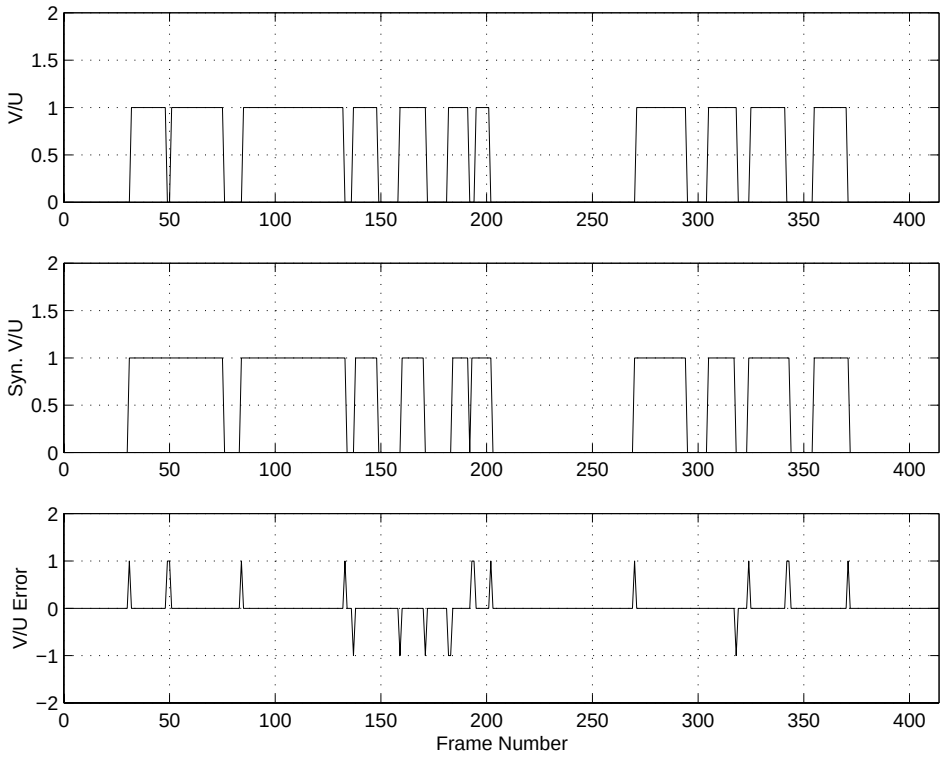


Figure 4.15: Original binary voicing parameters, $v(n)$, reconstructed binary voicing parameters, $\hat{p}(n)$, for the sentence utterance "*kantan na shiryou wo ookuri shimasu node, shibaraku omachi kudasai.*" The percentage number of frames with voicing errors is 4.59%.

and pitch targets have low correlation. Another advantage is that the proposed scheme quantizes rapidly and slowly changing gain and pitch targets in separate quantizers, which allows for different resolution for these two cases.

We applied the method of pitch quantization proposed in [36] to quantizing gain and pitch targets. The function of the algorithm is as follows: First the logarithm of gain target (respectively the logarithm of pitch target) is computed. The logarithmic gain target (respectively the logarithmic pitch target) is used in two branches of the algorithm. In the first branch, the gain target (respectively pitch target) is directly quantized in a uniform quantizer. In the second branch, the previous value of the quantized gain target (respectively pitch target) is subtracted before quantization, and added back after quantization, to form a differential quantization scheme. The output of the two branches are compared to the unquantized gain target (pitch target), and the best is selected for transmission. The full algorithm for quantization of gain target (respectively pitch target) using 5 bits is depicted in Fig. 4.17 and described as follows. Notice that t is denoted for gain target (respectively pitch target).

1. Initialization (this is only done for the first call), t_{min} and t_{max} are the minimum and maximum gain target (respectively pitch target), respectively.

   - $t_{range} = \log t_{max} - \log t_{min}$ (the range of logarithmic gain or pitch target).
   - $\varepsilon_1 = t_{range}/20 \times \{0, 1, 2, ..., 20\} + \log t_{min}$ (21 entries, index 0-20).

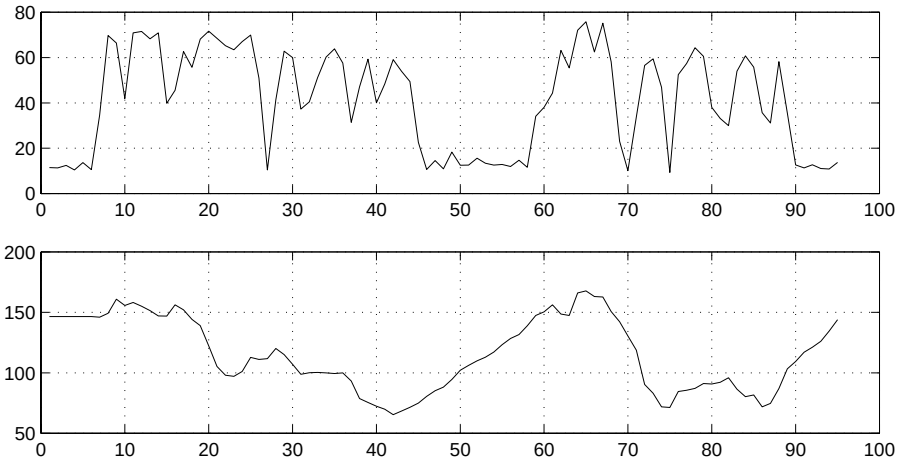


Figure 4.16: Top: gain target contour and bottom: pitch target contour, for the sentence utterance "*kantan na shiryou wo ookuri shimasu node, shibaraku omachi kudasai.*"

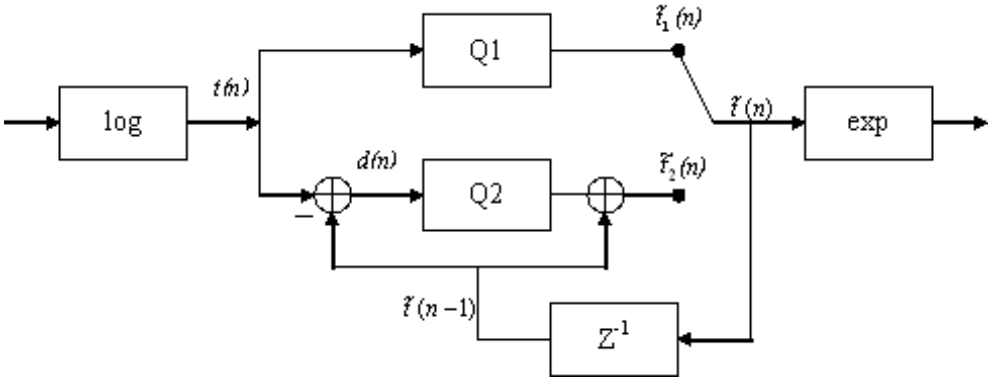


Figure 4.17: Block diagram of excitation target quantization scheme

- $\varepsilon_2 = \log 1.06 \times \{-5, -4, ..., 4, 5\}$ (11 entries, index 21-31).

2. Get an estimated gain target (respectively pitch target) $T(n)$, and compute the logarithmic gain target (respectively pitch target), $t(n) = \log T(n)$.
3. Find the closest value to $t(n)$ in ε_1: $\tilde{t}_1(n) = \arg\min_{c \in \varepsilon_1} \mid t(n) - c \mid^2$
4. Find the closest value to $d(n) = t(n) - \tilde{t}(n-1)$ in ε_2: $\tilde{t}_2(n) = \arg\min_{c \in \varepsilon_2} \mid d(n) - c \mid^2 + \tilde{t}(n-1)$
5. Compare $\tilde{t}_1(n)$ and $\tilde{t}_2(n)$ to $t(n)$, and select the best.

The index to the selected codebook entry (see definition of ε_1 and ε_2 for the index assignment) is output.

4.5.4 Experimental Results

The gain and pitch parameters $g(n)$ and $p(n)$, respectively, were calculated at 10 ms frame intervals with a 40 ms analysis window, for the sentence utterance "*kantan na shiryou wo*

ookuri shimasu node, shibaraku omachi kudasai." Each parameter contour was MRTD analyzed using the event functions obtained from MRTD analysis of LSF parameters. After that, gain and pitch targets were quantized and transmitted using the procedure described above.

Fig. 4.18 shows the plots of original gain parameters and reconstructed gain parameters (after quantization), and the plot of frame-wise gain error, $e_g(n)$, where $e_g(n) = \tilde{g}(n) - g(n)$. The RMS gain error, $\sqrt{E_g}$, where $E_g = \frac{1}{N}\sum_{n=1}^{N} {e_g}^2(n)$, was found to be about 5.74 dB.

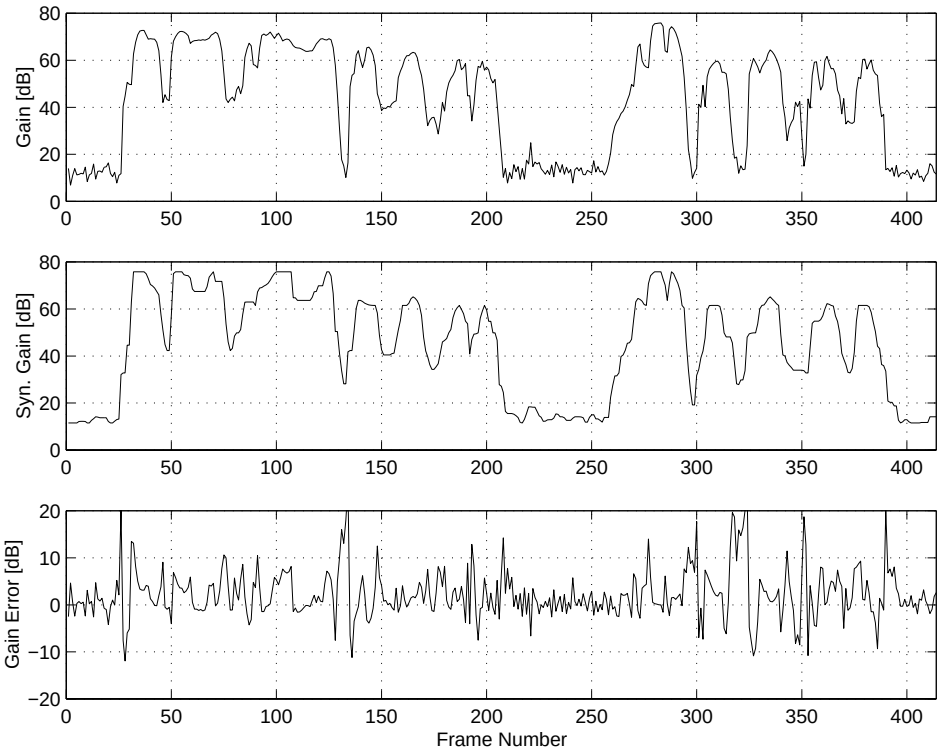


Figure 4.18: Original gain parameters, $g(n)$, reconstructed gain parameters after quantization, $\tilde{g}(n)$, for the sentence utterance "*kantan na shiryou wo ookuri shimasu node, shibaraku omachi kudasai.*" The RMS gain error is 5.74 dB.

Fig. 4.19 shows the plots of original pitch parameters and reconstructed pitch parameters (after quantization), and the plot of frame-wise pitch error, $e_p(n)$, where $e_p(n) = \tilde{p}(n) - p(n)$. The RMS pitch error, $\sqrt{E_p}$, where $E_p = \frac{1}{N}\sum_{n=1}^{N} {e_p}^2(n)$, was found to be about 2.61 Hz.

In the case of binary voicing parameters, results obtained after quantization are the same in comparison with those obtained from MRTD analysis only since binary voicing target can be transmitted accurately using 1 bit/target.

Also, we have evaluated the performance of MRTD analysis and quantization in terms of gain and pitch parameters over the set of 250 Japanese sentence utterances mentioned earlier. The RMS gain error, RMS pitch error were found to be about 6.27 dB, 9.59 Hz, respectively. It was also observed that the RMS gain error and RMS pitch error can be mainly attributed to some discrete time points, where the corresponding frame-wise gain error and pitch error obtained very high values.

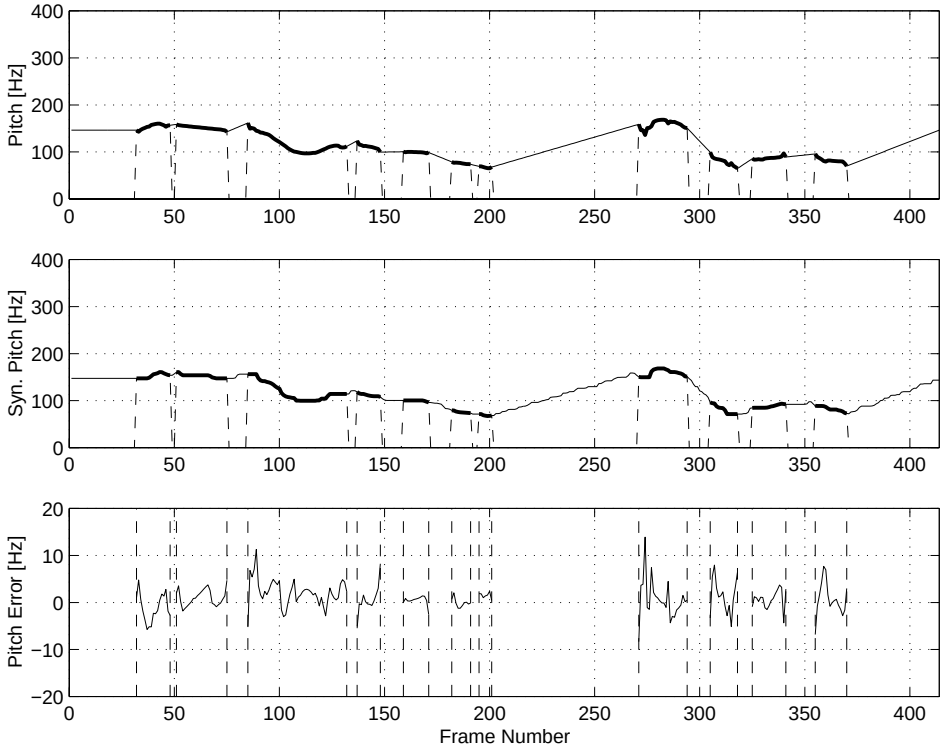


Figure 4.19: Original pitch parameters, $p(n)$, reconstructed pitch parameters after quantization, $\tilde{p}(n)$, for the sentence utterance "*kantan na shiryou wo ookuri shimasu node, shibaraku omachi kudasai*." The RMS pitch error is 2.61 Hz.

The significant match between the original and reconstructed excitation parameters after MRTD analysis and quantization results in the fact that MRTD can be regarded as a reasonable technique of analyzing and quantizing excitation information of speech.

4.6 Conclusion

We have presented a method of temporal decomposition, MRTD, for the LSF parameters. The additional constraint on the event functions in the second order TD model makes them monotonic during the transition from one event towards the next, from which the event functions can describe the temporal structure of speech more effectively. Also, this reduces the quantization error of event functions when vector quantized. The ordering property of LSFs has completely been ensured for the event targets using the improved algorithm so that MRTD can be used for decomposing the LSF parameters.

We have also extended the MRTD technique to describe the temporal patterns of speech excitation parameters. Results from this description, namely, excitation targets can be quantized efficiently at 11 bits/target, which results in a bit-rate of 220 bps required for encoding excitation information of speech. The low reconstruction error in excitation parameters after MRTD analysis as well as excitation target quantization justifies the fact that speech excitation parameters can not only be well represented by excitation targets using a common set of event functions derived from MRTD analysis of LSF parameters, but also can be encoded efficiently by quantizing and transmitting the excitation targets.

Chapter 5

Very Low-Bit-Rate Speech Coding Based on STRAIGHT Using Temporal Decomposition

5.1 Introduction

As shown earlier, speech in the temporal decomposition (TD) model is no longer represented by a vector updated frame by frame, but instead by the continuous trajectory of a vector. The trajectory is decomposed into a set of phoneme-like events, i.e. a series of temporally overlapping event functions and a corresponding series of event targets. Since the updating rate of events is much less than the frame rate, TD has been considered for efficient coding of spectral parameters, such as in [8, 49, 71, 103, 122].

STRAIGHT (stands for "Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum") proposed by Kawahara et al. [67] is a very high-quality vocoder. STRAIGHT can decompose a speech waveform into spectral envelopes, i.e. spectrogram, F0 (fundamental frequency) information, and noise ratios. Those parameters and the maximum value of amplitude are required for synthesizing speech. The spectrogram derived from STRAIGHT is very smooth thanks to a time-frequency interpolation procedure. It follows that line spectral frequency (LSF) parameters extracted from the spectrogram are correlated among frames, and thus the corresponding LSF parameter vector trajectory is smooth also. It is not the case of a normal LPC analysis method, where LSF parameters are extracted independently on a frame-by-frame basis. To make STRAIGHT applicable to very low-bit-rate speech coding, the bit rate required to represent the spectral envelope must be minimized. Since the spectral envelope can be further analyzed into LSF parameters and gain information, the Modified Restricted Temporal Decomposition (MRTD) algorithm can be incorporated with STRAIGHT to create high-quality speech coders working at low-bit rates.

We have investigated the application of MRTD in speech coding. In this chapter, a method of very low-bit-rate speech coding based on STRAIGHT, a very high-quality speech analysis-synthesis method [67], using MRTD is presented. Here, spectral information of speech is encoded using MRTD based vector quantization (VQ), whilst other speech parameters are encoded using scalar quantization (SQ). As a result, very low-bit-rate speech coders operating at rates around 1.2 kbps have been realized. Subjective test results indicate that the speech quality of this speech coding method is close to that of the

4.8 kbps US Federal Standard (FS-1016) CELP coder. The encoder and decoder block diagrams are shown in Fig. 5.1 and a detailed description of the proposed speech coding method is shown in the sections followed.

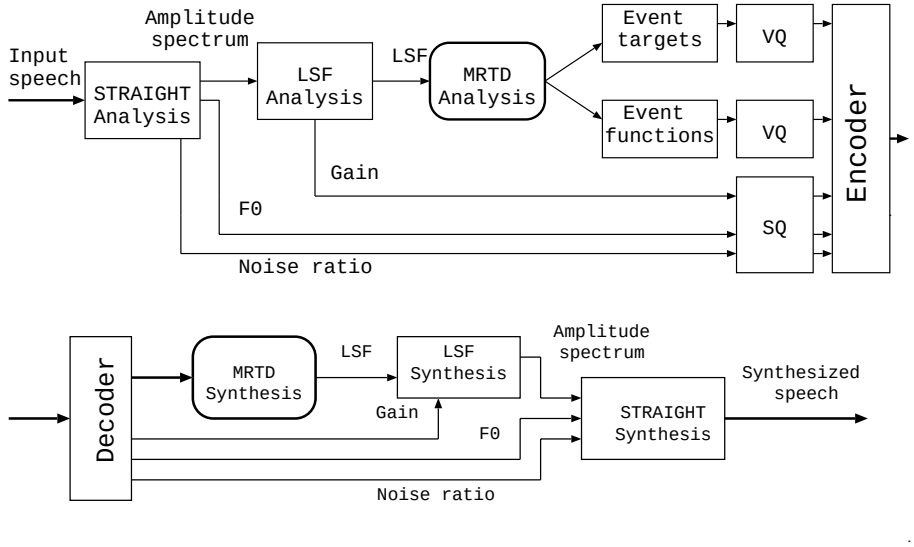


Figure 5.1: Proposed speech encoder and decoder block diagrams (top: encoder, bottom: decoder).

5.2 Determination of LSF's Order

5.2.1 Spectral Distortion vs. LSF's Order

Log spectral distortion (LSD) [109, 110] measure was used as the objective measure of performance to determine the suitable order of LSFs. A set of 112 phoneme balanced sentences uttered by speaker MMY of the ATR Japanese speech database [2] were used as the speech data. This speech data set were re-sampled at 8 kHz sampling frequency and then STRAIGHT analyzed. In the following, the spectral envelopes obtained from STRAIGHT analysis were transformed to LSF parameters of orders 14, 18, 22, 26, 30, and 34 using the procedure described in Section 2.2.2. Finally, the resulting LSF parameters were MRTD analyzed.

The spectral distortion results obtained from STRAIGHT analysis and LSF transformation, abbreviated as STRAIGHT-LSF, from STRAIGHT analysis, LSF transformation, and MRTD analysis, abbreviated as STRAIGHT-LSF & MRTD, are shown in Fig. 5.2. Note that these results were obtained before the quantization step. The horizontal and vertical axes indicate the order of LSFs and the average log spectral distortion, respectively. Results show that a considerable reduction in spectral distortion for STRAIGHT-LSF & MRTD is not achieved when the order of LSFs exceeds the order of 22.

5.2.2 Quality of Synthesized Speech vs. LSF's Order

We used the Scheffe's method of paired comparison [125] to subjectively evaluate the quality of the synthesized speech as a function of the LSF's order. Six graduate students

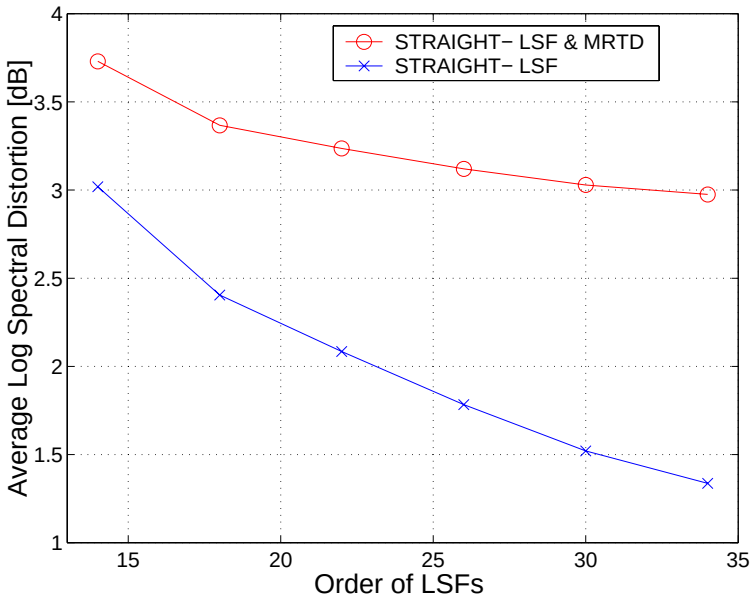


Figure 5.2: Spectral distortion vs. the order of LSFs.

known to have normal hearing ability were recruited for the listening experiment. Each listener was asked to make one of the following statements for each ordered pair of stimuli (i, j).

(-2) Distortion of i is much larger than that of j.
(-1) Distortion of i is slightly larger than that of j.
(0) Distortion of i is equivalent to that of j.
(1) Distortion of j is slightly larger than that of i.
(2) Distortion of j is much larger than that of i.

Two phoneme balanced sentences uttered by speaker MMY of the ATR Japanese speech database [2] were re-sampled at 8 kHz sampling frequency, and then analyzed by STRAIGHT. The resulting spectral envelopes were transformed to LSF parameters of orders 10, 14, 18, 22, 26, and 30. In the following, the LSF parameters were analyzed into event targets and event functions using the MRTD method. Those event targets and event functions were combined to reconstruct LSF parameters used for synthesizing stimuli.

Fig. 5.3 shows the results of the listening experiment. In this figure, the positions on the horizontal axis indicate the relative distances of stimuli. Here, the positive values mean that the distortion is small whilst the negative values indicate the high distortion. The number on each arrow corresponds to the order of LSFs. Results also show that an increase in distortion of the speech quality is not easily realized when the order of LSFs exceeds the order of 22.

For the reasons presented in Sections 5.2.1 and 5.2.2, the 22nd LSF parameters were considered for use in the proposed speech coding method.

5.3 MRTD Based VQ of LSF Parameters

The reason for interpolating the vector trajectory of LSF parameters by using TD is that the updating rate of events is much less than the frame rate, and both event targets and

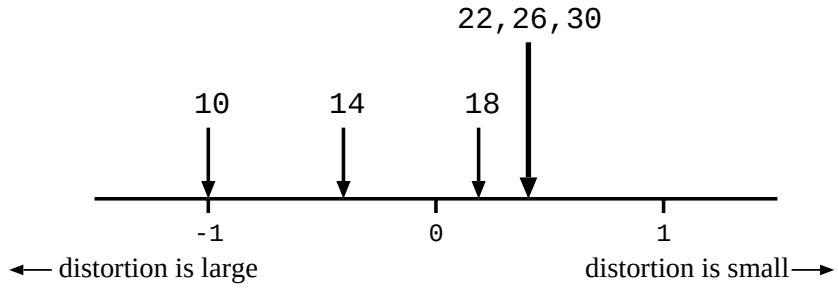


Figure 5.3: Speech quality vs. the order of LSFs.

event functions can be quantized efficiently. In other words, the LSF parameters can be quantized efficiently by transforming them into the event sequences first, and then quantizing event targets and event functions.

5.3.1 VQ of Event Targets

As shown in Chapter 4, the event targets obtained from the MRTD method are valid LSF parameter vectors, they therefore can be quantized by usual quantization methods for the LSF parameters. Here, the Split-VQ method [109] was adopted. Due to the distribution of LSFs, the event targets were divided into three subvectors of dimensions 7, 7, 8 and each subvector was quantized independently. We assigned 8 or 9 bits to each subvector, which resulted in the number of bits allocated to one event target was 24 or 27, respectively.

5.3.2 VQ of Event Functions

In the case of event functions, normalizing event functions is necessary to fix the dimension of the event function vector space. Notice that only quantizing $\phi_k(n)$ in the interval $[n_k; n_{k+1}]$ is enough to reconstruct the whole event function $\phi_k(n)$. Moreover, $\phi_k(n)$ always starts from one and goes down to zero in that interval, and the type of decrease (after normalizing the length of $\phi_k(n)$) can be vector quantized. Therefore, an event function $\phi_k(n)$ can be quantized by its length $L(k) = n_{k+1} - n_k$ and shape in $[n_k+1; n_{k+1}-1]$. In this work, 10 equidistant samples were taken from each event function for length-normalization and then vector quantized by a 7-bit codebook. Considering that all intervals between two consecutive event locations are less than 256 frames long (note that the frame period used in STRAIGHT analysis is 1 ms long), we used 8 bits for quantizing the length of each event function.

5.4 Coding Excitation Parameters

5.4.1 Coding F0 Parameters

For encoding F0 information, the lengths of voiced and unvoiced segments were quantized by using SQ first, with an average bit rate of 36 bps. In the following, linear interpolation was used within the unvoiced segments to form a continuous F0 contour. The continuous

F0 contour was re-sampled at 28 ms intervals, and then quantized by a 5-bit logarithmic quantizer.

In the decoder, F0 values were reconstructed from the quantized samples using the linear interpolation. In the sequel, F0 values of unvoiced intervals were set to zero. The root-mean-squared (RMS) F0 error was found to be about 3.7 Hz for the speech data set used in Section 5.2.1.

5.4.2 Coding Gain Parameters

The gain contour was re-sampled at 20 ms intervals. Logarithmic quantization was performed using 6 bits for each sampled value. The quantized samples and the spline interpolation were used in the decoder to form the reconstructed gain contour. The RMS gain error was found to be about 4.6 dB for the speech data set used in Section 5.2.1.

5.4.3 Coding Noise Ratio Parameters

The noise ratio parameters were estimated from the noise ratio targets and the event functions as follows:

$$\hat{i}(n) = \sum_{k=1}^{K} i_k \phi_k(n), \quad 1 \leq n \leq N \tag{5.1}$$

where $\hat{i}(n)$ and i_k are the reconstructed noise ratio parameter for the n^{th} frame and the k^{th} noise ratio target, respectively. The noise ratio targets were determined by minimizing the sum squared error, E_i, between the original and the interpolated noise ratio parameters.

$$E_i = \sum_{n=1}^{N} \Big(i(n) - \hat{i}(n) \Big)^2 = \sum_{n=1}^{N} \Big(i(n) - \sum_{k=1}^{K} i_k \phi_k(n) \Big)^2 \tag{5.2}$$

where $i(n)$ is the original noise ratio parameter for the n^{th} frame. The noise ratio targets were quantized by using SQ with 5 bits. The RMS noise ratio error was found to be about 0.1 for the speech data set used in Section 5.2.1.

5.5 Bit Allocation

The bit allocation for the proposed speech coding method is shown in Table 5.1. The average number of events per second, i.e. the event rate, was set as 15 events/sec. We allocated 8 bits and 9 bits to each subvector of the event targets, which resulted in 1.18 kbps and 1.23 kbps speech coders, respectively.

5.6 Subjective Tests

In order to evaluate the performance of the proposed speech coding method, the quality of the reconstructed speech was compared to that of well-known speech coders: the 4.8 kbps FS-1016 CELP and 2.4 kbps FS-1015 LPC-10E coders.

A listening experiment was carried out by using the Scheffe's method of paired comparison [125] similarly to that in Section 5.2.2. A set of 108 phoneme balanced sentences

Table 5.1: Bit allocation for the proposed speech coders.

Parameter	Proposed Coder 1	Proposed Coder 2
Event target	24 bits (8+8+8)	27 bits (9+9+9)
Event function	7 bits	7 bits
Event location	8 bits	8 bits
Noise ratio target	5 bits	5 bits
Subtotal A (sum × event rate)	660 bps	705 bps
F0	215 bps	215 bps
Gain	300 bps	300 bps
Maximum amplitude of input speech	5 bps	5 bps
Subtotal B	520 bps	520 bps
Total (A+B)	1180 bps	1225 bps

of the ATR Japanese speech database [2] were selected as the training data for the proposed speech coders. Speakers were 3 males & 3 females reading each of sentences. These speech utterances were re-sampled at 8 kHz sampling frequency, and then STRAIGHT analyzed using the frame shift of 1 ms. 22nd order LSF transformation was performed and the resulting LSF parameters were MRTD analyzed. Two phoneme balanced sentences, which are out of training set, uttered by a male and a female were used as the testing data. Stimuli were synthesized by using the following coders: 4.8 kbps FS-1016 CELP, 2.4 kbps FS-1015 LPC-10E, proposed 1.18 kbps speech coder 1, and proposed 1.23 kbps speech coder 2. Also, four other stimuli were STRAIGHT synthesized using the speech parameters obtained from STRAIGHT-LSF and STRAIGHT-LSF & MRTD.

Results of the listening experiment are shown in Fig. 5.6. It can be seen from this figure that the quality of the reconstructed speech obtained from the proposed speech coder 2 is close to that of the 4.8 kbps FS-1016 CELP coder and is much better than that of the 2.4 kbps FS-1015 LPC-10E coder.

As shown previously, the reconstructed LSF parameters after RTD analyzed and synthesized may be invalid, which causes the reconstructed speech to be noisy as well as to have click tones. We therefore did not evaluate the performance of the method for speech coding using RTD.

5.7 Conclusion

We have described a method of very low-bit-rate speech coding based on STRAIGHT using the MRTD algorithm, where MRTD based vector quantization is used for encoding spectral information of speech. As a result, two very low-bit-rate speech coders operating at rates around 1.2 kbps were produced. Although the quality of the reconstructed speech is little bit lower than that of the 4.8 kbps FS-1016 CELP coder according to the listening experiment, it is much better than that of the 2.4 kbps FS-1015 LPC-10E coder. However, the speech quality of the proposed speech coding method can be improved by increasing the event rate, which results in an increase in the bit-rate required for encoding speech.

It is necessary to evaluate other attributes of the proposed speech coding method,

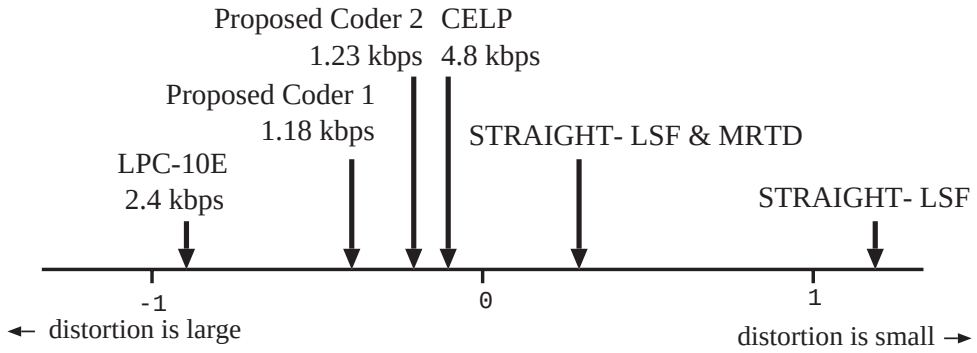


Figure 5.4: Results of the listening experiment.

for example, algorithmic delay, complexity, and noise robustness. In this work the event rate was set as 15 events/sec, thus resulting in an average algorithmic delay of about 90 ms. We can add additional events, if necessary, to keep the algorithmic delay below 100 ms. Meanwhile, the computational cost and noise robustness of the proposed speech coding method depend mainly on STRAIGHT. This is because MRTD has significantly reduced the computational cost of TD by avoiding the use of the computationally costly singular value decomposition routine and the adaptive Gauss-Seidel iterations used in Atal's method. On the other hand, MRTD can be applied to analyzing any LSF parameter vector trajectory. Currently, a real-time method for STRAIGHT based very low-bit-rate speech coding using MRTD still remains for future research.

Chapter 6

On the Application of Temporal Decomposition to Speaker Recognition

6.1 Introduction

Speaker recognition is the process of automatically recognizing the person speaking based on individual information included in speech waves. There are two types of tasks within speaker recognition: identification and verification. The objective of a speaker identification (ID) system is to determine the identity of an individual from a sample of his or her voice. Speaker ID can be further subdivided into two categories: closed set or open set. A closed-set speaker ID system identifies the speaker as one of those enrolled, even if he or she is not actually enrolled in the system. On the other hand, an open-set speaker ID system should be able to determine whether a speaker is enrolled or not, if enrolled, determine his or her identity [5].

Another distinguishing aspect of speaker recognition systems is that they can be either text-dependent or text-independent. In the text-dependent case, the input sentence or phrase is fixed for each speaker, whereas in the text-independent case, there is no restriction on the sentence or phrase to be spoken. Speaker ID consists of two stages, namely, feature extraction and classification as shown in Fig. 6.1. This chapter focuses on the feature extraction aspect of the problem of text-independent closed-set speaker ID.

Feature extraction is the process of deriving a compact set of parameters that are characteristics of a given speaker. Ideally, these parameters should efficiently preserve all the information relevant to the speaker's identity while eliminating any irrelevant information. That is, they should minimize the intra-speaker variance and at the same time maximize the inter-speaker variances [5]. The majority of speaker recognition systems use some types of short-term spectral analysis. The most effective and widely used spectral analysis techniques for speaker recognition are linear prediction (LP) analysis [5, 7] and filter bank analysis [74, 118]. This chapter focuses on LP-derived features, namely, event targets that are extracted from the line spectral frequency (LSF) parameters using the so-called temporal decomposition (TD) technique [8].

The state-of-the-art in classification techniques used in speaker recognition include Dynamic Time Warping (DTW), Hidden Markov Modeling (HMM), Gaussian Mixture Modeling (GMM), and Vector Quantization (VQ) [43]. In this chapter, the VQ-based

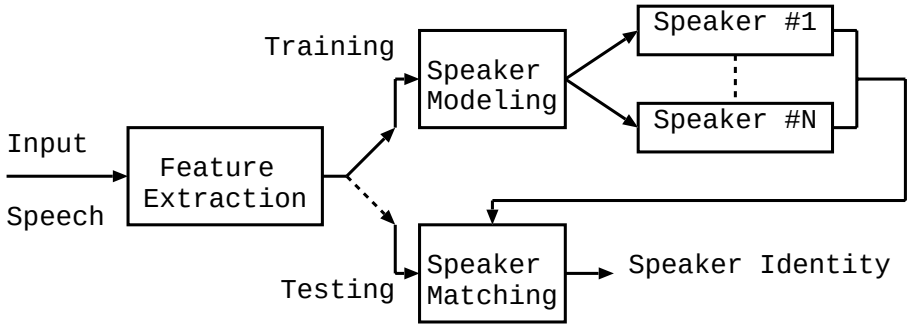


Figure 6.1: Block diagram of speaker identification systems.

speaker identification is used due to ease of implementation and high accuracy. It is well-known that the VQ approach has demonstrated good performance on limited vocabulary tasks. However, this method is somewhat impractical when the number of training and/or testing vectors is large, since the memory and amount of computation required become prohibitively high. Alternatively, event targets as a new set of features for speaker recognition can help to alleviate this problem.

The rest of this chapter is organized as follows: In Section 6.2, we briefly review the baseline VQ-based speaker ID. Next, the process of event target extraction is described in Section 6.3 and experimental results are reported in Section 6.4. Finally, conclusions are drawn in Section 6.5.

6.2 VQ-Based Speaker Identification

Vector quantization (VQ) is a process of mapping vectors from a large vector space to a finite number of regions in that space. Each region is called a cluster and can be represented by its center called a codeword. The collection of all codewords is called a codebook. VQ is used in both training and matching phases of a VQ-based speaker ID system. The system is shown in Fig. 6.2.

In the training phase, a speaker-specific VQ codebook is generated for each known speaker by clustering his or her training acoustic vectors based on the LBG algorithm [86]. There will be M codebooks, one pertaining to each of the M speakers.

In the matching phase, an input utterance of an unknown speaker is converted to a set of feature vectors. Consider a particular test feature vector. This is quantized by each of M codebooks. The quantized vector is that which is closest according to some distance measure to the test feature vector. We use the squared Euclidean distance as the measure. The distance from a test feature vector to the closest codeword is called a VQ distortion. Hence, M different VQ distortions are recorded, one for each codebook. This process is repeated for every test feature vector. The total distortion is computed over the entire set of feature vectors. The codebook corresponding to smallest total distortion identifies the speaker. When many utterances are tested, the success rate is the number of utterances for which the speaker is identified correctly divided by the total number of utterances tested.

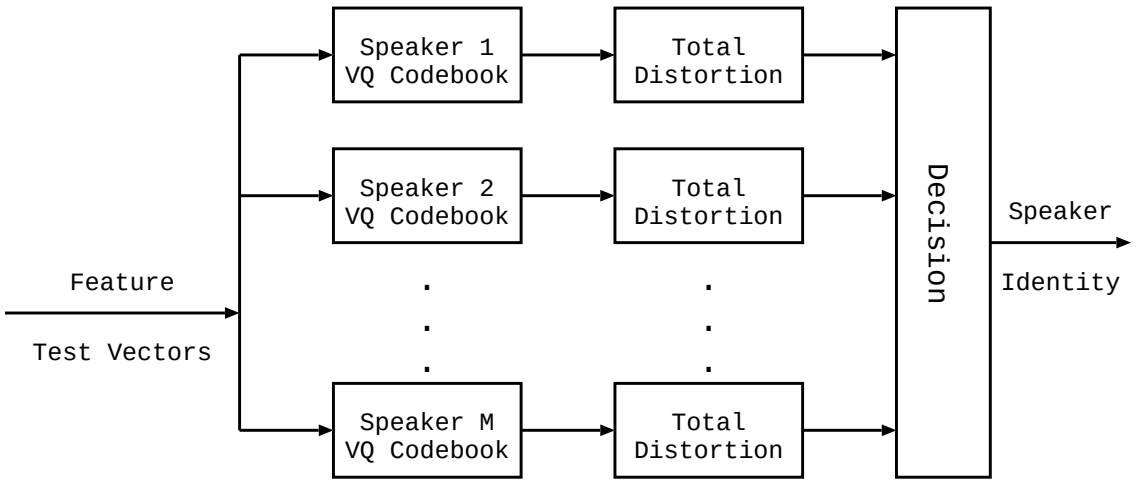


Figure 6.2: Block diagram of VQ-based speaker identification systems.

6.3 Extraction of Event Targets

As mentioned earlier, in articulatory phonetics, speech is described as a sequence of distinct gestures, each of which produces an acoustic event that should approximate an phonetic target [141]. Due to the overlap of the gestures, these phonetic targets are often partly realized. The temporal decomposition of speech initiated by Atal [8] is a technique of modeling the spectral parameter trajectory in terms of a sequence of overlapping event functions and corresponding event targets. The event targets are assumed to be associated with ideal articulatory target positions. It would be clear to mention that this is only possible for each speech event only one event function, and thus only one event target, is found. In this chapter, the event targets are extracted from LSF parameters using the Modified Restricted Temporal Decomposition (MRTD) method described in Chapter 4. These event targets are valid LSF parameters of the same dimension as a frame of input LSF parameters.

The application of TD to VQ-based speaker ID is motivated by the fact that TD is promising as a means of segmenting speech into a sequence of overlapping events closely related to phonetic structure of the speech signals [141, 143]. On the other hand, the VQ-based speaker ID can be regarded as a method that use phoneme-class-dependent speaker characteristics in short-term spectral features through implicit phoneme-class recognition. In other words, phoneme-classes and speakers are simultaneously recognized in this method [43]. Therefore, the event targets extracted from spectral parameters using TD can be considered as a new set of features for VQ-based speaker ID.

Fig. 6.3 shows the log power spectrum of an event target in comparison with that of the original LSF parameter vector at the corresponding event location. It can be seen from the figure that the event target is different from the original LSF parameter vector. This is because of the iterative refinement of event targets. More interestingly, the event target seems to emphasize the spectrum characteristics of the speaker.

The log power spectra of event targets extracted from MRTD analysis of LSF parameters for a whole Female/Japanese speech utterance are shown in Fig. 6.4 together with the original LSF vectors also provided in the form of log power spectrum.

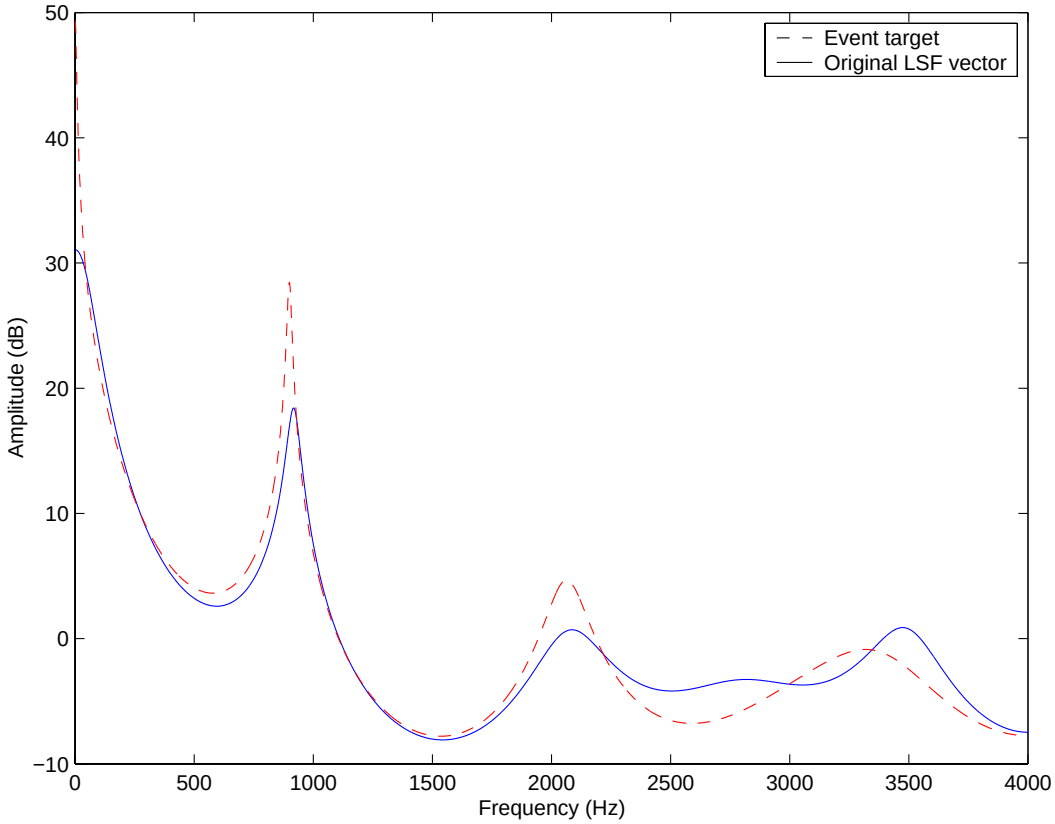


Figure 6.3: Example of an event target obtained from MRTD analysis. The dashed line shows the log power spectra of the event target. The log power spectra of the original LSF parameter vector at the corresponding event location is also provided for reference. Note that the event target is different from the original LSF parameter vector.

6.4 Experimental Results

6.4.1 Database

In the experiments, we used a speaker set of 49 speakers collected from the New England dialect of TIMIT speech corpus [44]. The ratio of male and female speakers is not equal in the set. For each speaker, there are ten sentences. The training set is generated using the eight files with "sx" and "si" prefixes, whereas the two files with "sa" prefix are individually used for testing. Summary of the speaker set is given in Table 6.1.

Table 6.1: Summary of the speaker set.

# Speakers	49 (31 M+18 F)
Avg. duration of training utterance	24.5 sec/speaker
Avg. duration of testing utterance	3.1 sec/sentence

Table 6.2: Total number of feature vectors used in the experiments.

Phase	# MFCC Vectors	# Event Targets
Training	118861	23511
Testing	30231	5929

6.4.2 Preprocessing and Feature Extraction

Prior to any analysis, the speech files were downsampled from 16 to 8 kHz. High emphasis filtering with $H(z) = 1 - 0.95z^{-1}$ was then performed.

To derive event targets used for VQ-based speaker ID, 10^{th} order LSF parameters were calculated first, using a LPC analysis window of 30 ms at 10 ms frame intervals. In the following, the LSF parameters obtained were TD analyzed using the MRTD method. The event rate, i.e. the number of events per second, was set as about 20 events per second, resulting in the number of event targets reduced by a factor of five compared to that of the original LSF vectors.

For comparison, conventional mel-frequency cepstrum coefficients (MFCC) were computed using the 12^{th} short-term mel-cepstrum analysis, also with a 30 ms Hamming window shifted by 10 ms, producing 100 feature vectors per second. The 12 lowest coefficients (excluding the 0^{th} coefficient, which corresponds to the total energy of the frame) were used as alternative features.

Table 6.2 gives the summary of feature vectors used in the experiments. It can be seen from the table that the number of event targets has significantly reduced compared to that of MFCC vectors in both training and testing phases.

6.4.3 Identification Results

A separate classifier was used for each feature set. The distance measure here is the Euclidean distance. The codebooks for each speaker were designed using the LBG algorithm [86]. Speaker ID results for different codebook sizes and the two feature sets are given in Table 6.3. The performance of VQ-based speaker ID on the initiated event targets that consist of the original LSF vectors at event locations is also shown together for reference.

For all codebook sizes listed in the table, the event target features almost show better performance than the other features. This is mainly attributed to the fact that TD can be considered as an effective method of decorrelating the inherent inter-frame correlation present in any frame-based parametric representation of speech. In addition, results also show that the iterative refinement of event targets has positively affected their speaker-specific information.

6.5 Conclusion

The event targets derived from LSF parameters using the temporal decomposition technique were found to be effective when applied in VQ-based speaker identification systems. Their performance is found to be superior to that of the popular MFCC features in the case of testing on clean speech. The number of feature vectors required for both training

Table 6.3: Identification success rates for different codebook sizes and feature sets. Note that LSF features were calculated at the event locations.

Codebook Size	MFCC	Event Targets	Original LSF vectors at event locations
16	89.80%	93.88 %	85.71 %
32	95.92%	95.92 %	93.88 %
64	94.90%	96.94 %	95.92 %
128	95.92%	96.94 %	95.92 %
256	95.92%	97.96 %	93.88 %

and testing phases has been reduced by five times compared to that of the MFCC features, while the identification results obtained are comparable or even better. More interestingly, it was shown that event targets can convey information about the identity of a speaker and the iterative refinement of event targets has positively affected their speaker-specific information. We plan to make future experiments in more demanding environments such as testing on noisy speech, speech at different speaking rates, and cross-language evaluation. The use of event targets as a feature set for speaker verification and other speaker identification systems should also be investigated.

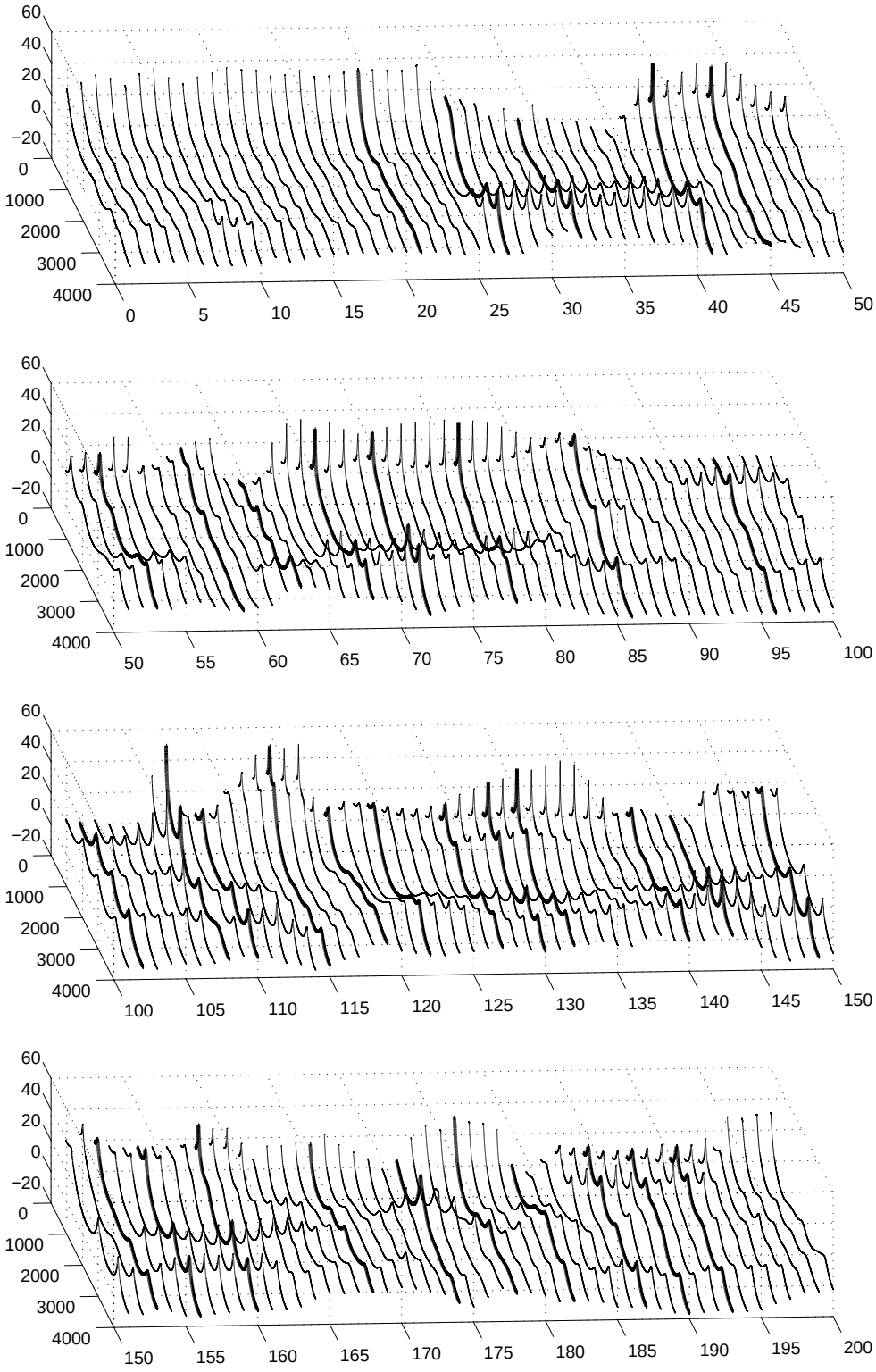


Figure 6.4: Event targets obtained from MRTD for the Female/Japanese speech utterance "*shimekiri ha geNshu desu ka.*" Dark solid lines show the log power spectra of event targets. The log power spectra of the original LSF vectors are also provided.

Chapter 7

Male to Female Voice Transformation

7.1 Introduction

Voice transformation, as defined in this chapter, is the process of transforming one or more features of an input speech signal to new target values [134]. By features we mean fundamental frequency of voicing and formant frequency characteristics. Aspects that are not subject to change should be maintained during the transformation process. The reconstructed signal should also be of high-quality, without artifacts due to the signal processing. This notion of voice transformation is related to, however, distinct from voice conversion where a source speech waveform is modified so that the resulting signal sounds as if it were spoken by a target speaker [1, 80, 97, 105, 132]. A conversion of this type is often done by mapping between detailed features of the two speakers. Our research focuses instead on simultaneous manipulation along the dimensions listed above, in order to achieve interesting transformations of speech and other vocal signals. We will use conversion and transformation interchangeably in the rest of this chapter.

Voice transformation has the potential to solve several problems associated with concatenative speech synthesis, where synthetic speech is constructed by concatenating speech units selected from a large corpus. Such a corpus-based method often succeeds in producing high-quality speech, but the cost of collecting and transcribing a corpus is high, making it impractical to collect data from multiple speakers. To deal with the problem of generating multiple voices, a voice transformation system can be used to generate speech that sounds distinctly different from the voice of the input corpus. The entire corpus can be preprocessed through a transformation phase that can change a female voice into a male-like or childlike voice, while preserving the temporal aspects. In this way, a corpus from a single speaker can thus be leveraged to yield an apparent multiple-speaker synthesis system.

In the past, female and child voices have been somewhat neglected in speech processing. The difficulty in reproducing a female voice is perhaps due to insufficient knowledge of the glottal source characteristics of female speech [124]. For excellent literature reviews pertaining to the problem, see [25, 75]. In addition to complications in higher frequencies which make formant frequencies difficult to estimate, it has also been shown that female voices tend to have a breathy quality. That is to say that the aspiration noise in the vowel spectrum has been found to be greater than for men. Finally, it has also been

hypothesized that female voices do not conform well to an all-pole model. This chapter focuses on the male to female aspect of the problem of voice transformation. However, adult male to child voice transformation can be carried out in the same manner with different scaling factors.

It is generally assumed that the overall shape of the spectral envelope together with the formant characteristics are the major features controlling speaker identity [80]. While an obvious difference between speakers is the variation in the range of fundamental frequency (F0), trying to scale it would give the impression that the same speaker is speaking in a different pitch range. Therefore, it is necessary to modify the spectral information as well: by raising the range of F0 and shortening the vocal tract results in female or child voice, lowering the range of F0 while lengthening the vocal tract will give the impression of an adult male voice. From a transformation point of view, it is convenient to represent the vocal tract system with articulatory parameters [105]. Since articulatory parameters are difficult to extract from the speech signal, at a compromise, formants are proposed for representing the vocal tract system information. In this study we consider the transformation of the average F0 and the formant characteristics only. This approach is somewhat similar to [6].

Research in speech synthesis has led to several voice transformation methods, which are mainly based on TD-PSOLA [100], linear prediction [6], or phase vocoder [134] models. The method we propose here is closely related to the above methods, and can perform general transformation tasks, while preserving excellent speech quality. Our system is based on a very high-quality vocoder, namely, STRAIGHT (stands for "Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum") [67]. For transforming formant characteristics, a newly proposed algorithm for formant modification in the line spectral frequency (LSF) domain [99] is employed.

The voice gender transformation method presented in this chapter can be extended to a general task of voice transformation. As already presented in Section 1.3, temporal decomposition of LSF parameters has desirable properties to be applied in voice modification. Moreover, it was shown in Chapter 6 that the event targets extracted from LSF parameters using the Modified Restricted Temporal Decomposition (MRTD) technique can convey information about the identity of a speaker. Therefore, we have been provided necessary motivation to investigate the potential applications of MRTD in voice conversion, speaker individuality, emotional speech, song synthesis, text-to-speech synthesis, etc. This chapter presents a work-in-progress to prepare for further work towards this end.

The rest of the chapter is organized as follows: Section 7.2 briefly reviews the voice gender differences. Section 7.3 is devoted to a brief description of the algorithm for formant modification in the LSF domain. In Section 7.4, we describe a system for voice transformation and present experimental results. Finally, summary is given and future work is presented in the last section.

7.2 Voice Gender Differences

7.2.1 Physical Differences Relating to Voice Gender

While there is little evidence that human female speech is less intelligible than male speech, coders and synthesizers have not modeled female speech well. The most obvious differences concern the shorter vocal tract and smaller vocal cords of women (with their

yet smaller vocal apparatus, children's speech is even more problematic). This leads to fewer resonances in a given bandwidth (e.g., over the telephone) and to more widely spaced harmonics, with correspondingly less clear definition of formants when using traditional spectral analysis methods. Higher F0 causes greater interaction between the glottal source and the vocal tract than for males. Also, the glottal excitation that women use appears to be more symmetric and breathy (the latter causing more randomness in their speech waveforms) than for men [25]. Women and children have also been somewhat neglected groups in the history of speech analysis by machine. One reason is that most acoustic studies tend to focus on formant frequencies as cues to phonetic contrasts. The higher fundamental frequencies of women and children make it more difficult to estimate formant-frequency locations. Furthermore, informal observations hint at the possibility that vowel spectra obtained from women's voices do not conform as well to an all-pole model, due to perhaps to tracheal coupling and source/tract interaction [75].

A detailed explanation of the speech production mechanism can be found in [124]. A typical male vocal tract is about 17.5 cm in length (i.e. from vocal cords to lips) while that of a female is about 15.2 cm. The adult male larynx is about 1.2 times the size of that of the female. During puberty the male larynx undergoes a change in shape (Adam's apple protrusion) such that the adult male vocal cord membrane length reaches about 1.6 times that of the female. Analysis of male and female voiced utterances shows that the female formant frequencies are about 15 % higher than male. This difference is in close agreement with the male-to-female vocal tract length ratio. Female pitch is generally about 1.7 times that of male. This difference is mainly attributed to the difference in vocal cord membrane length although other factors such as male/female differences in the way in which the cords open and close are believed to be relevant also [135].

We have conducted an experiment to evaluate the differences between male and female Japanese voices in fundamental frequency (F0). The speech data set used for the experiment consists of 10 speakers (5 males & 5 females), each reading 25 phoneme-balanced sentences, of the ATR Japanese speech database [2]. Overall, the F0 values of Japanese females averaged 225 Hz, whereas Japanese males averaged 129.5 Hz. Therefore, the average F0 of Japanese females is approximately 1.74 times higher than that of Japanese males, which comes very close to the 1.7 value in [75].

7.2.2 Voice Gender Perception

For a long time it was believed that pitch was the dominant gender cue. However, recent studies of the speech production mechanism have revealed more subtle differences between features of the male and female speech waveforms. One such study [24] showed that grouped formant information can provide a slightly higher automatic gender distinction success rate (98.1%) than pitch information (96.2%). These figures suggest that both pitch and formant information are important cues in voice gender distinction.

7.3 Formant Modification

When speech signal is analyzed by an all-pole model, a speech formant is usually characterized by a cluster of poles. However, it is difficult to change a formant by shifting pole frequencies which are mainly related to the formant. For example, the second formant

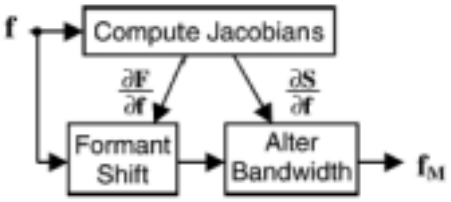


Figure 7.1: Block diagram of the formant modification algorithm.

may incorrectly merge with the first formant. This is called the the pole interaction problem [97]. A method for modifying formant locations and bandwidths directly in the LSF domain has been developed in [99]. This method is based on first-order approximations between different representations of the linear prediction spectrum and does not require finding the roots of the prediction polynomial. In addition, this method is less sensitive to pole interaction problems associated with direct pole modification. It is also computationally efficient, which makes it applicable to real-time applications that require formant modification, especially when LSFs have already been calculated. For these reasons, this method is employed in this work. The algorithm for formant modification in the LSF domain is summarized as follows. The reader is referred to [99] for further details.

The shifting algorithm uses the approximately linear relationship between the LSF frequencies and the formant locations and bandwidths. The block diagram of this algorithm is shown in Fig. 7.1. In this technique, the Jacobian matrices $\partial\mathbf{F}/\partial\mathbf{f}$ and $\partial\mathbf{S}/\partial\mathbf{f}$ are computed, where $\mathbf{F}$ is the vector of formant locations, and $\mathbf{S}$ is a vector containing both the formant locations and bandwidths. Given desired formant and bandwidth shifts, $\mathbf{\Delta F}$ and $\mathbf{\Delta B}$, respectively, the corresponding changes in the LSFs, $\mathbf{\Delta f_F}$ and $\mathbf{\Delta f_B}$, are as follows:

$$\min \|\mathbf{D}\mathbf{\Delta f_F}\|_2, \text{s.t.} \frac{\partial \mathbf{F}}{\partial \mathbf{f}} \mathbf{\Delta f_F} = \mathbf{\Delta F}, \tag{7.1}$$

$$\min \|\mathbf{D}\mathbf{\Delta f_B}\|_2, \text{s.t.} \frac{\partial \mathbf{S}}{\partial \mathbf{f}} \mathbf{\Delta f_B} = \begin{pmatrix} \mathbf{0} \\ \mathbf{\Delta B} \end{pmatrix}, \tag{7.2}$$

where $\mathbf{D}$ is a scaling matrix designed to help preserve the bandwidths in the formant shift. The final LSFs, $\mathbf{f_M}$, are found by summing the shifts,

$$\mathbf{f_M} = \mathbf{f} + \mathbf{\Delta f_F} + \mathbf{\Delta f_B} \tag{7.3}$$

followed by enforcement of LSF ordering to ensure a stable prediction filter.

Fig. 7.2 shows an example of the formant modification algorithm on a spectrum corresponding to the vowel [a] in "ha" of the Male/Japanese speech utterance "*shimekiri ha geNshu desu ka.*"

7.4 Voice Transformation

7.4.1 Method

The human auditory system is highly sensitive to voice perception and synthetic speech, though intelligible, often sounds unnatural. The challenge for voice transformation is to

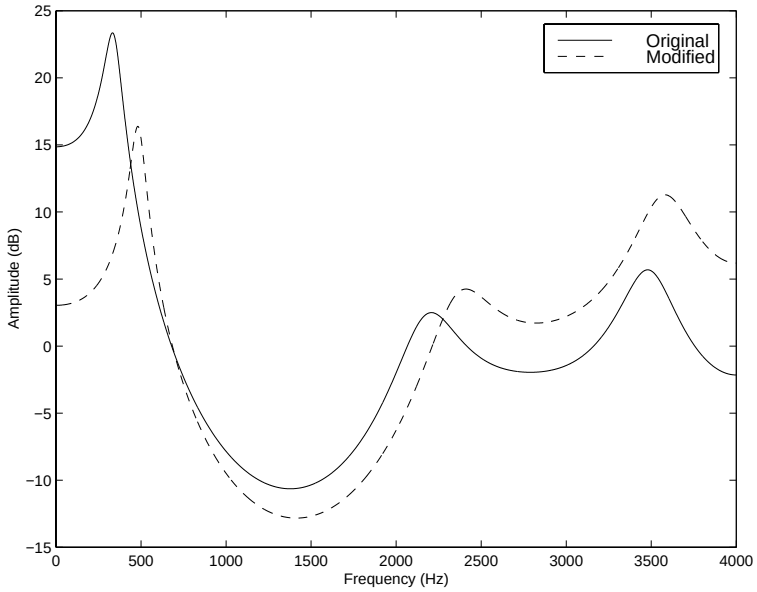


Figure 7.2: Example of the formant modification algorithm on a spectrum: $\Delta F1$=150 Hz, $\Delta F2$=200 Hz, $\Delta F3$=100 Hz.

convert the gender related parameters of the speech signal without affecting naturalness. For a long time it was felt that pitch was the dominant cue in voice gender perception. However, Childers and Wu [24] showed that grouped formant information give a higher automatic gender distinction success rate than pitch information. Hence, realistic voice gender transformation requires independent modification of the glottal (pitch) and vocal tract (formant) related features of the source speech signal. Atal and Hanauer [6] presented a male to female voice transformation algorithm using linear predictive coding analysis to deconvolve the glottal and vocal tract contributions. The authors applied independent scaling factors to the pitch frequency and the formant frequencies and bandwidths prior to resynthesizing to give gender converted speech. The scaling factors were based on typical male-to-female vocal cord membrane and vocal tract length ratios.

Our approach is similar to [6], but has two distinct features. Firstly, we use a very high-quality vocoder, namely, STRAIGHT for analyzing and resynthesizing speech. Secondly, we estimate line spectral frequency (LSF) paramaters from the spectral envelope and employ the newly proposed algorithm for modifying formant characteristics in the LSF domain. The fundamental frequency contour is modified by simply multiplying F0 of voicing with a scaling factor. Other parameters that are not subject to change are maintained during the transformation process. Finally, we resynthesize the converted speech. The block diagram of the proposed method for voice transformation is shown in Fig. 7.3.

7.4.2 Results

We applied the proposed voice transformation algorithm to two sentence utterances, MHTSC213 and MTKSC220, of the ATR Japanese speech corpus [2] spoken by two male speakers, MHT and MTK, respectively. We tried various combinations of pitch and formant scaling factors. For male-to-female the F0 scaling factors, SP, were in the range 1.2 to 2.2 and the formant scaling factors, SF, were in the range 1.10 to 1.20. We found

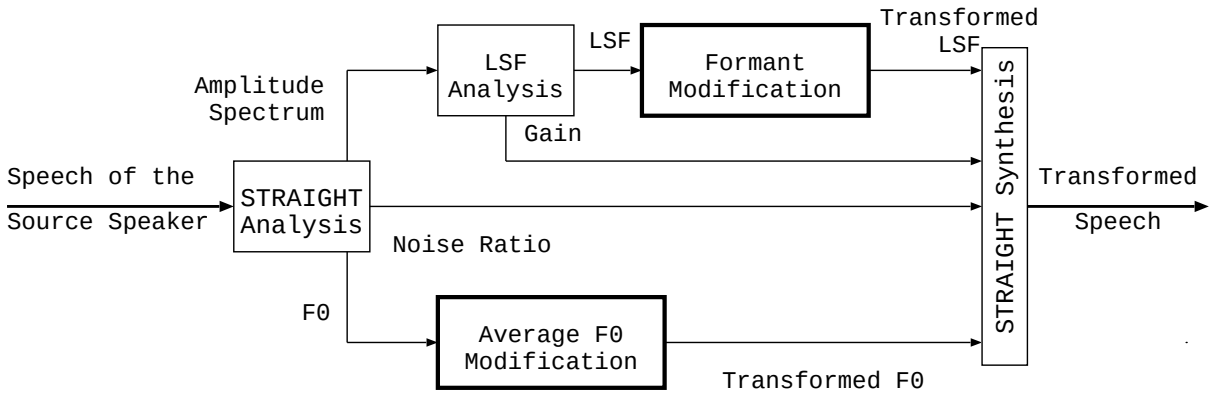


Figure 7.3: Block diagram of the method for voice transformation.

that SP = 1.9 and SF = 1.17 gave the best male to female result. Female-to-male voice transformation can be carried out using the proposed algorithm, but with the reciprocal scaling factors. Male-to-child voice transformation can also be processed by employing this algorithm, but with larger F0 scaling factors as well as formant scaling factors.

Five graduate students known to have normal hearing ability were recruited for the listening experiment. Each listener was asked to listen to each of the stimuli produced by this male-to-female voice transformation method and then make one of the following statements on how the speaker sounds:

- Male and Natural
- Male but Unnatural
- Neither
- Female but Unnatural
- Female and Natural

Most of the subjects regarded the stimuli to as "Female and Natural." This shows that the proposed algorithm can be used to produce natural sounding, high-quality female voices for speech synthesis from a single male speaker. In addition, child voices can also be created from a male voice remaining the high-quality as well as the naturalness. The impressive results are mainly attributed to the use of STRAIGHT, a very high-quality vocoder. This is not the case of previously proposed methods where the speech quality is much degraded during the transformation process. Fig. 7.4 shows an example of speech waveforms and spectrograms of a Male/Japanese sentence utterance before and after transformation. Notice that the formants in the transformed spectrogram are shifted upward.

The female-to-male results were found to be superior to the male-to-female results. These findings are consistent with those of other voice transformation methods [1, 23]. Our scaling factors were very close to those used in [23]. Childers et al. [23] suggested that the reason why male-to-female voice transformation is less successful when a single formant scaling factor is used is that the factor should itself be frequency dependent

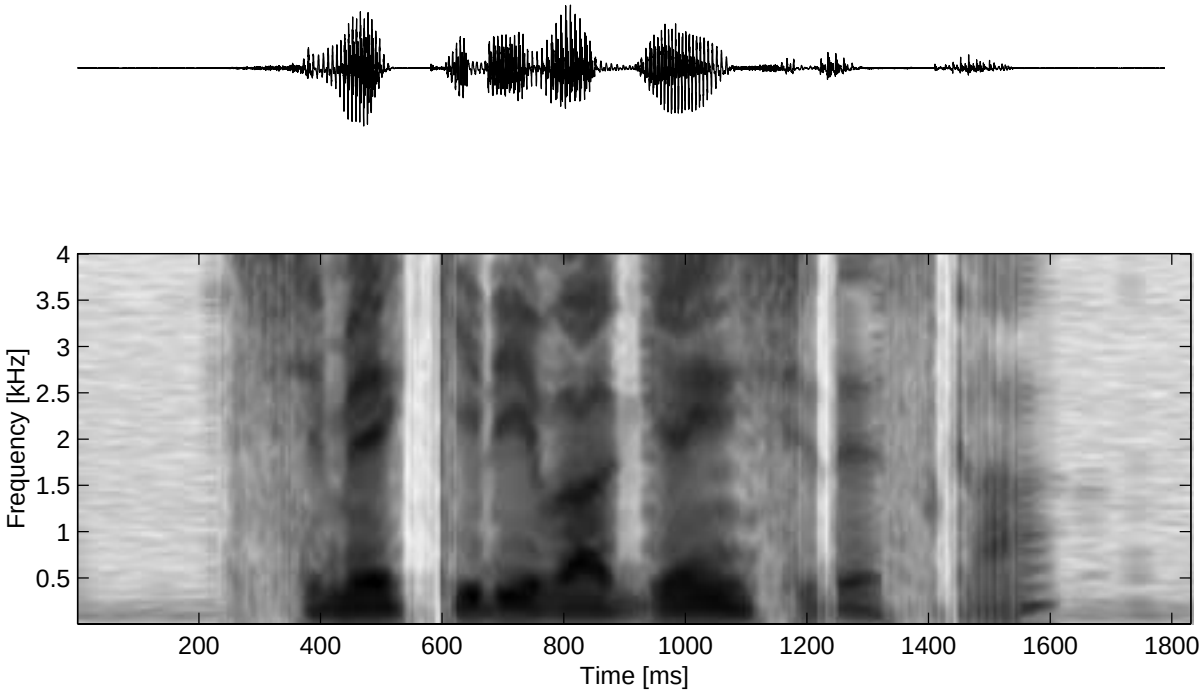
("higher frequency formants should be shifted less than lower frequency formants"). As the single compromise factor is greater than unity for male-to-female and less than unity for female-to-male, the inadequacies of the compromise will be more noticeable on male-to-female.

7.5 Summary and Further Work

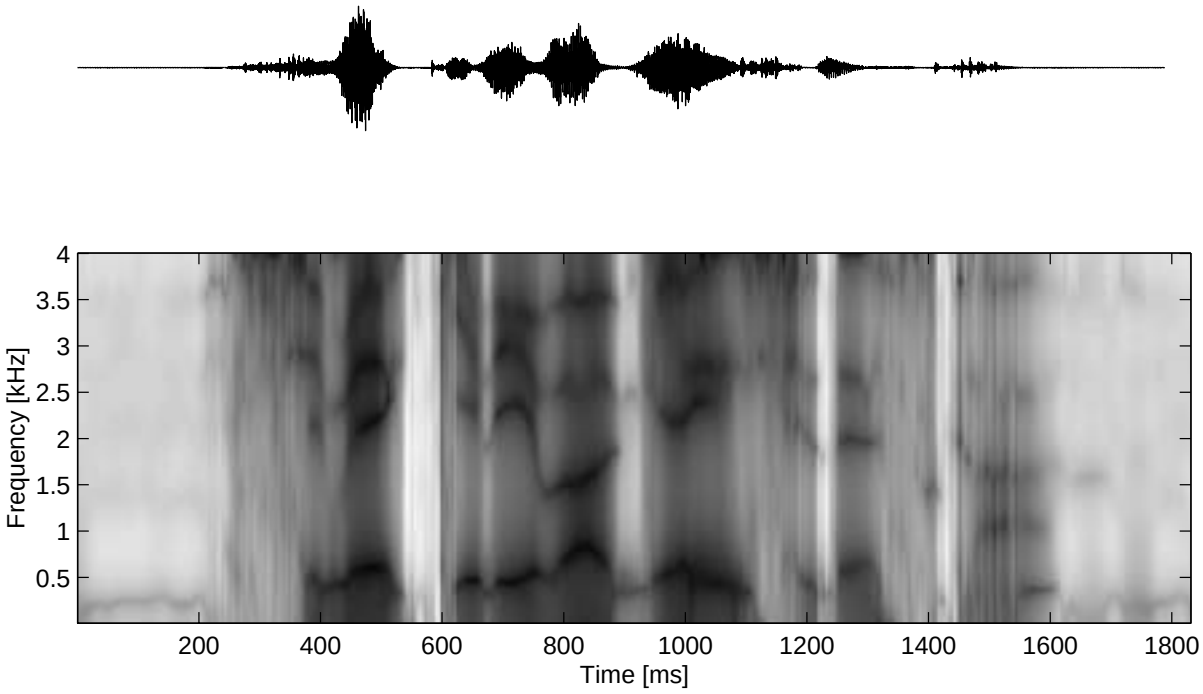
In this chapter, we have presented an efficient voice transformation method based on the modification of formants in the LSF domain. The use of the newly proposed algorithm for modifying formant characteristics in the LSF domain and the implementation of the voice transformation system in the context of STRAIGHT can be highlighted as the main features of the proposed method. A wide range of voices can be produced by using the proposed system. Specially, the system can produce natural sounding, high-quality female and child voices for speech synthesis from a single male speaker.

There are two research directions that should be investigated. The former is motivated by the work on voice conversion reported in [40]. In this research, the speech is first analyzed into LSF parameters, F0s, noise ratio parameters, and gain parameters using STRAIGHT and the procedure described in Section 2.2.2. Then, the author proposed to use the S^{2}BEL-TD algorithm described in Section 3.3 to decompose LSF parameters into events, i.e. event targets and functions. For the source speaker, the whole sentence utterance to be converted is analyzed. Meanwhile, only isolated vowels uttered by the target speaker are taken into account. A mapping rule is established between those events which correspond to each of vowel segments in the source speaker's utterance and the events corresponding to the same vowel, but uttered by the target speaker. In addition, F0s are also S^{2}BEL-TD analyzed into F0 events using the procedure described in Section 3.3.7, and F0 events are exchanged between the two speakers similarly. After exchanging the event targets and F0 events, the LSF parameters are S^{2}BEL-TD synthesized and finally, the speech is synthesized using STRAIGHT. Listening experiments demonstrated that this method is a promising approach to voice conversion, however, the speech quality needs to be improved. One possible reason for the yet high-quality converted speech is that S^{2}BEL-TD sometimes cannot be applied to analyzing LSF parameters. The Modified Restricted Temporal Decomposition (MRTD) method described in Chapter 4, which is capable of decomposing LSF parameters, can be an alternative.

The later research direction is inspired by the method of voice transformation based on the modification of formants in the LSF domain itself. It is known that temporal decomposition (TD) of LSF parameters has some advantages that make it desirable for many applications relating to voice modification. Furthermore, it has been shown in Chapter 6 that the event targets extracted from LSF parameters using the MRTD technique can convey speaker identity. Accordingly, it is worth investigating the possibility of a voice transformation method based on the modification of event targets in some sense. Furthermore, we can control spectral envelopes, durations, and F0 independently and flexibly using the temporal decomposition technique. This gives us necessary motivation to investigate the potential applications of MRTD in speaker individuality, emotional speech, song synthesis, text-to-speech synthesis, etc. Since the voice transformation method presented in this chapter also operates in the LSF domain, it therefore can be used as a baseline for such kinds of applications.



(a) Original speech waveform and spectrogram.



(b) Transformed speech waveform and spectrogram.

Figure 7.4: Speech waveforms and spectrograms of a Male/Japanese sentence utterance "*shimekiri ha geNshu desu ka*" before and after transformation. Notice that the formants in the transformed spectrogram are shifted upward.

Chapter 8

Limited Error Based Event Localizing Temporal Decomposition

8.1 Introduction

Most existing low rate speech coders analyze speech in frames according to a model of speech production. Such a model is the linear predictive coding (LPC) model [88]. However, speech production can be considered as a sequence of overlapping articulatory gestures, each of which producing an acoustic event that should approximate an articulatory target [39]. Due to co-articulation and reduction in fluent speech, a target may not be reached before articulation towards the next phonetic target begins. The non-uniform distribution of these speech events is not exploited in frame-based systems.

The temporal decomposition (TD) method [8] for analyzing the speech signals achieves the objective of decomposing speech into targets and their temporal evolutionary patterns without any recourse to any explicit phonetic knowledge. Considering that speech events do not occur at uniformly spaced time intervals and that articulatory movements are sometimes fast, sometimes slow, Atal [8] concluded that uniform time sampling of speech parameters is not efficient. Thus, he proposed the temporal decomposition method in order to break up the continuous variation of speech parameters into discrete overlapping units of variable lengths located at non-uniformly spaced time intervals.

Despite the fact that TD has the potential to become a versatile tool in speech analysis, its high computational complexity and long algorithmic delay make it impractical for real-time applications. In the original TD method by [8], TD analysis was performed on each speech segment of about 200-300 ms, thus resulting in an algorithmic delay of more than 200 ms. In addition, Atal's method is very computationally costly, which has been mainly attributed to the use of the singular value decomposition (SVD) routine and the iterative refinement process [141]. These prevent Atal's method from real-time applications. The method of TD proposed in [103], Spectral Stability Based Event Localizing Temporal Decomposition (S^2BEL-TD), reduced the computational cost of TD by avoiding the use of SVD, but the long algorithmic delay has more or less remained the same. S^2BEL-TD uses a spectral stability criterion to determine the initial event locations. Meanwhile, the event localization in the Optimized Temporal Decomposition (OTD) method [11] was performed using an optimized approach (dynamic programming). The OTD method can achieve very good results in terms of reconstruction accuracy, but its long algorithmic delay (more than 450 ms) makes it suitable for speech storage related applications only. Also, both

the OTD and S^{2}BEL-TD methods use the line spectral frequency (LSF) parameters [62] as input, which might cause the corresponding LPC synthesis filter to be unstable. The Restricted Temporal Decomposition (RTD) [71] and the modified RTD (MRTD) [107] methods considered the ordering property of LSFs in order to make LSF parameters possible for TD. These methods require an average algorithmic delay of about 95 ms, while can achieve relatively good results.

In this chapter, we propose a new algorithm for temporal decomposition of speech called "Limited Error Based Event Localizing Temporal Decomposition" (LEBEL-TD). This method employs the restricted second order model, i.e. at any instant of time only two event functions can overlap and they sum up to one.

$$\begin{aligned} \hat{\mathbf{y}}(n) &= \mathbf{a}_k \phi_k(n) + \mathbf{a}_{k+1} \phi_{k+1}(n) \\ &= \mathbf{a}_k \phi_k(n) + \mathbf{a}_{k+1} \left(1 - \phi_k(n)\right), \quad n_k \leq n < n_{k+1} \end{aligned} \tag{8.1}$$

where n_k and n_{k+1} are the locations of event k and event $k+1$, respectively. These constraints were geometrically argued to be necessary for describing TD as a breakpoint analysis procedure that yields an approximation of a time sequence of spectral parameters by means of straight line segments.

In addition, the LEBEL-TD method also employs a novel approach to the event localization. Here, the event localization is initially performed based on a limited error criterion, and then further refined by a local optimization strategy. In the following, the event targets are set as the original spectral parameter vectors at the event locations and thus, it can be applied to decomposing the LSF parameters without considering the ordering property of LSFs. This algorithm for TD requires only 75 ms average algorithmic delay[1], while can achieve results comparable to the S^{2}BEL-TD, RTD and MRTD methods. Moreover, LEBEL-TD uses neither the computationally costly SVD routine nor the iterative refinement process, thus resulting in a very low computational cost required for TD analysis.

We have investigated the usefulness of LEBEL-TD in speech coding. In this chapter, a method of variable-rate speech coding based on STRAIGHT (Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum) [67], a very high-quality vocoder, using LEBEL-TD is also presented. For encoding spectral information of speech, LEBEL-TD based vector quantization (VQ) is utilized, whilst other speech parameters are quantized using scalar quantization (SQ). Subjective results indicate that the speech quality of the proposed speech coding method at rates around 1.8 kbps can be comparable to that of the 4.8 kbps US Federal Standard FS-1016 CELP coder.

8.2 LEBEL-TD of Speech Spectral Parameters

8.2.1 Determination of Event Functions

Assume that the locations n_k and n_{k+1} of two consecutive events are known. Then, the right half of the kth event function and left half of the $(k+1)$th event function can be optimally evaluated by using $\mathbf{a}_k = \mathbf{y}(n_k)$ and $\mathbf{a}_{k+1} = \mathbf{y}(n_{k+1})$. The reconstruction error, $E(n)$, for the nth spectral parameter vector is

$$E(n) = \| \mathbf{y}(n) - \hat{\mathbf{y}}(n) \|^2$$

[1] the frame period is 10 ms long, as considered in S^{2}BEL-TD, MRTD, and OTD.

$$= \| (\mathbf{y}(n) - \mathbf{a}_{k+1}) - (\mathbf{a}_k - \mathbf{a}_{k+1})\phi_k(n) \|^2 \quad (8.2)$$

where, $n_k \leq n < n_{k+1}$. Therefore, $\phi_k(n)$ should be determined so that $E(n)$ is minimized.

We suggest using the determination of event functions considered in Chapter 4, where, geometrically speaking, the two event targets $\mathbf{a}_k$ and $\mathbf{a}_{k+1}$ define a plane in the spectral parameter vector space. The determination of event functions is depicted as the projection of vector $\mathbf{y}(n)$ onto this plane. In order to make all event functions well-shaped and model the temporal structure of speech more effectively, the event functions are determined corresponding to the point, $\hat{\mathbf{y}}(n)$, of the line segment between $\hat{\mathbf{y}}(n-1)$ and $\mathbf{a}_{k+1}$ with minimum distance from $\mathbf{y}(n)$. In mathematical form, the above determination of event functions can be written as

$$\phi_k(n) = \begin{cases} 1 - \phi_{k-1}(n), & \text{if } n_{k-1} < n < n_k \\ 1, & \text{if } n = n_k \\ \min(\phi_k(n-1), \max(0, \hat{\phi}_k(n))), & \text{if } n_k < n < n_{k+1} \\ 0, & \text{otherwise} \end{cases} \quad (8.3)$$

where

$$\hat{\phi}_k(n) = \frac{\langle (\mathbf{y}(n) - \mathbf{a}_{k+1}), (\mathbf{a}_k - \mathbf{a}_{k+1}) \rangle}{\| \mathbf{a}_k - \mathbf{a}_{k+1} \|^2}$$

Here, $< ., . >$ and $\| . \|$ denote the inner product of two vectors and the norm of a vector, respectively.

8.2.2 LEBEL-TD Algorithm

The section of spectral parameters, $\mathbf{y}(n)$, where $n_k \leq n < n_{k+1}$, is termed a segment. The total accumulated error, $E_{seg}(n_k, n_{k+1})$, for the segment is

$$E_{seg}(n_k, n_{k+1}) = \sum_{n=n_k}^{n_{k+1}-1} E(n) \quad (8.4)$$

where, $E(n)$ can be calculated for every $n_k \leq n < n_{k+1}$ using Equations (8.2) and (8.3) once n_k and n_{k+1} are known. The buffering technique for LEBEL-TD is depicted in Fig. 8.1, and the whole algorithm is described as follows:

Step 0. Set $k \leftarrow 1$, $n_1 \leftarrow 1$, $a_1 \leftarrow \mathbf{y}(1)$; set n_2 as the last location from n_1 on so that the reconstruction error for every frame in the interval (n_1, n_2) is less than a predetermined number ε.

Step 1. Similarly, set n_3 as the last location from n_2 on so that the reconstruction error for every frame in the interval (n_2, n_3) is less than ε.

Step 2. Local optimize the location of n_2 in the interval (n_1, n_3).

$$n_2^* = \arg \min_{n_1 < n_2 < n_3} \{E_{seg}(n_1, n_2) + E_{seg}(n_2, n_3)\}$$

where, only n_2 that makes $E(n) < \varepsilon$ for every $n_1 < n < n_3$ is taken into account. If n_3 is the last frame, set $k \leftarrow k+1$, $a_k \leftarrow \mathbf{y}(n_2^*)$, $a_{k+1} \leftarrow \mathbf{y}(n_3)$; and exit.

Step 3. Set $k \leftarrow k+1$, $a_k \leftarrow \mathbf{y}(n_2^*)$; then set $n_1 \leftarrow n_2^*$, $n_2 \leftarrow n_3$; and go back to step 1.

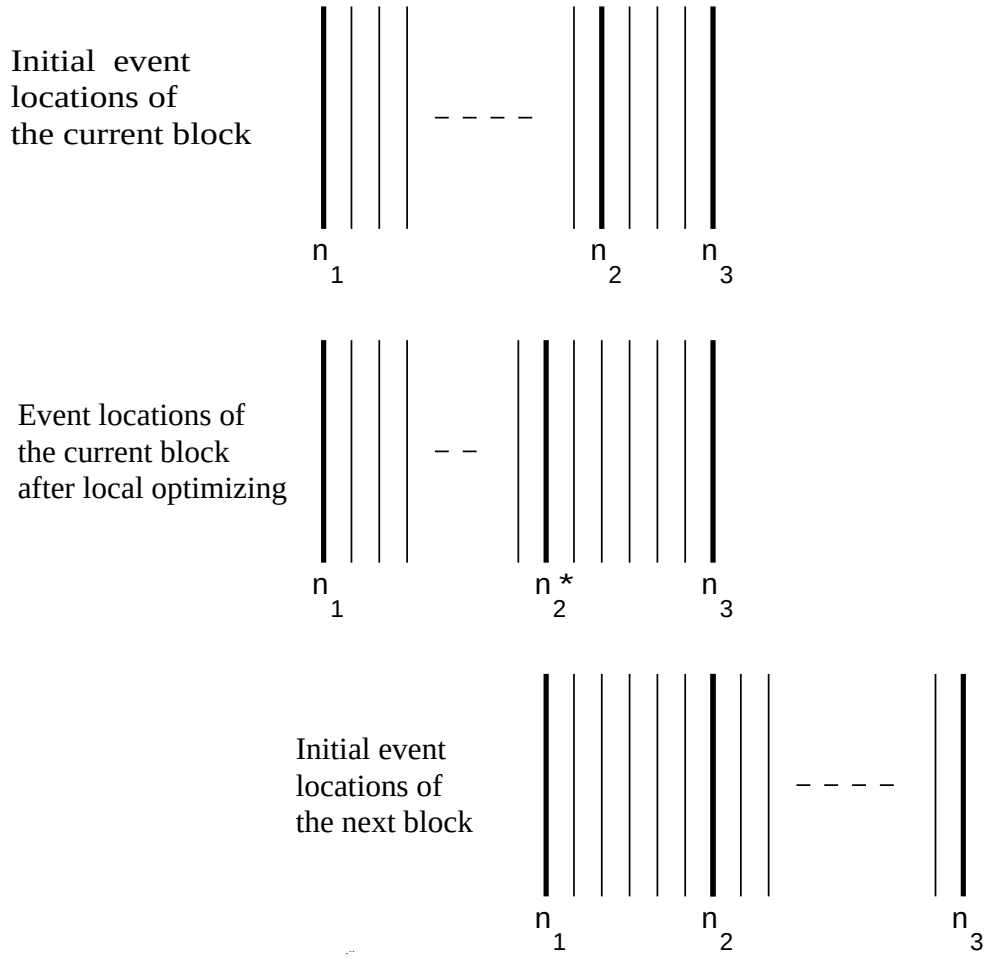


Figure 8.1: Buffering technique for LEBEL-TD

The predetermined number ε is called the reconstruction error threshold, and it is the only parameter that effects the number and locations of the events. It controls the event rate, i.e. the number of events per second, and can be appropriately selected to achieve the optimal performance of LEBEL-TD for different applications.

$\varepsilon = 0.045$ was empirically chosen as a suitable value for the reconstruction error threshold to produce the event rate of about 20 events/sec (note that the spectral parameter here is LSF with the frame period of 10 ms). This event rate results in an average buffering delay of about 50 ms, i.e. 5 frames, along with a 10 ms, i.e. one frame, look-ahead. On the other hand, the LPC analysis window is 30 ms long, which implies a 15 ms look-ahead. Therefore, the average algorithmic delay for LEBEL-TD is about 75 ms and has been known to be the lowest algorithmic delay for TD so far. Moreover, LEBEL-TD has significantly reduced the computational cost of TD because it uses neither the computationally costly SVD routine nor the iterative refinement process. These make LEBEL-TD suitable for real-time applications.

In the LEBEL-TD method, the event targets are set as the spectral parameter vectors corresponding to the event locations. Obviously, the event targets are valid spectral parameter vectors and the stability of the corresponding LPC synthesis filter can be thus ensured after spectral transformation performed by LEBEL-TD. Consequently, LEBEL-

TD can be applied to analyzing any current types of parametric representations of speech. Meanwhile, most conventional TD methods use an iterative refinement of event targets, which might cause the reconstructed spectral parameter vectors to be invalid, for example when TD is applied to decomposing LSF parameters, and thus resulting in an unstable LPC synthesis filter.

Fig. 8.2 shows the plot of event functions obtained from the LEBEL-TD method for an example of a Female/Japanese sentence utterance "*shimekiri ha geNshu desu ka.*" The spectral parameter used here is the LSF. In Fig. 8.3, the plots of original and reconstructed LSF parameters after LEBEL-TD analysis are shown for the same utterance as utilized in Fig. 8.2.

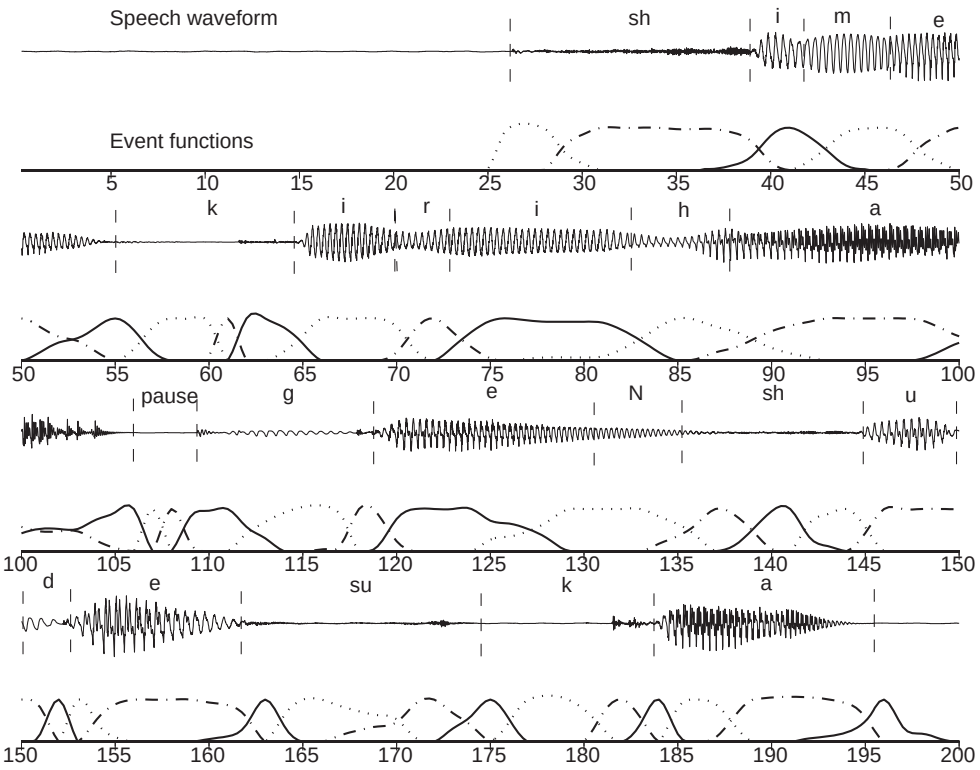


Figure 8.2: Plot of the event functions obtained from the LEBEL-TD method for the Female/Japanese speech utterance *"shimekiri ha geNshu desu ka."* The speech waveform is also shown together with the phonetic transcription for reference. The numerals indicate the frame numbers.

8.3 Performance Evaluation

The ATR Japanese speech database [2] was used for the speech data. Line spectral frequency (LSF) parameters introduced by Itakura [62] have been selected as the spectral parameter for the LEBEL-TD. This is because it is well-known that LSF parameters have the best interpolation [110] and quantization [109] properties over the other LPC related spectral parameters.

The objective performance of TD algorithms can be presented in terms of log spectral

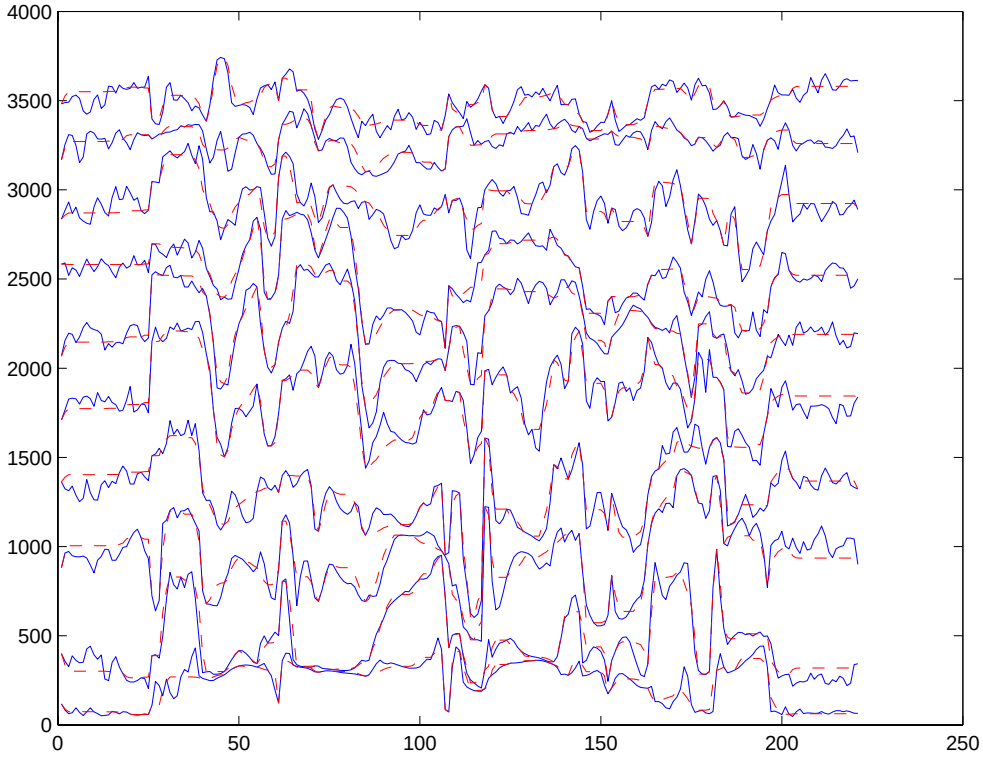


Figure 8.3: Plots of the original and reconstructed LSF parameters obtained from the LEBEL-TD method for the Female/Japanese speech utterance "*shimekiri ha geNshu desu ka.*" The solid line indicates the original LSF parameter vector trajectory and the dashed line indicates the reconstructed LSF parameter vector trajectory. The average log spectral distortion was found to be 1.6276 dB.

distortion (LSD). This is a measure of the distortion between original spectral parameters and the spectral parameters produced by the TD model [11, 127, 103]. The evaluation procedure used in Section 3.3.6 is also recruited to evaluate the interpolation performance of the LEBEL-TD method.

A set of 250 sentence utterances of the ATR Japanese speech database [2] were selected as the speech data. This speech data set consists of about 20 minutes of speech spoken by 10 speakers (5 males & 5 females) re-sampled at 8 kHz sampling frequency. 10th order LSF parameters were calculated using a LPC analysis window of 30 ms at 10 ms frame intervals, and LEBEL-TD analyzed. Additionally, log spectral distortion was also evaluated over the same speech data set for three other methods of TD: S^2BEL-TD [103], RTD [71], and MRTD with LSF as the spectral parameter. The event rate was set as around 20 events/sec for all the four methods.

Table 8.1 gives a comparison of the log spectral distortion results for the LEBEL-TD, S^2BEL-TD, RTD, and MRTD algorithms. The distribution of the log spectral distortion in the form of histograms is shown in Fig. 8.4. Results indicate slightly better performance in the case of S^2BEL-TD over the others, followed by LEBEL-TD and then RTD. However, it has been shown in Chapter 4 that the S^2BEL-TD and RTD methods, in the current forms, cannot always be applied to analyzing LSF parameters due to the stability problems in the LPC synthesis filter. Also, LEBEL-TD requires a lower computational cost for TD

analysis than MRTD, which is mainly attributed to the fact that LEBEL-TD does not employ the iterative refinement process. In addition, LEBEL-TD also needs a shorter algorithmic delay than MRTD.

Table 8.1: Event rate, average LSD, and percentage number of outlier frames obtained from the LEBEL-TD, S^2BEL-TD, RTD and MRTD methods. The spectral parameter is LSF. Speech data set consists of 250 sentence utterances spoken by 10 speakers (5 males & 5 females) of the ATR Japanese speech database.

Method	Event rate	Avg. LSD	2-4 dB	> 4 dB
LEBEL-TD	19.996	1.5125 dB	32.52%	0.07%
S^2BEL-TD	19.455	1.4643 dB	18.48%	0.94%
RTD	20.163	1.5629 dB	22.97%	0.96%
MRTD	20.163	1.5681 dB	23.15%	0.98%

We have also evaluated the performance of LEBEL-TD on the above speech data set for some ε. Table 8.2 gives the summary of LSD and the event rate obtained from LEBEL-TD analysis for some different values of ε. As can be seen from the table, the event rate decreases and the average LSD increases as ε increases. Fig. 8.5 illustrates the average log spectral distortion versus the event rate.

Table 8.2: Event rate, average LSD, and percentage number of outlier frames obtained from the LEBEL-TD method for some ε. The spectral parameter is LSF. Speech data set consists of 250 sentence utterances spoken by 10 speakers (5 males & 5 females) of the ATR Japanese speech database.

ε	Event rate	Avg. LSD	2-4 dB	> 4 dB
0.072	15.059	1.9220 dB	48.52%	1.60%
0.065	16.051	1.8255 dB	45.54%	0.89%
0.058	17.156	1.7270 dB	41.91%	0.46%
0.053	18.107	1.6491 dB	38.69%	0.23%
0.049	18.999	1.5802 dB	35.64%	0.13%
0.045	19.996	1.5125 dB	32.52%	0.07%
0.041	21.124	1.4400 dB	29.02%	0.04%
0.038	22.117	1.3833 dB	26.21%	0.02%
0.0355	23.050	1.3331 dB	23.75%	0.014%
0.033	24.106	1.2795 dB	20.96%	0.01%
0.031	25.028	1.2336 dB	18.69%	0.00%

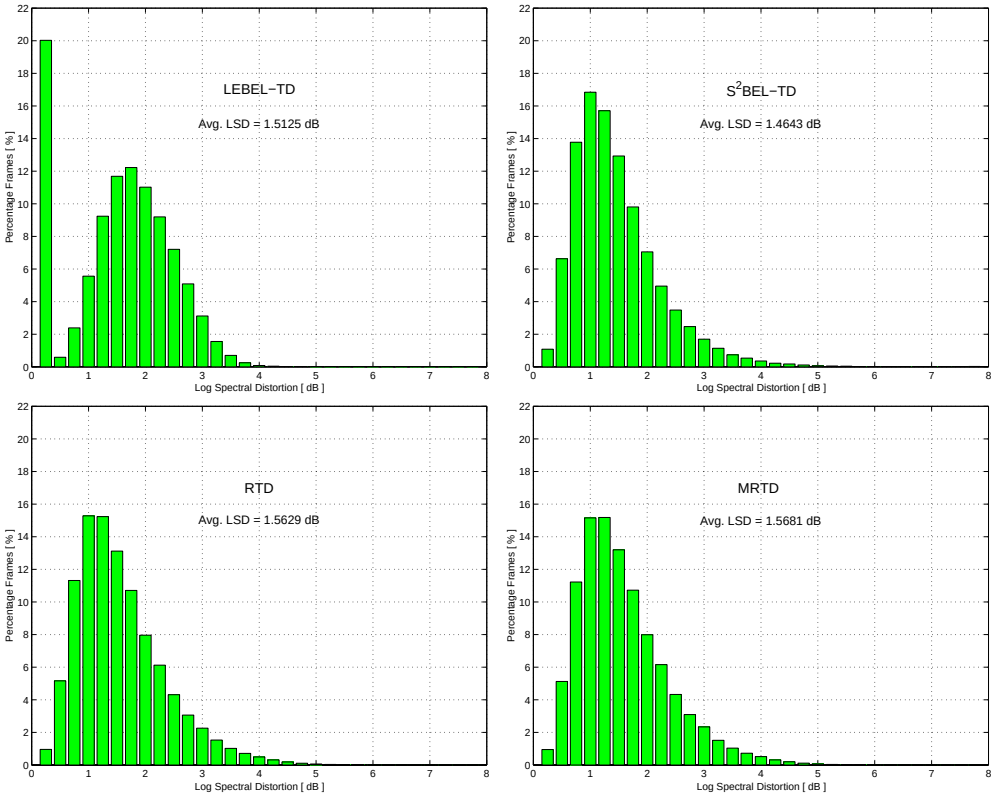


Figure 8.4: Distribution of the log spectral distortion (LSD) between the original and reconstructed LSF parameters in the form of histograms. Top left: LSD histogram for LEBEL-TD. Top right: LSD histogram for S^2BEL-TD. Bottom left: LSD histogram for RTD. Bottom right: LSD histogram for MRTD. Speech data set consists of 250 sentence utterances spoken by 10 speakers (5 males & 5 females) of the ATR Japanese speech database.

8.4 Variable-rate Speech Coding Based on STRAIGHT Using LEBEL-TD

As shown previously, in temporal decomposition (TD) the speech is no longer represented by a vector updated frame by frame, but instead by the continuous trajectory of a vector. The trajectory is decomposed into a set of phoneme-like events, i.e. a series of temporally overlapping event functions and a corresponding series of event targets. Since the event rate varies in time, TD can be considered as a technique for variable-rate speech coding.

As mentioned in Section 2.2.1, STRAIGHT (stands for "Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum") described in [67] is a very high-quality speech analysis-modification-synthesis method. STRAIGHT can decompose a speech waveform into a spectral envelope, i.e. amplitude spectrum, F0 (fundamental frequency) information and noise ratios. Those parameters and the maximum value of amplitude are required for synthesizing speech. Moreover, we can derive LSF parameters from the spectral envelope obtained from STRAIGHT analysis according to the procedure provided in Section 2.2.2.

In this section, we propose a new method of variable-rate speech coding by incor-

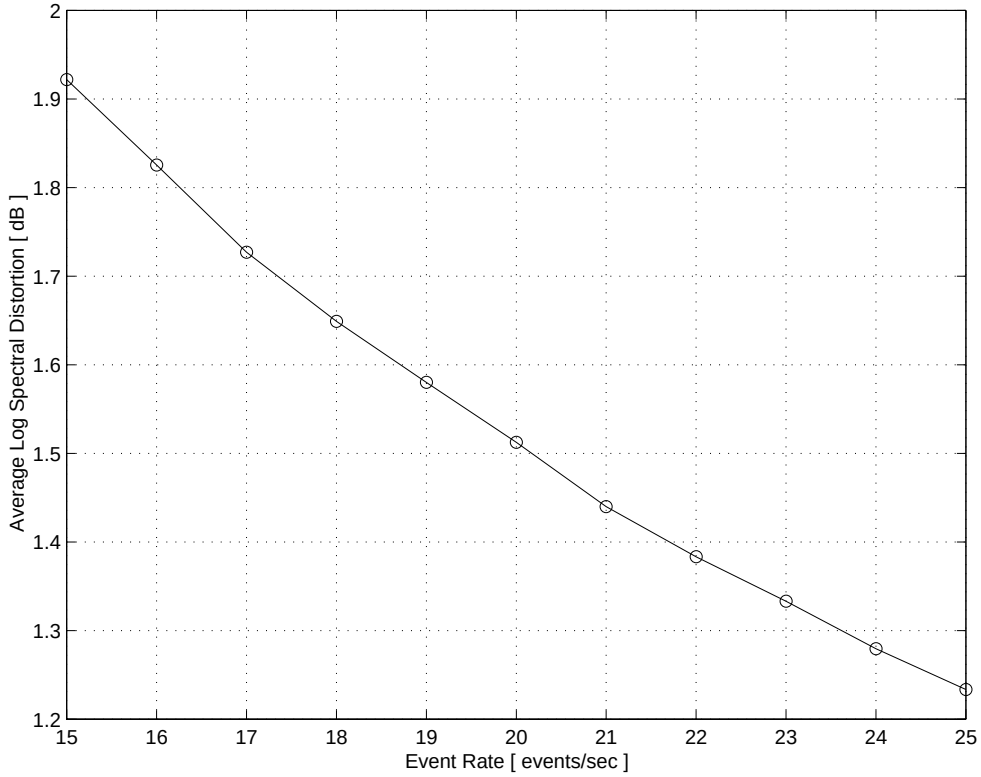


Figure 8.5: Average log spectral distortion (dB) versus the event rate (events/sec).

porating STRAIGHT and LEBEL-TD. The proposed speech encoder and decoder block diagrams are shown in Fig. 8.6, and a detailed description of the proposed speech coding method is shown in the sections followed. Experimental results show that this speech coding method can produce high-quality speech at an average bit rate below 2 kbps.

8.4.1 LEBEL-TD Based Vector Quantization of LSF Parameters

Vector Quantization of Event Targets

Since the event targets obtained from LEBEL-TD method are valid LSF parameter vectors, they can be quantized by usual quantization methods for LSF parameters. Here, the split vector quantization introduced in [109] was adopted. In this work, the order of LSFs was empirically selected as 32 to increase the quality of reconstructed speech. Every event target was divided into four subvectors of dimensions 7, 8, 8, 9 due to the distribution of LSFs, and each subvector was quantized independently. We assigned 8 bits to each subvector, which resulted in 32 bits allocated to one event target.

Vector Quantization of Event Functions

In the case of event functions, normalization of the event functions is necessary to fix the dimension of the event function vector space. Notice that only quantizing $\phi_k(n)$ in the interval $[n_k; n_{k+1}]$ is enough to reconstruct the whole event function $\phi_k(n)$. Moreover, $\phi_k(n)$ always starts from one and goes down to zero in that interval, and the type of

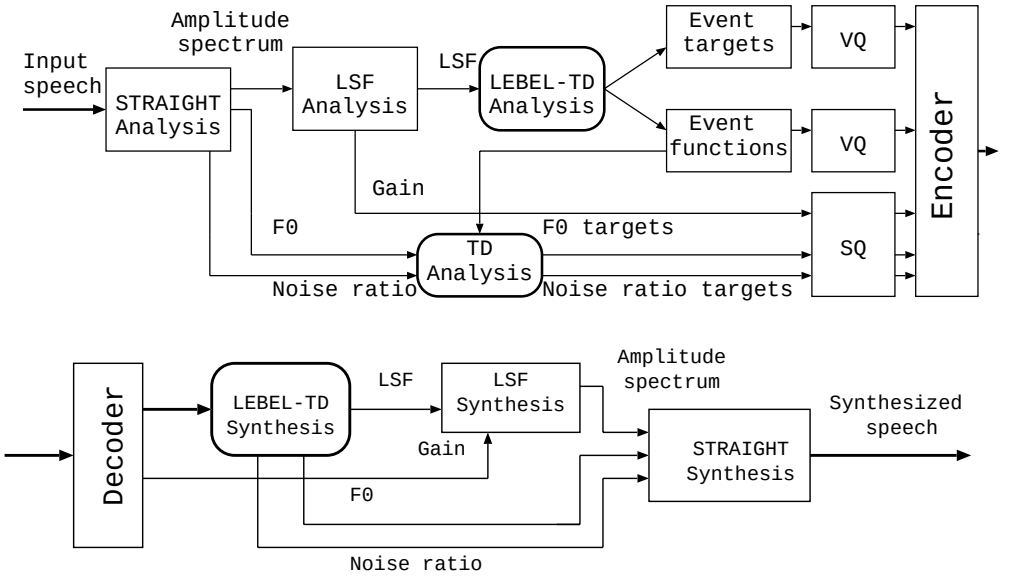


Figure 8.6: Proposed speech encoder and decoder block diagrams (top: encoder, bottom: decoder).

decrease (after normalizing the length of $\phi_k(n)$) can be vector quantized. In this work, 15 equidistant samples were taken from each event function for length-normalization and then vector quantized by a 7-bit codebook. Considering that all intervals between two consecutive event locations are less than 256 frames long (note that the frame period used in STRAIGHT analysis is 1 ms long), we used 8 bits for quantizing the length of each event function. Shortly speaking, each $\phi_k(n)$ was quantized by its length and the type of decrease.

8.4.2 Coding Speech Excitation Parameters

Coding Noise Ratio Parameters

The same event functions obtained from LEBEL-TD analysis of LSF parameters are also used to describe the temporal evolution of the noise ratio parameters. Let $i(n)$ be a noise ratio parameter. We have $0 \leq i(n) \leq 1$, where $i(n) = 1$ for white noise and $i(n) = 0$ for pure pulse. Then $i(n)$ is approximated by $\hat{i}(n)$, the reconstructed noise ratio parameter for the nth frame, as follows in terms of noise ratio targets, i_ks, and the event functions, $\phi_k(n)$s.

$$\hat{i}(n) = \sum_{k=1}^{K} i_k \phi_k(n), \quad 1 \leq n \leq N$$

The noise ratio targets are determined by minimizing the sum squared error, E_i, between the original and the interpolated noise ratio parameters with respect to i_ks.

$$E_i = \sum_{n=1}^{N} \Big(i(n) - \hat{i}(n) \Big)^2 = \sum_{n=1}^{N} \Big(i(n) - \sum_{k=1}^{K} i_k \phi_k(n) \Big)^2$$

where, $i(n)$ is the original noise ratio parameter for the nth frame. Finally, the noise ratio targets are quantized by using scalar quantization. In this work, we used 6 bits for quantizing each noise ratio target.

Fig. 8.7 shows the plots of original and reconstructed noise ratio parameters and the plot of frame-wise noise ratio error, $e_i(n)$, where $e_i(n) = \hat{i}(n) - i(n)$, for a Male/Japanese sentence utterance. The root mean squared (RMS) noise ratio error, $\sqrt{E_i}$, where $E_i = \frac{1}{N}\sum_{n=1}^{N} {e_i}^2(n)$, was found to be about 0.1166.

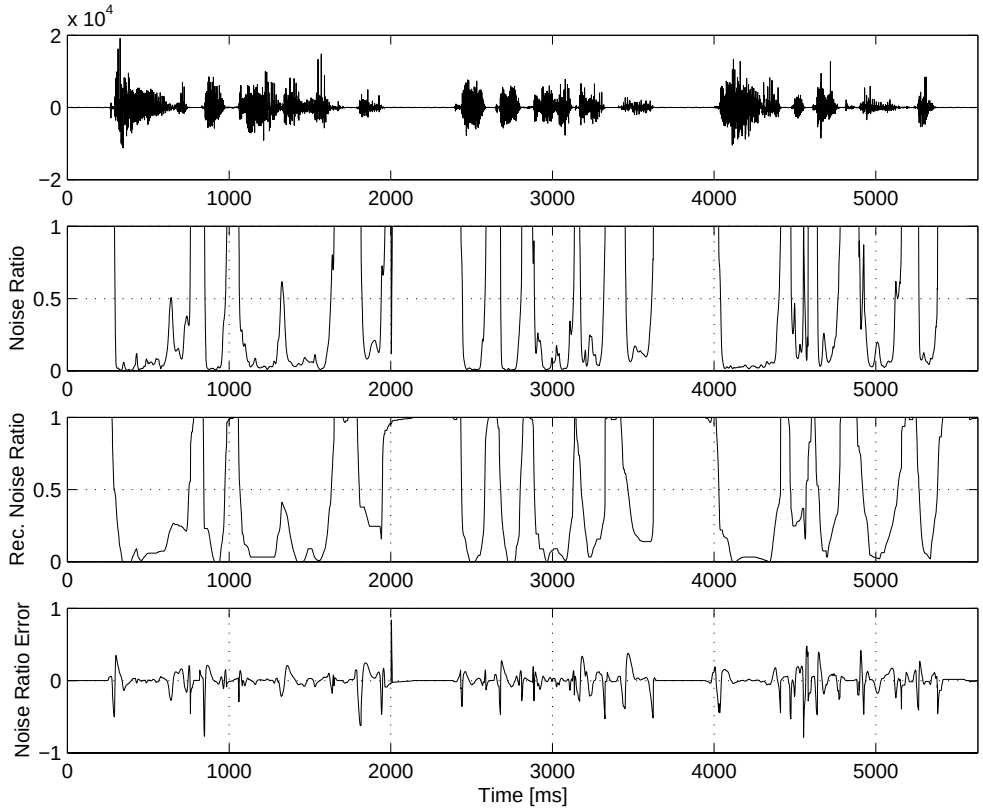


Figure 8.7: Original noise ratio parameters, $i(n)$, reconstructed noise ratio parameters, $\hat{i}(n)$, and frame-wise noise ratio error, $e_i(n) = \hat{i}(n) - i(n)$, for the sentence utterance 'kaigi ni happyou surunodeha nakute choukou surudake dato, hiyou ha ikura kakari masu ka,' of the ATR Japanese speech database. The RMS noise ratio error is 0.1166. The speech waveform is also shown together for reference.

Coding F0 Parameters

For encoding F0 information, the lengths of voiced and unvoiced segments were quantized by scalar quantization first, with an average bit rate of 36 bps. Next, linear interpolation was used within the unvoiced segments to form a continuous F0 contour. The continuous F0 contour was then described by the event functions obtained from LEBEL-TD analysis of LSF parameters and the so-called F0 targets, similarly to that applied to describing the noise ratio parameters presented above. The F0 targets were quantized by a 6-bit logarithmic quantizer. In the decoder, F0 values were reconstructed from the quantized

event functions and F0 targets using the TD synthesis. Meanwhile, F0 values of unvoiced intervals were set to zero.

Fig. 8.8 shows the plots of original and reconstructed F0 parameters and the plot of frame-wise F0 error, $e_p(n)$, where $e_p(n) = \hat{p}(n) - p(n)$ with $p(n)$ and $\hat{p}(n)$ are the original and reconstructed F0 parameters, respectively, for the same sentence utterance as in Fig. 8.7. The RMS F0 error, $\sqrt{E_p}$, where $E_p = \frac{1}{N}\sum_{n=1}^{N} {e_p}^2(n)$, was found to be about 3.6183 Hz.

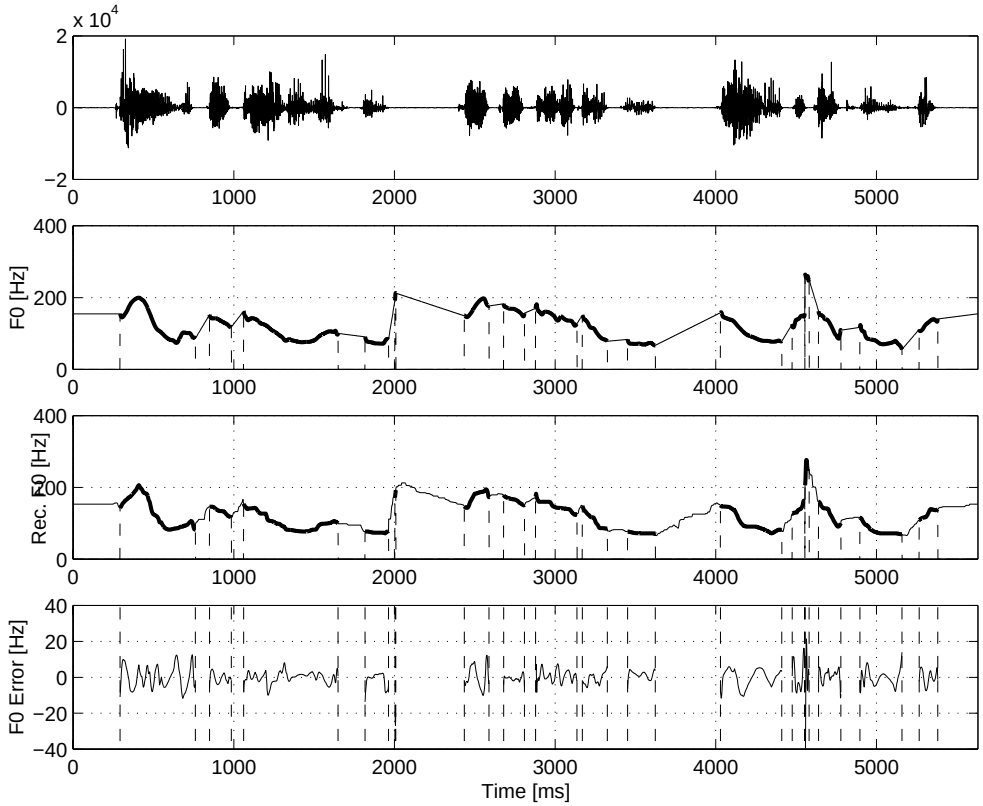


Figure 8.8: Original F0 parameters, $p(n)$, reconstructed F0 parameters, $\hat{p}(n)$, and frame-wise F0 error, $e_p(n) = \hat{p}(n) - p(n)$, for the sentence utterance 'kaigi ni happyou surunodeha nakute choukou surudake dato, hiyou ha ikura kakari masu ka,' of the ATR Japanese speech database. F0 error is shown only for the voiced segments of the utterance. The RMS F0 error is 3.6183 Hz. The speech waveform is also shown together for reference.

Coding Gain Parameters

The gain contour was re-sampled at 20 ms intervals. Logarithmic quantization was performed using 6 bits for each sampled value. The quantized samples and the spline interpolation were used in the decoder to form the reconstructed gain contour.

8.4.3 Bit Allocation

The bit allocation for the proposed speech coding method is shown in Table 8.3. Note that the average number of events per second, i.e. the event rate, was set as 25 events/sec.

Table 8.3: Bit allocation for the proposed speech coder.

Parameter	Proposed Speech Coder
Event target	32 bits (8+8+8+8)
Event function	7 bits
Event location	8 bits
F0 target	6 bits
Noise ratio target	6 bits
Subtotal A (sum × event rate)	1475 bps
Gain	300 bps
Lengths of voiced and unvoiced segments	36 bps
Maximum amplitude of input speech	5 bps
Subtotal B	341 bps
Total (A+B)	1816 bps

8.4.4 Subjective Tests

In order to evaluate the performance of the proposed speech coding method, the quality of the reconstructed speech was compared to that of other low bit rate speech coders such as the 4.8 kbps FS-1016 CELP and 2.4 kbps FS-1015 LPC-10E coders.

A subjective test was carried out using the Scheffe's method of paired comparison [125]. Six graduate students known to have normal hearing ability were recruited for the listening experiment. Each listener was asked to grade from -2 to 2 the degradation perceived in speech quality when comparing the second stimulus to the first, in each pair. The speech data set used in Section 8.3 were selected as the training data for the proposed speech coder. They were re-sampled at 8 kHz sampling frequency and STRAIGHT analyzed using the frame shift of 1 ms. LSF transformation was then performed and the resulting 32nd order LSF parameters were TD analyzed by using the LEBEL-TD method. Two phoneme balanced sentences, which are out of training set, uttered by a male and a female were used as the testing data. Stimuli were synthesized by using the following coders: 4.8 kbps FS-1016 CELP, 2.4 kbps FS-1015 LPC-10E, and the proposed 1.8 kbps speech coder. Also, four other stimuli were STRAIGHT synthesized using the unquantized speech parameters obtained from STRAIGHT analysis & LSF transformation (STRAIGHT-LSF), and from STRAIGHT analysis, LSF transformation & LEBEL-TD analysis (STRAIGHT-LSF & LEBEL-TD).

Results of the listening experiment are shown in Fig. 8.9. It can be seen from this figure that the quality of the reconstructed speech obtained from the proposed speech coder is comparable to that of the 4.8 kbps FS-1016 CELP coder and is much better than that of the 2.4 kbps FS-1015 LPC-10E coder.

8.5 Comparison with the Conventional TD

Although this chapter shows that the LEBEL-TD algorithm is developed from the conventional TD to solve the problem of applying TD to LSF parameters and reducing the algorithmic delay, LEBEL-TD works very similarly to the way of an interpolation method. LEBEL-TD approximates the intermediate samples with only two adjacent event targets as the conventional interpolation method does. However, LEBEL-TD has its uniqueness

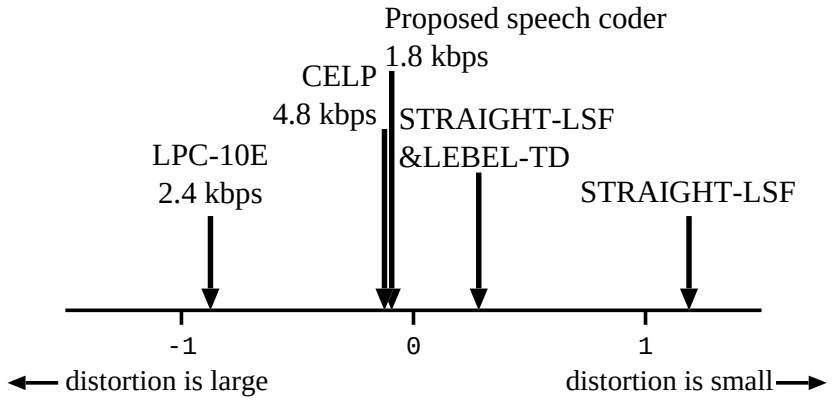


Figure 8.9: Results of the listening experiment.

and differs from both of the conventional TD and interpolation method.

There are two major differences between LEBEL-TD and a primitive linear interpolation method as follows. First, LEBEL-TD uses adaptive updating points, instead of fixed-rate updating ones. It utilizes a limited error based criterion to segment the input speech and a simple local optimization strategy to relocate events more precisely. Consequently, LEBEL-TD possesses the variable-rate sampling effect. Second, LEBEL-TD uses a non-linear interpolation function. The weights for interpolating intermediate samples are determined by minimizing the reconstruction error. Since the interpolation function is not linear, it requires to transmit the weights for reconstruction.

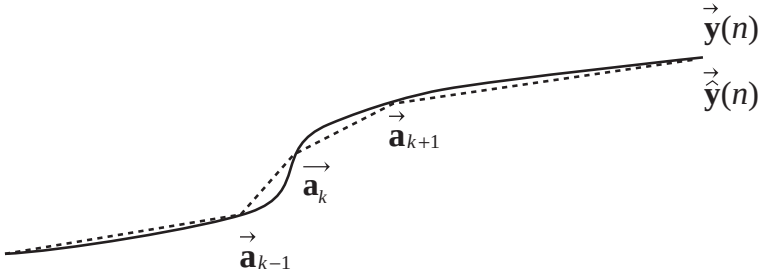


Figure 8.10: The path in parameter space described by the sequence of spectral parameters $\mathbf{y}(n)$ is approximated by means of straight line segments between breakpoints using the LEBEL-TD technique. Note that the breakpoints lie on the path describing the sequence of spectral parameters since the event targets are determined as the original spectral parameter vectors at the event locations , i.e., $\mathbf{a}_k = \mathbf{y}(n_k)$ for every k.

LEBEL-TD employs a restricted second order TD model as MRTD does. In addition, both TD methods use the same determination of event functions given two consecutive event targets. However, LEBEL-TD differs from MRTD and the conventional TD in the following. First, LEBEL-TD utilizes a novel approach to event localization which tries to minimize the overall reconstruction error while keeping the reconstruction error for each frame below a predetermined threshold. This helps to reduce the algorithmic delay down to only 75 ms compared with 95 ms as in MRTD and more than 200 ms as in other algorithms for TD. Moreover, it requires neither the computationally costly singular

value decomposition (SVD) routine as in the conventional TD nor finding spectral stability points as in S^{2}BEL-TD and MRTD, thus resulting in lower computational cost required for locating events. Second, the event targets are determined as the original spectral parameter vectors at the event locations, which is different from the determination of event targets considered in MRTD, as shown in Fig. 8.10. This determination of event targets gives LEBEL-TD several advantages. On one hand, LEBEL-TD can be applied to analyzing any types of spectral parameters, including LSF parameters. On the other hand, it helps LEBEL-TD reduce the computational cost further as the iterative refinement procedure imposed on MRTD and the conventional TD is not required. For those reasons, it can be stated that LEBEL-TD requires lower algorithmic delay and less computational cost compared with other algorithms for TD that have been developed so far.

Unfortunately, the optimization approach utilized in the LEBEL-TD algorithm seems to make this method suitable for speech coding purpose only. It does not give us enough motivation to investigate the usefulness of LEBEL-TD in other application areas of speech processing.

8.6 Conclusion

In this chapter, we have presented a new algorithm for temporal decomposition of speech. The proposed LEBEL-TD method uses the limited error criterion for initially estimating the event locations, and then further refines them using the local optimization strategy. This method achieves results comparable to other TD methods such as S^{2}BEL-TD and MRTD while requiring lower algorithmic delay and less computational cost. Moreover, the buffering technique used for continuous speech analysis has been well developed and the stability of the corresponding LPC synthesis filter after spectral transformation performed by LEBEL-TD has been completely ensured. It is shown that the temporal pattern of the speech excitation parameters can also be well described using the LEBEL-TD technique.

We have also described a method of variable-rate speech coding based on STRAIGHT using LEBEL-TD. For encoding spectral information of speech, LEBEL-TD based vector quantization was used. Other speech parameters were quantized by scalar quantization. As a result, a variable-rate speech coder operating at rates around 1.8 kbps was produced. The quality of the reconstructed speech is comparable to that of the 4.8 kbps FS-1016 CELP coder according to the listening experiment. It can be stated that the proposed speech coding method can produce high-quality speech with less than 2 kbps. The comparison of LEBEL-TD with the conventional TD and interpolation methods suggests its application in speech coding. However, it is still worth investigating whether the optimization approach to TD, i.e. LEBEL-TD, can lead to good results when applied in other application areas, e.g., speech segmentation.

Chapter 9

Conclusion

9.1 Summary of the Thesis

Temporal decomposition (TD) of speech was initiated by Atal [8] as a speech analysis technique for efficient coding of spectral parameters. This method involves the modeling of the spectral parameter trajectory in terms of overlapping units of variable length called events, each of which consists of a target parameter vector (event target), and an associated interpolation function (event function). The event target are assumed to model ideal articulatory configurations of an acoustic event in speech, while the event function describes its temporal evolution. Although TD was initially proposed as a technique for economical coding of speech, subsequent researches have found the potential applications of TD in speech synthesis, speech recognition, and speech segmentation.

The spectral parameter set used by Atal [8] is the log-area parameters. Many other representations of the acoustic speech signal have been used as input for TD. To be possible for TD, the spectral parameters must be valid after spectral transformation performed by TD. For this reason, the line spectral frequency (LSF) parameters have rarely been considered as a candidate spectral parameter set for TD although they have been proved to outperform other parametric representations of speech in terms of both interpolation and quantization. In addition, the LSFs have some properties that make them desirable for voice modification. Temporal decomposition of LSF parameters would have a wide range of applications including very low-bit-rate speech coding, voice transformation, emotional speech, etc. Accordingly, the main objective of this thesis research is to propose efficient algorithms for TD of LSF parameters.

On the other hand, for the use in real-time applications, it is crucial to have a method of TD with short algorithmic delay and low computational cost. However, most algorithms for TD method require more than 200 ms buffering delay which is not suitable for such kinds of applications. Moreover, they are very computationally costly, which has been mainly attributed to the use of the singular value decomposition (SVD) routine and the Gauss-Seidel iterations. Therefore, this thesis also concentrates on the problem of reducing the algorithmic delay and computational cost required for TD analysis.

On the application side, the major application of TD is in speech coding, where producing high-quality speech at rates below 2.4 kbps is still a challenging issue. Working towards this end, we have presented two methods of speech coding using the proposed TD algorithms. Also, the application of TD in some other application areas such as speaker recognition and voice transformation has been investigated and directed in the thesis.

In **Chapter 2**, this thesis first provided an overview of the temporal decomposition technique and its derivations. For the sake of convenience, prior to that, **Chapter 2** outlined the linear predictive coding (LPC) analysis that is now popular in most speech coders. Some alternative parametric representations of LPC coefficients were presented with emphasis on the advantages of the line spectral frequencies (LSFs). This chapter also briefly described modified algorithms for TD and pointed out the problems occurred while performing TD analysis of LSF parameters.

Chapter 3 then described three methods of temporal decomposition that have a direct impact on this thesis investigation: the original TD method (Atal's method) [8], the Spectral Stability Based Event Localizing Temporal Decomposition (S^2BEL-TD) method [103], and the Restricted Temporal Decomposition (RTD) method for LSF parameters [71]. Additionally, the advantages and shortcomings of these TD algorithms were also discussed in the chapter.

The computational procedure of Atal's method includes first expressing the event functions as a linear combination of orthogonal functions using singular value decomposition (SVD) of the spectral parameter matrix. Then, the event functions are computed based on the minimization of a compactness measure of event functions. The event targets are then calculated by minimizing the mean squared error between the original and reconstructed spectral parameters. Finally, an iterative refinement procedure is adopted to refine the event functions and event targets, further improving the reconstruction accuracy of the TD model.

In S^2BEL-TD, the event localization is performed based on a maximum spectral stability criterion. This overcomes the high parameter sensitivity of events of Atal's method. Also, S^2BEL-TD avoids the use of the time consuming singular value decomposition routine imposed on Atal's method, thus resulting in a computationally simpler algorithm for TD. Simulation results showed that an average log spectral distortion of about 1.5 dB can be achieved with line spectral frequencies as the spectral parameter. It was shown that the temporal pattern of the speech excitation parameters can also be well described using the S^2BEL-TD technique. A well-defined evaluation procedure was added to confirm the efficiency of the S^2BEL-TD method.

The RTD method was presented for LSF parameters. This method intends to achieve this end by preserving the LSF ordering property of the event targets at the cost of sacrificing an acceptable spectral distortion. Additionally, RTD employs a restricted second order TD model, in which only two adjacent event functions can overlap and all event functions at any instant of time sum up to one. The RTD method can achieve results comparable to S^2BEL-TD in terms of reconstruction accuracy with lower computational complexity and higher stability. In this method, initial approximations of event targets are determined corresponding to the original LSF parameter vectors at the event locations, where these locations are detected a priori as local minimal points of a spectral transition measure, similarly to that in S^2BEL-TD.

Chapter 4 focused on an algorithm for temporal decomposition (TD) of LSF parameters, called "Modified Restricted Temporal Decomposition" (MRTD), which was derived from the RTD method. LSF parameters have not been used as input for TD due to the stability problems in the LPC synthesis filter. As mentioned above, the RTD algorithm intends to make LSF parameters possible for TD by trying to preserve the LSF ordering property of the event targets, however, RTD still has not completely ensured this property for them. In addition, event functions obtained from RTD analysis may be ill-shaped,

i.e. some of them may have more than one peak, which is undesirable from speech coding point of view. Basing on the geometric interpretation of the TD results, the MRTD algorithm imposes a new constraint on the event functions so that the drawbacks of RTD in terms of ill-shaped event functions have been overcome. In addition, it uses an improved procedure to preserve the LSF ordering property of event targets so that the stability of the corresponding LPC synthesis filter after spectral transformation performed by MRTD has been completely ensured. Similar to RTD, the MRTD method employs the restricted second order TD model. Also, this method uses a spectral stability criterion considered in the S^2BEL-TD method for event localizing. It was shown that spectral information of speech can be efficiently encoded using MRTD based vector quantization. Moreover, excitation information of speech can also be well described and quantized using the MRTD technique.

In **Chapter 5**, a method of very low-bit-rate speech coding using MRTD was developed. This speech coder was implemented in the context of STRAIGHT (stands for "Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum") which is a very high-quality speech analysis-modification-synthesis method [67]. For encoding spectral information of speech, MRTD based vector quantization was used. Meanwhile, pitch and gain parameters were encoded using linear and spline interpolation, respectively. In this chapter, listening experiments were conducted and it was shown that the proposed speech coding method can give the speech quality close to that of the 4.8 kbps FS-1016 CELP coder at the rates of around 1.2 kbps.

In **Chapter 6**, the event targets extracted from LSF parameters using the MRTD technique were found to be effective when used in VQ-based speaker identification systems as a set of features. The number of feature vectors required for both training and testing phases has been reduced by five times compared to that of the traditional mel-frequency cepstrum coefficients (MFCC) features, while the identification results obtained are comparable or even better. More interestingly, this resulted in the fact that the event targets in TD can convey information about the identity of a speaker and the iterative refinement of event targets has positively affected their speaker-specific information. However, the identification performance was evaluated on clean speech only.

Chapter 7 presented a pilot study of voice gender transformation. It is known that TD of LSF parameters has some merits that give a hint for investigating its usefulness in many application areas such as voice transformation, voice conversion, speaker individuality, emotional speech, song synthesis, text-to-speech synthesis, etc. To prepare for future research towards this end, a voice gender transformation method based on modification of formants in the LSF domain has been developed. Voice transformation, as considered in **Chapter 7**, is the process of transforming one or more features of an input speech signal to new target values. By features we mean fundamental frequency of voicing and formant frequency characteristics. The proposed method is based on the use of a formant modification algorithm in the LSF domain. This transformation method was implemented in the context of STRAIGHT system, which allows high-quality modifications of speech signals. Evaluation by informal listening tests showed that the proposed transform can achieve the converted speech of high quality and high naturalness. Moreover, this method can be extended to a general task of voice transformation. It has been expected that TD of LSF parameters can be incorporated with this voice transformation algorithm to be used in the applications relating to voice modification as mentioned above.

Chapter 8 proposed a low-delay algorithm for temporal decomposition (TD) of

speech, called "Limited Error Based Event Localizing Temporal Decomposition" (LEBEL-TD). In previous work with TD, TD analysis was usually performed on each speech segment of about 200-300 ms or more, making it impractical for real-time applications. In this present work, the event localization is determined based on a limited error criterion and a local optimization strategy, which results in an average algorithmic delay of 75 ms. Simulation results showed that an average log spectral distortion of about 1.5 dB can be achievable at an event rate of 20 events/sec. Also, LEBEL-TD uses neither the computationally costly singular value decomposition routine nor the event refinement process, thus reducing significantly the computational cost of TD. Furthermore, a method of variable-rate speech coding at average rates around 1.8 kbps based on STRAIGHT using LEBEL-TD was also presented in the chapter. Subjective test results indicated that the speech quality obtained from the proposed speech coding method is comparable to that of the 4.8 kbps FS-1016 CELP coder. Also, the differences between LEBEL-TD and the conventional TD as well as the conventional interpolation method were pointed out.

In summary, we have proposed two efficient algorithms for TD of LSF parameters, MRTD and LEBEL-TD, and investigated their applications in speech coding and speaker recognition. But it is more than that, their geometric interpretation as an effective breakpoint analysis procedures gives a means of speech segmentation and speech recognition. The fact that the event targets extracted by MRTD can convey speaker identity and the localized nature of each LSF provide necessary motivation to investigate the application of MRTD in voice conversion. More interestingly, using MRTD we can control spectral envelopes, durations, and fundamental frequencies independently and flexibly, which suggests its potential applications in emotional speech, song synthesis, text-to-speech synthesis, etc. To prepare for future research towards this end, we have developed a voice transformation system based on the modification of formants in the LSF domain.

Additionally, during this thesis investigation, we have also studied the S^{2}BEL-TD algorithm and proposed improvements to this method. In **Appendix A**, a mathematical proof of the convergence property of the iterative refinement procedure in the S^{2}BEL-TD was presented. Also, some modifications to the original S^{2}BEL-TD method which help accelerating the convergence of the iterative refinement process were introduced. Experimental results showed that the modified S^{2}BEL-TD gave a slightly better performance in terms of log spectral distortion over the original one while requiring fewer iterations.

9.2 Further Research Directions

As methods for temporal decomposition of LSF parameters are relatively new, only a few aspects of the methods have been explored. Many research topics still remain for further investigation and research. In the following, some suggestions will be given.

1. **Improvement of TD algorithms:** The objectives are to increase the reconstruction accuracy of TD modeling and, at the same time, to reduce the algorithmic delay for the use in real-time applications. It is expected to have such a TD method by using the dynamic programming based optimization strategy as in [11], but with some restrictions to set up an upper threshold for the buffering delay. Shortly speaking, this method is to be developed based on a compromise between the reconstruction accuracy and the algorithmic delay.

2. **Further work on speaker recognition:** The event targets extracted from LSF parameters using the MRTD technique were found to be effective when applied in VQ-based speaker identification systems as a set of feature vectors. However, the identification performance was evaluated on clean speech only. The use of event targets as a feature set in speaker verification and other speaker identification systems should also be investigated. Additionally, further experiments should be made in more demanding environments, such as noisy speech, speech at different speaking rates, speech with emotion, and cross-language evaluation.

3. **Development of a voice conversion system:** A complete voice conversion system using TD of LSF parameters should be improved and further developed. The event targets were found to convey information about the identity of a speaker. It is of interest to investigate whether the exchange of event and excitation targets in some way would lead to a method of voice conversion.

4. **Speech segmentation:** The two proposed TD methods, MRTD and LEBEL-TD, should be experimentally proved to give better performance in terms of segmental relevance. This is because both MRTD and LEBEL-TD methods make use of a new determination of event functions that was claimed to describe the temporal structure of speech more effectively in the interpretation of TD as a breakpoint analysis procedure. This fact, however, should be verified by experiments.

5. **Speech recognition:** There have been quite a few studies focusing on the application of temporal decomposition in speech recognition. Bimbot et al. [17] reported the results of a preliminary recognition experiment on a small corpus of continuously spelled French surnames. In the training phase, event targets are automatically extracted and manually labelled. In the recognition phase, a lattice of the three best candidate phonemes is obtained and searched through, taking into account the lexical constraints of the French alphabet. They claimed a recognition score of 70% on the letter level. Although this identification score may not seem very high, it should be noted that unlike many other recognition approaches, the number of words to be recognized is not restricted. Niranjan and Fallside [106] suggested connecting temporal decomposition with Hidden Markov Modeling (HMM). Kim [73] built a simple word recognition utilizing the coded speech input. It is of interest to investigate whether MRTD and LEBEL-TD can be effectively applied in speech recognition.

6. **Incorporation of TD and ICA:** The formulation of temporal decomposition is very similar to that of the Independent Component Analysis (ICA). It would certainly be interesting to investigate whether a combination of these two techniques would lead to better results.

7. **Others:** Temporal decomposition of LSF parameters has many advantages to be useful in voice transformation, speaker individuality, emotional speech, song synthesis, text-to-speech synthesis, etc. It would be interesting to investigate the usefulness of MRTD and LEBEL-TD in these applications.

Appendix A

Convergence Property of the Iterative Refinement Procedure in the S^{2}BEL-TD Method

The original method of temporal decomposition (TD) proposed by Atal [8] is known to have the two major drawbacks of high computational cost, and the high parameter sensitivity of the number and locations of events. Spectral Stability Based Event Localizing Temporal Decomposition (S^{2}BEL-TD) [103] has been proposed to overcome these drawbacks of Atal's method. To achieve this end, S^{2}BEL-TD implements TD in a mathematically simpler way, i.e. by avoiding singular value decomposition (SVD) routine, while adopting a maximum spectral stability criterion to determine the number and locations of the events, which avoids the necessity of redundant evaluation of event functions.

As already described in Section 3.3, S^{2}BEL-TD determines the event targets, $\mathbf{a}_k$, and event functions, $\phi_k(n)$, once the spectral parameters, $\mathbf{y}(n)$, of a speech segment are given. This method is based on an assumption that each acoustic event that exists in speech gives rise to a spectrally stable point in its neighborhood. Therefore, the locations of the spectrally stable points and the corresponding spectral parameter vectors can be used as a good approximation to the event locations and event targets, respectively. Given these locations, the subsequent computation of refined event targets and event functions is much less demanding than the traditional TD method. Also, this makes the number and locations of the events more parameter independent. Following the first approximation of event targets, an iterative refinement procedure is required to shape up the event functions and to refine the event targets. This procedure aims at improving the reconstruction accuracy of TD results.

In S^{2}BEL-TD, however, the convergence property of the iterative refinement procedure has not been mathematically established. This appendix proposes a new criterion for the termination of iterations as well as a mathematical proof for the convergence of iterations in that procedure. Also, some modifications are made to the original S^{2}BEL-TD method to improve the robustness of its iterative refinement procedure in this respect. Experimental results confirm that the S^{2}BEL-TD method can work well with these modifications.

A.1 Iterative Refinement Procedure in S^{2}BEL-TD

As mentioned earlier, 4 to 5 iterations in general are required for the iterative refinement procedure described in Section 3.3.3 to shape up the event functions. However, this is merely empirical and there is no evidence that the procedure can be terminated. In other words, we cannot ensure the convergence property for the iterative refinement procedure adopted in the S^{2}BEL-TD. In this section, we propose to make some modifications to the original S^{2}BEL-TD method as follows. Firstly, the order of the refinement process is subject to change, i.e., the refinement of event targets is carried out before that of event functions. Secondly, the minor-lobes of those event functions which are considered for the recalculation of event targets are truncated before use in order to accelerate the convergence. Lastly, a new termination criterion is established. With these modifications, a mathematical proof for the convergence property of the iterative refinement procedure adopted in S^{2}BEL-TD has been realized. It is shown that the performance of the modified method is comparable to that of the original one while requiring fewer iterations.

A.1.1 Refinement of Event Targets

Refinement of event targets involves the recalculation of them by minimizing the sum squared error between the original and the reconstructed spectral parameters, with respect to the target vectors. Event targets at the lth iteration are calculated from the event functions at the lth iteration, as described below.

$$\mathbf{\Phi}^{(l)} \rightarrow \mathbf{A}^{(l)}, \quad 1 \leq l \leq S$$

$$E_i^{(l)} = \sum_{n=1}^{N} \left(y_i(n) - \sum_{k=1}^{K} a_{ik}^{(l)} \phi_k^{(l)}(n) \right)^2, \quad 1 \leq i \leq P$$

where the minor-lobes of event functions were truncated before use.

It is worth noting that the initial approximation of the event target matrix is denoted as $\mathbf{A}^{(0)}$ and the first approximation of the event function matrix is denoted as $\mathbf{\Phi}^{(1)}$. By setting the partial derivative of $E_i^{(l)}$ with respect to a_{ir}, to zero:

$$\sum_{k=1}^{K} a_{ik}^{(l)} \sum_{n=1}^{N} \phi_k^{(l)}(n) \phi_r^{(l)}(n) = \sum_{n=1}^{N} y_i(n) \phi_r^{(l)}(n) \tag{A.1}$$

where $1 \leq r \leq K$, $1 \leq i \leq P$. Equation (A.1) gives P sets of K variable simultaneous equations, using which $a_{ik}^{(l)}$, where $1 \leq k \leq K$ and $1 \leq i \leq P$, could be evaluated. Therefore, the event targets matrix at the iteration step l can be formed as follows:

$$\mathbf{A}^{(l)} = \left[\; a_{ik}^{(l)} \; \right]_{1 \leq i \leq P, \;\; 1 \leq k \leq K}$$

In matrix form, Equation (A.1) can be written as

$$\mathbf{A}^{(l)} = \mathbf{Y} \mathbf{\Phi}^{(l)\#} \tag{A.2}$$

where, $\mathbf{\Phi}^{(l)\#} = \mathbf{\Phi}^{(l)T} \left(\mathbf{\Phi}^{(l)} \mathbf{\Phi}^{(l)T} \right)^{-1}$

The matrix of event targets at the lth iteration, $\mathbf{A}^{(l)}$, can also be calculated using Equation (A.2) alternatively.

A.1.2 Refinement of Event Functions

The functional $J\Big(\boldsymbol{\phi}^{(l+1)}(n), \lambda^{(l)}\Big)$ is formulated by taking into account the sum squared error between the original and reconstructed spectral parameters at the lth iteration, and a constraint to limit the spreading of event functions in time, as given in Equation (A.3).

$$J\left(\boldsymbol{\phi}^{(l+1)}(n), \lambda^{(l)}\right) = \sum_{i=1}^{P}\left(y_i(n) - \hat{y_i}^{(l)}(n)\right)^2 + \lambda^{(l)}\sum_{k=1}^{K} w_k^{(l)}(n)^2 \phi_k^{(l)}(n)^2 \tag{A.3}$$

The event functions at the lth iteration are calculated using the procedure described in Section 3.3.2, but use an adaptive weighting function and the quantitative balancing of the two error-terms of the functional $J\Big(\phi^{(l+1)}(n), \lambda^{(l)}\Big)$, as given below.

$$\left(\mathbf{A}^{(l)}, \boldsymbol{\Phi}^{(l)}\right) \rightarrow \boldsymbol{\Phi}^{(l+1)}, \quad 1 \leq l \leq S$$

where l and S are the iteration step number and total number of iterations, respectively.

Adaptive weighting function: An adaptive weighting function is defined as given in Equation (A.4). It is adaptive to the major-lobe limits of the event functions.

$$w_k^{(l)}(n) = \begin{cases} l_k^{(l)} - n, & \text{if } 1 \leq n < l_k^{(l)} \\ 0, & \text{if } l_k^{(l)} \leq n \leq r_k^{(l)} \\ n - r_k^{(l)}, & \text{if } r_k^{(l)} < n \leq N \end{cases} \tag{A.4}$$

where, $l_k^{(l)}$ and $r_k^{(l)}$ are the left and right limits of the major lobe of the event function $\phi_k(n)^{(l)}$. This definition of adaptive weighting function restricts the minor-lobes while allowing the major-lobe to evolve freely. Therefore, it gives rise to major-lobe expansion and contraction, with a simultaneous minor-lobe reduction, when the iterations are performed.

Quantitative balancing of the functional $J\left(\phi(n), \lambda\right)$:
Weighting factor $\lambda^{(l)}$ at the iteration step l are selected so as to balance the two error terms of the functional $J\Big(\phi^{(l+1)}(n), \lambda^{(l)}\Big)$ using the results obtained at the iteration steps $(l-1)$ and l, i.e., $\boldsymbol{\Phi}^{(l)}$ and $\mathbf{A}^{(l)}$, as given below.

$$\lambda^{(l)} = \sigma \times \left(\frac{\sum_{n=1}^{N}\sum_{i=1}^{P}\left(y_i(n) - \hat{y_i}^{(l)}(n)\right)^2}{\sum_{n=1}^{N}\sum_{k=1}^{K} w_k^{(l)}(n)^2 \phi_k^{(l)}(n)^2} \right)$$

where, $\hat{y_i}^{(l)}(n) = \sum_{k=1}^{K} a_{ik}^{(l)} \phi_k^{(l)}(n)$, σ is the constant.

The event functions matrix at the iteration step l, $\boldsymbol{\Phi}^{(l+1)}$, is calculated as follows:

$$\begin{aligned} \phi(n)^{(l+1)} &= \left(\mathbf{A}^{(l)^T}\mathbf{A}^{(l)} + \lambda^{(l)}\mathbf{W}_n^{(l)^T}\mathbf{W}_n^{(l)}\right)^{-1} \\ &\times \mathbf{A}^{(l)^T}\mathbf{y}(n), \quad 1 \leq n \leq N \end{aligned} \tag{A.5}$$

where,

$$\mathbf{W}_n^{(l)} = \text{diag}\left[\ w_1^{(l)}(n) \quad w_2^{(l)}(n) \cdots \quad w_K^{(l)}(n)\ \right]$$

Hence,

$$\boldsymbol{\Phi}^{(l+1)} = \left(\ \boldsymbol{\phi}(1)^{(l+1)} \quad \boldsymbol{\phi}(2)^{(l+1)} \quad \cdots \quad \boldsymbol{\phi}(N)^{(l+1)}\ \right)$$

A.1.3 Convergence of the Iterative Refinement Procedure

In this section, the convergence property of the iterative refinement procedure has been mathematically proved. In order to prove the convergence property of the event functions matrix, $\mathbf{\Phi}^{(l)}$, through iterations we prove the convergence property of an arbitrary column vector $\phi_n^{(l)}$, where $1 \leq n \leq N$. $\phi(n)$ hereafter denoted as ϕ_n for simplicity. Equation (A.5) can be written in scalar form as

$$\begin{aligned}\sum_{i=1}^{P} a_{ir}^{(l)} \sum_{k\neq r}^{K} a_{ik}^{(l)} \phi_k^{(l+1)}(n) &+ \left(\sum_{i=1}^{P} {a_{ir}^{(l)}}^2 + \lambda^{(l)} {w_r^{(l)}(n)}^2\right)\phi_r^{(l+1)}(n) \\ &= \sum_{i=1}^{P} a_{ir}^{(l)} y_i(n), \quad 1 \leq r \leq K\end{aligned}$$

By mimicking the traditional iterative methods for linear systems [114], where $\mathbf{A}$ is not updated at each iteration, i.e., once $a_{ik}^{(l)}$ and $\phi_j^{(l)}(n)$ are updated, the rth event function at the $(l+1)$th iteration step can be calculated as

$$\phi_r^{(l+1)}(n) = \frac{\sum_{i=1}^{P} a_{ir}^{(l)} y_i(n) - \sum_{k\neq r}^{K} \phi_k^{(l)}(n) \sum_{i=1}^{P} a_{ik}^{(l)} a_{ir}^{(l)}}{\sum_{i=1}^{P} {a_{ir}^{(l)}}^2 + \lambda^{(l)} {w_r^{(l)}(n)}^2} \tag{A.6}$$

Let

$$\mathbf{A}^{(l)^T}\mathbf{A}^{(l)} + \lambda^{(l)}\mathbf{W}_n^{(l)^T}\mathbf{W}_n^{(l)} = \mathbf{\Delta}_n^{(l)} = \mathbf{L}_n^{(l)} + \mathbf{D}_n^{(l)} + \mathbf{U}_n^{(l)}$$

where, $\mathbf{D}_n^{(l)}$ is a diagonal matrix and $\mathbf{U}_n^{(l)}$ and $\mathbf{L}_n^{(l)}$ are upper and lower triangular matrices with zeros on the diagonal. Note that $\mathbf{W}_n^{(l)^T}\mathbf{W}_n^{(l)}$ is a diagonal matrix, Equation (A.6) can be rewritten in matrix form:

$$\phi_n^{(l+1)} = -\mathbf{D}_n^{(l)^{-1}}\left(\mathbf{U}_n^{(l)} + \mathbf{L}_n^{(l)}\right)\phi_n^{(l)} + \mathbf{D}^{(l)^{-1}}\mathbf{A}^T\mathbf{y}_n \tag{A.7}$$

Suppose that, for the rth event function calculation, the first $r-1$ event functions at the $(l+1)$th iteration step have been obtained. It is natural to replace them in Equation (A.6), i.e.,

$$\begin{aligned}\phi_r^{(l+1)}(n) = \Bigg(&\sum_{i=1}^{P} a_{ir}^{(l)} y_i(n) - \sum_{k=1}^{r-1} \phi_k^{(l+1)}(n) \sum_{i=1}^{P} a_{ik}^{(l)} a_{ir}^{(l)} \\ &- \sum_{k=r+1}^{K} \phi_k^{(l)}(n) \sum_{i=1}^{P} a_{ik}^{(l)} a_{ir}^{(l)}\Bigg) \Big/ \left(\sum_{i=1}^{P} {a_{ir}^{(l)}}^2 + \lambda^{(l)} {w_r^{(l)}(n)}^2\right)\end{aligned} \tag{A.8}$$

which can also be written is matrix form:

$$\begin{aligned}\phi_n^{(l+1)} &= \left(\mathbf{L}_n^{(l)} + \mathbf{D}_n^{(l)}\right)^{-1}\left(\mathbf{A}^{(l)^T}\mathbf{y}_n - \mathbf{U}_n^{(l)}\mathbf{\Phi}_n^{(l)}\right) \\ &= -\left(\mathbf{L}_n^{(l)} + \mathbf{D}_n^{(l)}\right)^{-1}\mathbf{U}_n^{(l)}\phi_n^{(l)} + \left(\mathbf{L}_n^{(l)} + \mathbf{D}_n^{(l)}\right)^{-1}\mathbf{A}^{(l)^T}\mathbf{y}_n\end{aligned} \tag{A.9}$$

Since $\mathbf{A}$ and $\mathbf{W}_n$ are adaptively updated on each iteration, Equations (A.6), (A.7) and (A.8), (A.9) are thus denoted as an adaptive Jacobi iteration and an adaptive Gauss-Seidel iteration, respectively.

Set $\mathbf{B}_n^{(l)} = -\left(\mathbf{L}_n^{(l)} + \mathbf{D}_n^{(l)}\right)^{-1}\mathbf{U}_n^{(l)}$

and, $\mathbf{C}_n^{(l)} = \left(\mathbf{L}_n^{(l)} + \mathbf{D}_n^{(l)}\right)^{-1}\mathbf{A}^{(l)^T}\mathbf{y}_n$

Equation (A.9) can be rewritten as

$$\phi_n^{(l+1)} = \mathbf{B}_n^{(l)}\phi_n^{(l)} + \mathbf{C}_n^{(l)}$$

Assume that there is a true solution ϕ_n satisfying $\phi_n = \mathbf{B}_n^{(l)}\phi_n^{(l)} + \mathbf{C}_n^{(l)}$. The error at the lth iteration step, $\varepsilon_n^{(l)}$, is thus calculated as

$$\begin{aligned} \varepsilon_n^{(l)} &= \phi_n^{(l+1)} - \phi_n = \mathbf{B}_n^{(l)}\phi_n^{(l)} - \mathbf{B}_n^{(l)}\phi_n = \mathbf{B}_n^{(l)}\left(\phi_n^{(l)} - \phi_n\right) \\ &= \mathbf{B}_n^{(l)}\varepsilon_n^{(l-1)} = \cdots = \mathbf{B}_n^{(l)}\mathbf{B}_n^{(l-1)}\cdots\mathbf{B}_n^{(1)}\varepsilon_n^{(1)} \end{aligned} \tag{A.10}$$

It is obvious that the iterative refinement procedure converges iff

$$\lim_{l\to\infty} \parallel \varepsilon_n^{(l)} \parallel = 0$$

where, $\parallel \cdot \parallel$ is an arbitrary norm. Let us introduce a special norm, the L-norm.

$$\parallel X \parallel_L = \parallel LXL^{-1} \parallel_2$$

where, X is an arbitrary square matrix, L is a nonsingular matrix, and $\parallel \cdot \parallel_2$ is the spectral norm [150]. The L-norm has an important property as follows:

Theorem 1: If $\mathbf{S}(X) < 1$, then there exists a nonsingular matrix L such that

$$\parallel X \parallel_L < \mathbf{S}(X)$$

In this theorem $\mathbf{S}(X)$ is the spectral radius, which determined as the maximal module of the eigenvalues of matrix $\mathbf{X}$ [150]. It is just a mathematical manipulation to prove in the adaptive Gauss-Seidel iteration that [114]:

Theorem 2: If the matrix $\mathbf{\Delta}_n^{(l)}$ is positive-definite, then

$$\mathbf{S}\left(\mathbf{B}_n^{(l)}\right) < 1$$

Since $\mathbf{A}^{(l)^T}\mathbf{A}^{(l)}$ is nonnegative-definite [4] and it is easy to verify that $\lambda^{(l)}\mathbf{W}_n^{(l)^T}\mathbf{W}_n^{(l)}$ is positive-definite, $\mathbf{\Delta}_n^{(l)}$ is always positive-definite. Using theorem 1, theorem 2 and taking into account the Schwartz inequality of an arbitrary norm:

$$\begin{aligned} \lim_{l\to\infty} \parallel \mathbf{B}_n^{(l)}\cdots\mathbf{B}_n^{(1)} \parallel_L &\leq \lim_{l\to\infty} \parallel \mathbf{B}_n^{(l)} \parallel_L \cdots \parallel \mathbf{B}_n^{(1)} \parallel_L \\ &\leq \lim_{l\to\infty} \mathbf{S}\left(\mathbf{B}_n^{(l)}\right)\cdots\mathbf{S}\left(\mathbf{B}_n^{(1)}\right) = 0 \end{aligned}$$

Because a norm is nonnegative

$$\lim_{l\to\infty} \parallel \mathbf{B}_n^{(l)}\cdots\mathbf{B}_n^{(1)} \parallel_L = 0$$

This procedure proves the following theorem:

Theorem 3: A sufficient condition that the adaptive Gauss-Seidel iteration is convergent is that the matrix $\mathbf{\Delta}_n^{(l)}$ is positive-definite.

It follows that the convergence property of the iterative refinement procedure in S^{2}BEL-TD has been mathematically proved.

A.1.4 Alternative Termination Criterion of Iterations

As presented earlier, the event function error at lth iteration step, $\varepsilon_n^{(l)}$, converges to zero. This property can be employed as a hint to establish a new termination criterion.

In the modified S^{2}BEL-TD method, the root-mean-squared error between the two event functions matrices at the lth iteration step, $E_{rms}^{(l)}$, is defined as follows:

$$E_{rms}^{(l)} = \sqrt{\frac{1}{KN}\sum_{k=1}^{K}\sum_{n=1}^{N}\left(\phi_k^{(l+1)}(n) - \phi_k^{(l)}(n)\right)^2}$$

$E_{rms}^{(l)}$ can be considered for an alternatively terminative condition. Here, the two steps, refinement of event targets and refinement of event functions, are repeated until $E_{rms}^{(l)}$ drops below a certain predetermined threshold level, e.g., 0.005.

It is empirically shown that, in most cases, fewer iterations are required to shape up the event functions in the modified method compared to those needed in the original one.

A.2 Experimental Results

A set of 250 utterances of the ATR Japanese speech database [2] were selected for log spectral distortion evaluation. This speech data set consists of about 20 minutes of speech spoken by 10 speakers (5 males and 5 females) re-sampled at 8 kHz sampling frequency. 10^{th} order LSF parameter were calculated using a LPC analysis window of 40 ms at 10 ms frame intervals, and S^{2}BEL-TD analyzed using the original and modified S^{2}BEL-TD method. The window size for the SFTR calculation is $2M = 40$. $\lambda^{(0)} = 0.005$ and $\sigma = 1$ were selected as appropriate values for the initial weighting factor and balancing ratio, respectively, based on simulation results. The threshold levels for MLC^l and $E_{rms}^{(l)}$ were set as 1% and 0.005 in the original and modified S^{2}BEL-TD, respectively. The event rate is found to be about 20 events/sec in both cases.

Table A.1: Average log spectral distortion and percentage number of outlier frames for LSF parameters. The speech data set consists of 250 sentence utterances spoken by 10 speakers (5 males & 5 females) of the ATR Japanese speech database.

Method	Avg. LSD (dB)	$\leq$ 2 dB	2-4 dB	$>$ 4 dB
Original S^{2}BEL-TD	1.464	80.58 %	18.48 %	0.94 %
Modified S^{2}BEL-TD	1.444	81.90 %	17.17 %	0.93 %

Table A.1 gives the summary of the log spectral distortion results obtained for the above set of utterances with LSF as spectral parameters. Results indicate slightly better performance in terms of LSD in the case of the modified method. It is shown that the modified S^{2}BEL-TD method performed TD analysis faster than the original one in most cases.

In conclusion, the convergence property of the iterative refinement procedure in the S^{2}BEL-TD has been mathematically proved, which ensures the robust performance of the method. Additionally, some modifications have been made to the original S^{2}BEL-TD to improve its robustness in this respect. Experimental results confirm the efficiency of the modified method.

Bibliography

[1] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," *Journal of the Acoustical Society of Japan*, vol. E-11, no. 2, pp. 71-76, 1990.

[2] M. Abe, Y. Sagisaka, T. Umeda, and H. Kuwabara, "Speech database user's manual," *ATR Technical Report*, TR-I-0166, 1990.

[3] G. Ahlbom, F. Bimbot, G. Chollet, "Modeling spectral speech transitions using temporal decomposition techniques," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'87)*, pp. 13-16, 1987.

[4] W.A. Anisworth (Eds.), *Advances in Speech, Hearing and Language Processing*. Alden Press, Oxford, London, 1990.

[5] K.T. Assaleh and R.J. Mammone, "New LP-derived features for speaker identification," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 630-638, 1994.

[6] B.S. Atal and S.L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *Journal of the Acoustical Society of America*, vol. 50, no. 2 (Part II), pp. 637-655, 1971.

[7] B.S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *Journal of the Acoustical Society of America*, vol. 55, no. 6, pp. 1304-1312, 1974.

[8] B.S. Atal, "Efficient coding of LPC parameters by temporal decomposition," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'83)*, pp. 81-84, 1983.

[9] B.S. Atal, R.V. Cox, and P. Kroon, "Spectral quantization and interpolation for CELP coders," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'89)*, pp. 69-72, 1989.

[10] B.S. Atal, V. Cuperman, and A. Gersho, *Advances in Speech Coding*. Kluwer Academic Publishers, Dordrcht, The Netherlands, 1991.

[11] C.N. Athaudage, A.B. Brabley, and M. Lech, "Optimization of a temporal decomposition model of speech," *Proceedings of the International Symposium on Signal Processing and Its Applications (ISSPA'99)*, Brisbane, Australia, pp. 471-474, 1999.

[12] C.N. Athaudage, A.B. Brabley, and M. Lech, "Efficient compression of MELP spectral parameters using optimized temporal decomposition," *Proceedings of the Australian International Conference on Speech Science and Technology (SST-2000)*, Canberra, Australia, pp. 386-391, 2000.

[13] C.N. Athaudage, A.B. Brabley, and M. Lech, "On performance evaluation of a temporal decomposition based speech coder," *Proceedings of the International Conference on Information, Communications, and Signal Processing (ICICS-2001)*, Singapore, October 2001.

[14] C.N. Athaudage, "Speech compression using optimized temporal decomposition for voice storage applications," *Ph.D. Thesis*, Royal Melbourne Institute of Technology, Melbourne, Australia, 2001.

[15] G. Bailly, P.F. Marteau, and C. Aubry, "A new algorithm for temporal decomposition of speech. Application to a numerical model of coarticulation," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'89)*, pp. 508-511, 1989.

[16] F. Bimbot, G. Ahlbom, and G. Chollet, "From segmental synthesis to acoustic rules using temporal decomposition," *Proceedings of the 11^{th} ICPhS*, vol. 5, pp. 31-34, 1987.

[17] F. Bimbot, G. Chollet, P. Deleglise, and C. Montacie, "Temporal decomposition and acoustic-phonetic decoding of speech," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'88)*, pp. 445-448, 1988.

[18] F. Bimbot, G. Chollet, and P. Deleglise, "Speech synthesis by structured segments using temporal decomposition and a glottal excitation," *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech'89)*, pp. 183-186, 1989.

[19] F. Bimbot and B.S. Atal, "An evaluation of temporal decomposition," *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech'91)*, pp. 1089-1092, 1991.

[20] J.P. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437-1462, September 1997.

[21] Y.M. Cheng and D. O'Shaughnessy, "Short-term temporal decomposition and its properties for speech compression," *IEEE Transactions on Signal Processing*, vol. 39, no. 6, pp. 1282-1290, 1991.

[22] Y.M. Cheng and D. O'Shaughnessy, "On 450-600 b/s natural sounding speech coding," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 2, pp. 207-219, 1993.

[23] D.G. Childers, K. Wu, D.M. Hicks, and B. Yegnanarayana, "Voice conversion," *Speech Communication*, vol. 8, no. 2, pp. 147-158, 1989.

[24] D.G. Childers and K. Wu, "Gender recognition from speech. Part II: Fine analysis," *Journal of the Acoustical Society of America*, vol. 90, no. 4, pp. 1841-1856, 1991.

[25] D.G. Childers and C.K. Lee, "Vocal quality factor: Analysis, synthesis and perception," *Journal of the Acoustical Society of America*, vol. 90, no. 5, pp. 2394-2410, 1991.

[26] H.B. Choi, W.T.K. Wong, B.M.G. Cheetham, and C.C. Goodyear, "Interpolation of spectral information for low bit rate speech coding," *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech'95)*, pp. 1033-1036, 1995.

[27] G. Chollet, Y. Grenier, and S.M. Marcus, "Temporal decomposition and non-stationary modeling of speech," *Proceedings of the European Signal Processing Conference (Eusipco'86)*, pp. 365-368, 1986.

[28] C.J. Chung and S.H. Chen, "Variable frame rate speech coding using optimal interpolation," *IEEE Transactions on Communications*, vol. 42, no. 6, pp. 2215-2218, June 1994.

[29] A. Das, A.V. Rao, and A. Gersho, "Variable-dimension vector quantization," *IEEE Signal Processing Letters*, vol. 3, no. 7, pp. 200-202, July 1996.

[30] J.R. Deller, Jr., J.H.L. Hansen, and J.G. Proakis, *Discrete-Time Processing of Speech Signals*. IEEE Press, New York, 2000.

[31] S. Dimolitsas and J.G. Phipps, Jr., "Experimental quantification of voice transmission quality of mobile satellite personal communication systems," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 458-464, February 1995.

[32] P.J. Dix and G. Bloothooft, "A geometrical argument for imposing an additional constraints on temporal decomposition," *Proceedings of the International Conference on Spoken Language Processing (ICSLP'90)*, pp. 41-44, 1990.

[33] P.J. Dix and G. Bloothooft, "Segmentation by means of temporal decomposition," in *Visual representations of speech signals*, M. Cooke, S. Beet and M. Crawford (Eds.), John Wiley & Sons, Inc., New York, pp. 237-242, 1993.

[34] P.J. Dix and G. Bloothooft, "A breakpoint analysis procedure based on temporal decomposition," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 9-17, 1994.

[35] T. Eriksson, H.G. Kang, and P. Hedelin, "Low-rate quantization of spectrum parameters," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2000)*, pp. 1447-1450, 2000.

[36] T. Eriksson and H.G. Kang, "Pitch quantization in low bit-rate speech coding," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'99)*, pp. 489-492, 1999.

[37] J.S. Erkelens and P.M.T. Broersen, "LPC interpolation by approximation of the sample autocorrelation function," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 6, pp. 569-573, 1994.

[38] J.S. Erkelens and P.M.T. Broersen, "Analysis of spectral interpolation with weighting dependent on frame energy," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'94)*, pp. 481-484, 1994.

[39] F. Fallside and W.A. Woods (Eds.), *Computer Speech Processing*. Prentice-Hall, New York, 1995.

[40] Y. Fujino, "A study on speech morphing based on exchanging spectral events and fundamental frequencies," *Master's Thesis*, Japan Advanced Institute of Science and Technology, Hokuriku, Japan, 2001 (in Japanese).

[41] S. Furui, "On the role of spectral transition for speech perception," *Journal of the Acoustical Society of America*, vol. 80, no. 4, pp. 1016-1025, 1986.

[42] S. Furui and M.M. Sondhi, *Advances in Speech Signal Processing*. Marcel Dekker Inc., New York, 1991.

[43] S. Furui, "Recent advances in speaker recognition," *Pattern Recognition Letters*, vol. 18, no. 9, pp. 859-872, 1997.

[44] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, and N. Dahlgren, "Darpa TIMIT acoustic-phonetic continuous speech corpus CD-ROM," National Institute of Standards and Technology, October 1990.

[45] E.B. George, A.V. McCree, and V.R. Viswanathan, "Variable frame rate parameter encoding via adaptive frame selection using dynamic programming," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'96)*, pp. 271-274, 1996.

[46] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Dordrcht, The Netherlands, 1992.

[47] A. Gersho, "Advances in speech and audio compression," *Proceedings of the IEEE*, vol. 82, no. 6, pp. 900-918, June 1994.

[48] S. Ghaemmaghami and M. Deriche, "A new approach to very low-rate speech coding using temporal decomposition," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'96)*, pp. 224-227, 1996.

[49] S. Ghaemmaghami, M. Deriche, and B. Boashash, "Comparative study of different parameters for temporal decomposition based speech coding," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'97)*, pp. 1703-1706, 1997.

[50] S. Ghaemmaghami, M. Deriche, and B. Boashash, "On modeling event functions in temporal decomposition based coding," *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech'97)*, pp. 1299-1302, 1997.

[51] S. Ghaemmaghami and M. Deriche, "A new approach to modeling excitation in very low-rate speech coding," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'98)*, pp. 597-600, 1998.

[52] S. Ghaemmaghami, M. Deriche, and S. Sridharan, "Hierarchical temporal decomposition: A novel approach to efficient compression of spectral characteristics of speech," *Proceedings of the International Conference on Spoken Language Processing (ICSLP'98)*, pp. 2567-2570, 1998.

[53] S. Ghaemmaghami, S. Sridharan, and V. Chandran, "Speech compaction using temporal decomposition," *Electronics Letters*, vol. 34, no. 24, pp. 2317-2318, 1998.

[54] S. Ghaemmaghami and S. Sridharan, "Very low rate speech coding using temporal decomposition," *Electronics Letters*, vol. 35, no. 6, pp. 456-457, 1999.

[55] S. Ghaemmaghami, S. Sridharan, and V. Chandran, "Coding speech at very low rates using temporal decomposition based spectral interpolation and mixed excitation in the LPC model," *Applied Signal Processing*, vol. 6, no. 4, pp. 203-223, 1999.

[56] B. Gold and N. Morgan, *Speech and Audio Signal Processing–Processing and Perception of Speech and Music*. John Wiley & Sons, Inc., New York, 2000.

[57] R. Goldberg and L. Riek, *A Practical Handbook of Speech Coders*. CRC Press LLC, 2000.

[58] V. Goncharoff and M. Kaine-Krolak, "Interpolation of LPC spectra via pole shifting," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'95)*, pp. 780-7830, 1995.

[59] M. Honda and Y. Shiraki, "Very low-bit-rate speech coding," in *Advances in Speech Signal Processing*, S. Furui and M.M. Sondhi (Eds.), Marcel Dekker, pp. 209-230, 1992.

[60] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing*. Prentice-Hall, Inc., New Jersey, 2001.

[61] T. Islam and P. Kabal, "Partial-energy weighted interpolation of linear prediction coefficients," *Proceedings of the IEEE Workshop on Speech Coding*, pp. 105-107, 2000.

[62] F. Itakura, "Line spectrum representation of linear predictive coefficients of speech signal," *Journal of the Acoustical Society of America*, vol. 57, p. S35, April 1975.

[63] ITU-T, *Coding of Speech at 8 kbit/s using Conjugate-Structure Algebraic Code-Excited Linear Prediction (CS-ACELP)*. ITU-T Recommendation G.279, March 1996.

[64] N. Iwahashi and Y. Sagisaka, "Speech spectrum conversion conversion based on speaker interpolation and multi-functional representation with weighting by radial basis function networks," *Speech Communication*, vol. 16, no. 2, pp. 139-151, 1995.

[65] T.-P. Jung, A.K. Krishnamurthy, S.C. Ahalt, M.E. Beckman and S-H. Lee, "Deriving gestural scores from articulation-movement records using temporal decomposition," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, pp. 2-18, 1996.

[66] P. Kabal and R.P. Ramachandran, "The computation of line spectral frequencies using chebyshev polynomials," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, no. 6, pp. 1419-1426, December 1986.

[67] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, no. 3-4, pp. 187-207, 1999.

[68] H.K. Kim and H.S. Le, "Interlacing properties of line spectrum pair frequencies," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 1, pp. 87-91, 1999.

[69] M.Y. Kim, N.K. Ha and S.R. Kim, "Linked split vector quantization of LPC parameters," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'95)*, pp. 741-744 , 1995.

[70] S.J. Kim, S.H. Lee, W.J. Han, and Y.H. Oh, "Efficient quantization of LSF parameters based on temporal decomposition," *Proceedings of the International Conference on Spoken Language Processing (ICSLP'98)*, pp. 2575-2578 , 1998.

[71] S.J. Kim and Y.H. Oh, "Efficient quantisation method for LSF parameters based on restricted temporal decomposition," *Electronics Letters*, vol. 35, no. 12, pp. 962-964, 1999.

[72] S.J. Kim and Y.H. Oh, "Split vector quantization of LSF parameters with minimum of dLSF constraint," *IEEE Signal Processing Letters*, vol. 6, no. 9, pp. 227-229, 1999.

[73] S.J. Kim, "Very low bit rate speech coding based on temporal decomposition of line spectral frequencies," *Ph.D. Thesis*, Korea Advanced Institute of Science and Technology, Taejon, Korea, 2000.

[74] T. Kinnunen, "Designing a speaker-discriminative adaptive filter bank for speaker recognition," *Proceedings of the International Conference on Spoken Language Processing (ICSLP'02)*, pp. 2325-2328, 2002.

[75] D.H. Klatt and L.C. Klatt, "Analysis, synthesis and perception of voice quality variations among female and male talkers," *Journal of the Acoustical Society of America*," vol. 87, no. 2, pp. 820-857, 1990.

[76] W.B. Kleijn and K.K. Paliwal, *Speech Coding and Synthesis*. Elsevier Science B.V., Amsterdam, The Netherlands, 1995.

[77] H.P. Knagenhjelm and W.B. Kleijn, "Spectral dynamics is more important than spectral distortion," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'95)*, pp. 732-735, 1995.

[78] A.M. Kondoz, *Digital Speech – Coding for Low Bit Rate Communications Systems*. John Wiley & Sons Ltd, Chichester, 1994.

[79] B. Kovesi, S. Saoudi, J.M. Boucher, and G. Horvath, "Real time vector quantization of LSP parameters," *Speech Communication*, vol. 29, no. 1, pp. 39-47, 1999.

[80] H. Kuwabara and Y. Sagisaka, "Acoustic characteristics of speaker individuality: Control and conversion," *Speech Communication*, vol. 16, no. 2, pp. 165-173, 1995.

[81] B. Lawlor and A.D. Fagan, "A novel efficient algorithm for voice gender conversion," *Proceedings of the 14th International Congress of Phonetic Sciences (ICPhS'99)*, pp. 77-80, 1999.

[82] R. Laroia, N. Phamdo, and N. Farvardin, "Robust and efficient quantization of speech LSP parameters using structured vector quantizers," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'91)*, pp. 641-644, 1991.

[83] M.S. Lee, H.K. Kim, and H.S. Lee, "LPC analysis incorporating spectral interpolation for speech coding," *Electronics Letters*, vol. 35, no. 3, pp. 200-201, 1999.

[84] M.S. Lee, H.K. Kim, and H.S. Lee, "A new distortion measure for spectral quantization based on the LSF intermodel interlacing property," *Speech Communication*, vol. 35, no. 3-4, 2001.

[85] A.N. Lemma, W.B. Kleijin, and Ed.F. Deprettere, "LPC quantization using wavelet based temporal decomposition of the LSF," *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech'97)*, vol. 3, pp. 1259-1262, 1997.

[86] Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantiser design," *IEEE Transactions on Communications*, vol. 28, pp. 84-95, 1980.

[87] J.M. Lopez-Soler, V. Sanchez, A. de la Torre, and A.J. Rubio-Ayuso, "Linear interframe dependencies for very low bit-rate speech coding," *Speech Communication*, vol. 34, no. 4, pp. 333-349, 2001.

[88] J. Makhoul, "Linear Prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561-580, 1975.

[89] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proceedings of the IEEE*, vol. 73, no. 11, pp. 1551-1587, November 1985.

[90] R.J. Mammone, X. Zhang, and R.P. Ramachandran, "Robust speaker recognition–a feature-based approach," *IEEE Signal Processing Magazine*, vol. 13, no. 5, pp. 58-71, September 1996.

[91] S.M. Marcus and R.A.J.M. Van Lieshout, "Temporal decomposition of speech," *IPO Annual Progress Report 19*, pp. 26-31, 1984.

[92] P.F. Marteau, G. Bailly, and M.T. Janot-Giorgetti, "Stochastic model of diphone-like segments based on trajectory cocepts," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'88)*, pp. 615-618, 1988.

[93] A.V. McCree and T.P. Barnwell, "A mixed excitation LPC vocoder model for low bit rate speech coding," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 4, pp. 242-250, July 1995.

[94] A. McCree, K. Truong, E.B. George, T.P. Barnwell, and V. Viswanathan, "A 2.4 kbits/s MELP coder candidate for the new U.S. federal standard," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 200-203, 1996.

[95] A. McCree and J.C. De Martin, "A 1.7 kb/s MELP coder with improved analysis and quantization," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'98)*, pp. 2565-2568, 1998.

[96] G.A. Mian and G. Riccardi, "A localization property of line spectrum frequencies," *IEEE Transactions on Speech and Audio Processing*, Vol.2, no. 4, pp. 536-539, October 1994.

[97] H. Mizuno and M. Abe, "Voice conversion algorithm based on piecewise linear conversion rules of formant frequency and spectrum tilt," *Speech Communication*, vol. 16, no. 2, pp. 153-164, 1995.

[98] C. Montacie, P. Deleglise, F. Bimbot, and M.J. Caraty, "Cinematic techniques for speech processing: Temporal decomposition and multivariate linear prediction," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'92)*, pp. 153-156, 1992.

[99] R.W. Morris and M.A. Clements, "Modification of formants in the line spectrum domain," *IEEE Signal Processing Letters*, vol. 9, no. 1, pp. 19-21, 2002.

[100] E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Communication*, vol. 9, no. 5-6, pp. 453-467, 1990.

[101] E. Moulines and Y. Sagisaka (Eds.), *Voice Conversion: State of the Art and Perspectives (Special Issue of Speech Communication)*. Elsevier Science, The Netherlands, vol. 16, no. 2, February 1995.

[102] A.C.R. Nandasena, "A new approach to temporal decomposition of speech and its application to low-bit-rate speech coding," *Master's Thesis*, Japan Advanced Institute of Science and Technology, Hokuriku, Japan, 1997.

[103] A.C.R. Nandasena and M. Akagi, "Spectral stability based event localizing temporal decomposition," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'98)*, pp. 957-960, 1998.

[104] A.C.R. Nandasena, P.C. Nguyen, and M. Akagi, "Spectral stability based event localizing temporal decomposition," *Computer Speech and Language*, vol. 15, no. 4, pp. 381-401, 2001.

[105] M. Narendranath, H.A. Murthy, S. Rajendran, and B. Yegnanarayana, "Transformation of formants for voice conversion using artificial neural networks," *Speech Communication*, vol. 16, no. 2, pp. 207-216, 1995.

[106] M. Niranjan and F. Fallside, "Temporal decomposition: A framework for enhanced speech recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'89)*, pp. 655-658, 1989.

[107] P.C. Nguyen, T. Ochi, and M. Akagi, "Modified restricted temporal decomposition and its application to low bit rate speech coding," *IEICE Transactions on Information and Systems*, vol.E86-D, no. 3, pp. 397-405, 2003.

[108] P.E. Papamichalis, *Practical Approaches to Speech Coding*. Prentice-Hall, New York, 1987.

[109] K.K. Paliwal and B.S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 1, pp. 3-14, January 1993.

[110] K. K. Paliwal, "Interpolation properties of linear prediction parametric representations," *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech'95)*, pp. 1029-1032, 1995.

[111] J. Pan and T.R. Fischer, "Vector quantization of speech line spectrum pair parameters and reflection coefficients," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 106-115, March 1998.

[112] L. Rabiner and R.W. Schafer, *Digital Processing of Speech Signals*. Prentice-Hall, New York, 1978.

[113] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Prentice-Hall, New York, 1993.

[114] A. Ralston and P. Rabinnowitz, *First Course in Numerical Analysis*. McGraw-Hill, New York, 1978.

[115] R.P. Ramachandran and R. Mammone, *Modern Methods of Speech Processing*. Kluwer Academic Publishers, The Netherlands, 1995.

[116] R.P. Ramachandran, M.M. Sondhi, N. Seshadri, and B.S. Atal, "A two codebook format for robust quantization of line spectral frequencies," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 3, pp. 157-167, May 1995.

[117] R.P. Ramachandran, K.R. Farrell, R. Ramachandran, and R. Mammone, "Speaker recognition–general classifier approaches and data fusion methods," *Pattern Recognition*, vol. 35, no. 12, pp. 2801-2821, December 2002.

[118] D.A. Reynolds and R.C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 72-83, 1995.

[119] D.A. Reynolds, "An overview of automatic speaker recognition technology," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'02)*, pp. 4072-4075, 2002.

[120] C.H. Ritz, I.S. Burnett, and J. Lukasiak, "Very low rate speech coding using temporal decomposition and waveform interpolation," *Proceedings of the IEEE Workshop on Speech Coding*, pp. 29-31, 2000.

[121] C.H. Ritz and I.S. Burnett, "Split temporal decomposition," *Proceedings of the Australian International Conference on Speech Science and Technology (SST-2000)*, Canberra, Australia, pp. 416-420, 2000.

[122] C.H. Ritz, I.S. Burnett, "Temporal decomposition: A promising approach to low rate wideband speech compression," *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech 2001)*, pp. 2315-2318, 2001.

[123] A.E. Rosenberg, F.K. Soong, "Recent research in speaker recognition," in: *Advances in Speech Signal Processing*, S. Furui and M.M. Sondhi (Eds.), Marcel Dekker Inc., New York, pp. 701-738, 1991.

[124] D. O' Shaughnessy, *Speech Communication: Human and Machine*, 2nd Edition. IEEE Press, New York, 2000.

[125] H. Scheffe, "An analysis of variance for paired comparisons," *Journal of the American Statistical Association*, vol. 47, pp. 381-400, 1952.

[126] Y. Shiraki and M. Honda, "LPC speech coding based on variable-length segment quantization," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 9, pp. 1437-1444, September 1988.

[127] Y. Shiraki and M. Honda, "Extraction of temporal pattern of spectral sequence based on minimum distortion criterion," *Proceedings of the Autumn Meeting of ASJ*, pp. 233-234, 1991 (in Japanese).

[128] F.K. Soong and B.-H. Juang, "Line spectrum pair (LSP) and speech data compression," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'97)*, pp. 1101-1104, 1984.

[129] F.K. Soong, A.E. Rosenberg, B.-H. Juang, and L.R. Rabiner, "A vector quantization approach to speaker recognition," *AT&T Technical Journal*, vol. 66, no. 2, pp. 14-26, 1987.

[130] F.K. Soong and B.H. Juang, "Optimal quantization of LSP parameters," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 1, pp. 15-24, January 1993.

[131] A.S. Spanias, "Speech coding: A tutorial review," *Proceedings of the IEEE*, vol. 82, no. 10, pp. 1541-1540, October 1994.

[132] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 131-142, 1998.

[133] L.M. Supplee, R.P. Cohn, J.S. Collura, and A.V. McCree, "MELP: The new federal standard at 2400 bps," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'97)*, pp. 1591-1594, 1997.

[134] M. Tang, C. Wang, and S. Seneff, "Voice transformations: From speech synthesis to mammalian vocalizations," *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech 2001)*, pp. 353-356, 2001.

[135] I.R. Titze, "Physiologic and acoustic differences between male and female voices," *Journal of the Acoustical Society of America*, vol. 85, no. 4, pp. 1699-1707, 1989.

[136] T.E. Tremain, "The government standard linear predictive coding algorithm: LPC-10," *Speech Technology*, pp. 40-49, April 1982.

[137] C. Tsao and R.M. Gray, "Matrix quantization design for LPC speech using the generallized Lloyd algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, no. 3, pp. 537-545, June 1985.

[138] T. Toda, H. Saruwatari, and K. Shikano, "Voice conversion algorithm based on Gaussian mixture model with dynamic frequency warping of STRAIGHT spectrum," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'01)*, pp. 841-844, 2001.

[139] T. Toda, H. Saruwatari, and K. Shikano, "High quality voice conversion based on Gaussian mixture model with dynamic frequency warping," *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech'01)*, pp. 349-352, 2001.

[140] T. Umezaki and F. Itakura, "Analysis of time fluctuating characteristics of linear predictive coefficients," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'86)*, pp. 1257-1260, 1986.

[141] A.M.L. Van Dijk-Kappers and S.M. Marcus, "Temporal decomposition of speech," *Speech Communication*, vol. 8, no. 2, pp. 125-135, 1989.

[142] A.M.L. Van Dijk-Kappers,"Comparison of parameter sets for temporal decomposition," *Speech Communication*, vol. 8, no. 3, pp. 203-220, 1989.

[143] A.M.L. Van Dijk-Kappers, "Temporal decomposition of speech and its relation to phonetic information," *Ph.D. Thesis*, Eindhoven University of Technology, The Netherlands, 1989.

[144] V.R. Viswanathan, J. Makhoul, R.M. Schwartz, and A.W.F. Hugging, "Variable frame rate transmission: Methodology and application to narrow-band LPC speech coding," *IEEE Transactions on Communications*, vol. 30, no. 4, pp. 674-686, 1982.

[145] H.D. Wang, G. Bailly, D. Tufelli, "Automatic segmentation and alignment of continuous speech based on temporal decomposition," *Proceedings of the International Conference on Spoken Language Processing (ICSLP'90)*, pp. 457-460, 1990.

[146] D. Wong, B.-H. Juang, and A.H. Gray, Jr., "An 800 bps vector quantization LPC vocoder," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-30, no. 5, pp. 770-779, October 1982.

[147] K. Wu and D.G. Childers, "Gender recognition from speech. Part I: Coarse analysis," *Journal of the Acoustical Society of America*," vol. 90, no. 4, pp. 1828-1840, 1991.

[148] C.S. Xydeas and C. Papanastasious, "Split matrix quantization of LPC parameters," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 2, pp. 113-125, March 1999.

[149] M. Yong, "A new interpolation technique for CELP coders," *IEEE Transactions on Communications*, vol. 42, no. 1, pp. 34-38, January 1994.

[150] D.M. Young, *Iterative Solution of Large Linear Systems*. Academic Press Inc., New York, 1971.

[151] P. Zolfaghari, Y. Atake, K. Shikano, and H. Kawahara, "Investigation of analaysis and synthesis parameters of STRAIGHT by subjective evaluation," *Proceedings of the International Conference on Spoken Language Processing (ICSLP 2000)*, pp. 498-501, 2000.

Publications

[1] P.C. Nguyen, T. Ochi, and M. Akagi, "Modified restricted temporal decomposition and its application to low bit rate speech coding," *IEICE Transactions on Information and Systems*, vol.E86-D, no. 3, pp. 397-405, March 2003.

[2] A.C.R. Nandasena, P.C. Nguyen, and M. Akagi, "Spectral stability based event localizing temporal decomposition," *Journal of Computer Speech and Language*, Academic Press, vol. 15, no. 4, pp. 381-401, October 2001.

[3] P.C. Nguyen and M. Akagi, "Limited error based event localizing temporal decomposition. and its application to variable-rate speech coding," *Journal of Speech Communication*, Elsevier Science (conditionally accepted).

[4] P.C. Nguyen and M. Akagi, "Improvement of the restricted temporal decomposition method for line spectral frequency parameters," *Proceedings of the 27th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2002)*, Orlando, Florida, USA, pp. 265-268, May 2002.

[5] P.C. Nguyen and M. Akagi, "Limited error based event localizing temporal decomposition," *Proceedings of the 11th European Signal Processing Conference (EUSIPCO 2002)*, Toulouse, France, pp. 239-242, September 2002.

[6] P.C. Nguyen, T. Ochi, and M. Akagi, "Coding speech at very low rates using STRAIGHT and temporal decomposition," *Procceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002 - InterSpeech 2002)*, Denver, Colorado, USA, pp. 1849-1852, September 2002.

[7] P.C. Nguyen and M. Akagi, "Variable-rate speech coding using STRAIGHT and temporal decomposition," *Proceedings of the 2002 IEEE Speech Coding Workshop (SCW 2002)*, Tsukuba, Japan, pp. 26-28, October 2002.

[8] P.C. Nguyen and M. Akagi, "Efficient coding of speech excitation parameters by temporal decomposition," Accepted to *the 9th Australian International Conference on Speech Science and Technology (SST 2002)*, Melbourne, Australia, December 2002.

[9] P.C. Nguyen, M. Akagi, and T.B. Ho, "Temporal decomposition: A promising approach to VQ-based speaker identification," *Proceedings of the 28th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003)*, Hong Kong, China, pp. 184-187, April 2003.

[10] P.C. Nguyen, M. Akagi, and T.B. Ho, "Temporal decomposition: A promising approach to VQ-based speaker identification," *Proceedings of the 4th IEEE International Conference on Multimedia and Expo (ICME 2003)*, Baltimore, Maryland, USA, pp. 617-620, July 2003.

[11] P.C. Nguyen and M. Akagi, "Efficient quantization of speech excitation parameters by temporal decomposition," *Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech 2003 - InterSpeech 2003)*, Geneva, Switzerland, pp. 449-452, September 2003.

[12] P.C. Nguyen and M. Akagi, "Male to female voice transformation based on modification of formants in the line spectrum domain," *Technical Report*, Japan Advanced Institute of Science and Technology, Hokuriku, Japan, 2003.

[13] P.C. Nguyen and M. Akagi, "Improvement of the restricted temporal decomposition method for LSF parameters," *Proceedings of the 2001 Autumn Meeting of the Acoustical Society of Japan*, Oita, Japan, pp. 267-268, October 2001.

[14] P.C. Nguyen and M. Akagi, "Limited error based event localizing temporal decomposition," *Proceedings of the Spring Meeting of the Acoustical Society of Japan*, Kanagawa, Japan, pp. 325-326, March 2002.

[15] T. Ochi, P.C. Nguyen, and M. Akagi, "Very low bit rate speech coding using STRAIGHT and temporal decomposition," *Proceedings of the Spring Meeting of the Acoustical Society of Japan*, Kanagawa, Japan, pp. 349-350, March 2002 (in Japanese).

[16] P.C. Nguyen and M. Akagi, "Variable-rate speech coding based on STRAIGHT using temporal decomposition," *Proceedings of the Autumn Meeting of the Acoustical Society of Japan*, Akita, Japan, pp. 231-232, September 2002.

[17] P.C. Nguyen and M. Akagi, "On the application of temporal decomposition to VQ-based speaker identification," *Proceedings of the Autumn Meeting of the Acoustical Society of Japan*, Tokyo, Japan, pp. 117-118, March 2003.