University of California
Los Angeles

Source and Channel Coding for Speech Transmission and Remote Speech Recognition

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical Engineering

by

Alexis Pascal Bernard

2002

The dissertation of Alexis Pascal Bernard is approved.

Alan McCree

Stanley Osher

Lieven Vandenberghe

Richard D. Wesel

Abeer A.H. Alwan, Committee Chair

University of California, Los Angeles

2002

In memory of

Bonne-Mammie

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my deepest gratitude to my advisor, Professor Abeer Alwan. She has been such a wonderful advisor that it is impossible to describe the extent of my appreciation. Not only was she actively involved in this research, she provided the best research environment one could hope for. I appreciate her intellectual guidance which encourages the exploration of new ideas.

I have also had the opportunity to work closely with Professor Richard Wesel and Dr. Xueting Liu. Their involvement in our research is greatly appreciated. I acknowledge the remaining members of my doctoral committee, Professors Lieven Vandenberghe, Stanley Osher and Dr. Alan McCree.

Many of my SPAPL fellows helped my research through insightful discussions. I particularly would like to acknowledge Brian and Mark.

To the friends made during my years at UCLA, Adina, Andreas, Cedric, Fred, Greg, Iqbal, Jeet, Karen, Katherine, Ksenija, Lisa, Matthias, Markus, Max, Munish, Panchi, Raman, Sebastien, and others whom I hope can forgive the oversight, you made my time in Los Angeles the most enjoyable. To my good friends back in Belgium, Andrea, Regis, Sandrine and many others, for whom the distance was not an obstacle either for their friendship or their support. To Alyssa, who helped me with all her love and support throughout my journey as a graduate student, I am forever grateful. Finally, I would like to thank my parents and my brothers Frédéric and Nicolas for their immeasurable love and support.

This work was supported in part by ST Microelectronics, Broadcom Corporation, Hughes Research Laboratories (HRL) and Conexant through the UC Micro Program and in part by the Belgian American Educational Foundation (BAEF) and the National Science Foundation (NSF).

Vita

1973	Born, Brussels, Belgium.
1996	B.S. in Electrical Engineering B.A. in Philosophy Université Catholique de Louvain, Louvain-La-Neuve, Belgium
1998	M.S. in Electrical Engineering University of California, Los Angeles, Los Angeles, California
1997–2002	*Research Assistant*, Electrical Engineering Department, UCLA
1997–2000	*Teaching Assistant*, Electrical Engineering Department, UCLA
6-9/2000	*Intern*, Texas Instruments, Dallas, Texas
6-9/1999	*Intern*, Texas Instruments, Dallas, Texas
8-9/1997	*Intern*, Alcatel Telecom research, Antwerp, Belgium
Award	Belgian American Educational Foundation (B.A.E.F) fellow

Publications

A. Bernard, X. Liu, R. D. Wesel and A. Alwan, "Speech transmission using rate-compatible trellis codes and embedded source coding", *IEEE Transactions on Communications*, vol. 50, no. 2 pp. 309–320, February 2002.

A. Bernard and A. Alwan, "Channel decoding - Viterbi recognition for wireless applications", *Proceedings of Eurospeech*, Aalborg, Denmark, vol. 4, pp. 2703–2706, Sept. 2001.

A. Bernard and A. Alwan, "Source and channel coding for remote speech recognition over error-prone channels", *Proceedings of International Conference on Audio, Speech and Signal Processing*, Salt Lake City, Utah, vol. 4, pp. 2613–2616, May 2001.

A. Bernard and A. Alwan, "Perceptually based and embedded wideband CELP coding of speech", *Proceedings of Eurospeech*, Budapest, Hungary, vol. 4, pp. 1543–1546, Sept. 1999

A. Bernard, X. Liu, R. D. Wesel and A. Alwan, "Embedded joint source-channel coding of speech using symbol puncturing of trellis codes", *Proceedings of International Conference on Audio, Speech and Signal Processing*, Phoenix, Arizona, vol. 5, pp. 2427–2430, March 1999.

A. Bernard, X. Liu, R. Wesel and A. Alwan, "Channel adaptive joint source-channel coding of speech", *Proceedings of the 32nd Asilomar Conference on Signals, Systems, and Computers*, Monterey, California, vol. 1, pp. 357–361, November 1998.

A. McCree, T. Unno, A. Anandakumar, A. Bernard and E. Paksoy, "An embedded adaptive multi-rate wideband speech coder", *Proceedings of International Conference on Audio, Speech and Signal Processing*, Salt Lake City, Utah, vol. 4, pp. 2613–2616, May 2001.

J.J. Quisquater, B. Macq, M. Joye, N. Degand and A. Bernard, "Practical solution to authentication of images with a secure camera", *Proceedings of SPIE International Society for Optical Engineering*, vol. 3022, pp. 290-7, 1997.

Abstract of the Dissertation

Source and Channel Coding for Speech Transmission and Remote Speech Recognition

by

Alexis Pascal Bernard

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2002

Professor Abeer A.H. Alwan, Chair

This dissertation addresses the issue of designing source and channel coding techniques for two types of speech processing applications: speech transmission and remote speech recognition.

In the first part, adaptive multi-rate (AMR) speech transmission systems that switch between operating modes depending on channel conditions are presented. We address the design of such an adaptive scheme using variable bit rate embedded source encoders and rate-compatible channel coders providing unequal error protection. A novel technique, the rate-compatible punctured trellis code (RCPT) for obtaining unequal error protection via progressive puncturing of symbols in a trellis, is presented and compared with the rate-compatible punctured convolutional code with and without bit-interleaved coded modulation. The perceptually-based speech coder proposed displays a wide range of bit error sensitivities, and is used in combination with rate-compatible punctured channel codes providing adequate levels of protection. The resulting system operates over a wide range of channel conditions with graceful performance degradation as the channel signal-to-noise ratio decreases.

In the second part, we present a framework for developing source coding, channel coding, channel decoding, and frame erasure concealment techniques adapted for remote speech recognition applications. It is shown that speech recognition, as opposed to speech coding, is more sensitive to channel errors than channel erasures. Appropriate channel coding design criteria are determined. For channel decoding, we introduce a novel technique for combining soft decision decoding with error detection. The technique outperforms the often used hard decision strategy. In addition, frame erasure concealment techniques are used at the decoder to deal with unreliable frames. At the recognition stage, we present a technique to modify the recognition engine to take into account the time-varying reliability of the decoded feature after channel transmission. The resulting engine, referred to as weighted Viterbi recognition (WVR), further improves recognition accuracy. Together, source coding, channel coding and the modified recognition engine are shown to provide good recognition accuracy over a wide range of communication channels at very low bit rates.

CHAPTER 1

Introduction

Two inventions have played an important role in the development of the field of speech signal processing. First, the invention of the telephone by Alexander Graham Bell in 1876 permitted the transmission of speech, after transducing the speech acoustic waveform into an electrical signal whose intensity varies with the pressure of the waveform. Transduction of the speech signal into an electrical signal allowed for analog processing of speech. Another landmark in the history of speech processing was the passage from an *analog* to a *digital* representation of bandwidth limited speech signals using Nyquist's sampling theorem and amplitude quantization. This paved the road for digital speech processing applications, including *speech coding* to reduce the bit rate necessary for speech transmission, *speech recognition* to allow machines to understand spoken speech, and *speech synthesis* to enable machines to speak.

This dissertation focuses on two speech applications: coding and recognition. For the former, we present source and channel coding techniques for the development of adaptive multi-rate (AMR) speech communication systems. For the latter, we develop source and channel coding solutions suitable for low bit rate remote speech recognition over error-prone channels.

In adaptive multi-rate speech transmission, the idea is to design a commu-

nication system, including adaptive source and channel coding techniques, that allow for reliable speech transmission over a wide range of channel conditions. We address the design of such an adaptive scheme using embedded source encoders and rate-compatible channel coders providing adequate unequal error protection.

In remote speech recognition, speech recognition features are extracted from the speech signal by the client and transmitted to the server for recognition. We address the design of source coding, channel coding and channel decoding techniques that improve performance of remote speech recognition over noisy channels.

The remainder of this chapter is organized as follows. Sections 1.1 and 1.2 analyze important characteristics of human speech production and auditory perception that have been exploited in speech processing applications. Speech coding and recognition are introduced in Sections 1.3 and 1.4, respectively. Section 1.5 provides an overview of digital communication. Sections 1.6 and 1.7 introduce the two applications considered in this dissertation, adaptive multi-rate speech transmission and remote speech recognition. Finally, a road map of the dissertation is presented in Section 1.8.

1.1 Speech production overview

1.1.1 Physiological mechanisms of speech production

A schematic diagram of the human vocal mechanism is shown in Figure 1.1. This is a representation of the linear model of speech production, such as described in [4, 5]. The model assumes that speech is produced by acoustically exciting a time-varying cavity, the vocal tract. The speech spectrum $S(\omega)$ is the result of the multiplication (convolution in the time domain) of the excitation spectrum

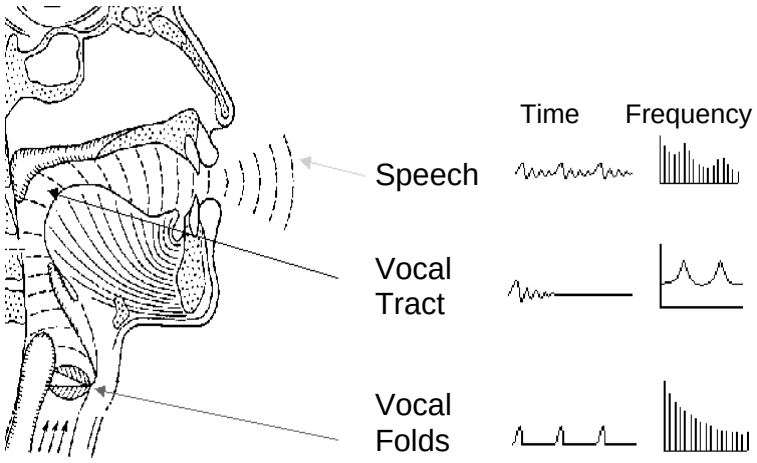


Figure 1.1: Speech production overview (inspired by [1]).

$U(\omega)$ by the vocal tract transfer function $H(\omega)$,

$$S(\omega) = U(\omega) \cdot H(\omega). \tag{1.1}$$

The various speech sounds are produced by adjusting the type of excitation as well as the shape of the vocal tract.

The *vocal tract* begins at the opening of the vocal cords, or glottis, and ends at the lips. The shape and cross-sectional profile of the vocal tract is adjusted by articulatory motion, which includes manipulating the tongue, lips, velum, mouth and lower jaw. The shape of the vocal tract determines its frequency response. Because the vocal tract is essentially a tube with varying cross-sectional areas, the transfer function of energy from the excitation source to the output can be described in terms of resonance frequencies of the tube. Such resonances are called *formant frequencies* in speech production. Typically, there are three formants below 3500 Hz for a human vocal tract [6]. For most speech sounds, the envelope of the power spectrum is an important factor for determining their linguistic interpretation.

The *excitation* consists of air flow from the lungs which is forced through the trachea and the vocal cords. Many speech signals can be classified as either voiced or unvoiced. For *voiced* sounds, the vocal folds open and close with regular periodicity, called *pitch*. Pitch periodicity can be changed by varying the tension of the vocal cords. Voiced speech segments have a harmonic frequency-domain structure and a nearly periodic time-domain signal. For *unvoiced* sounds, the vocal cords are relaxed and the glottis is open. The air flow either passes through a constriction in the vocal tract and thereby becomes turbulent (e.g. /s/ and /sh/), or builds up pressure behind a point of total closure within the vocal tract, and when the closure is opened, the pressure is suddenly released, causing a brief transient sound (e.g. /p/ and /t/).

Typically, the excitation signal has a spectral tilt of -12 dB per octave, and the radiation from the lips introduces spectral slope of +6 dB per octave.

Note that the linear speech production model forms a basis for some linear prediction speech coders, as will be seen in Section 1.3.

1.1.2 Implications for speech coding and recognition

The speech production mechanism induces some characteristics on the resulting speech signals which are exploited by speech processing applications.

Table 1.1, inspired by [1], illustrates from the speech production point of view the different sources of information found in the speech signal depending on its frequency range. In the range of 1-15 Hz, the phonetic articulation is found and can be seen in a spectrogram by analyzing the formant and voicing transitions. The pitch or voicing information is found in the interval 40-400 Hz, the typical range for fundamental frequencies. Finally, the spectral range is found in the interval 400-12000 Hz.

	Syllabic range 1 – 15 Hz	**Voicing range** 40 – 400 Hz	**Spectral range** 200 – 12000 Hz
Speech Production	Articulatory motion	Vocal fold vibration	Articulatory position
Speech Perception	Formant motion	Pitch frequency	Formant position
Speech Spectrum	Time-varying spectrum	Pitch harmonics	Spectral envelope
Speech Signal	Time-varying signal	Long-term correlation	Short-term correlation
Speech Coding	Coder parameters prediction	Long-term prediction	Short-term prediction
Speech Recognition	Markov models	Removed in most applications	Spectral estimates

Table 1.1: Classifying frequency ranges of speech.

In terms of the implications of the speech production model on the characteristics of the speech signal, the harmonic structure of the spectrum corresponds to "long-term" autocorrelation in the speech signal, while the power-spectral envelope corresponds to "short-term" autocorrelation [7]. Both redundancies can be removed in speech coding, using short-term and long-term prediction (see Section 1.3).

Significant phonetic information is thought to be carried by the spectral envelope governed by articulatory position. Spectral estimates of the speech signals are captured by the speech recognizer front-end, while stochastic (Markov) models capture time variations in the speech spectrum. Pitch information is typically not utilized in speech recognition systems (see Section 1.4).

1.2 Auditory perception overview

In speech transmission, the receiver is the human ear. Insight into human auditory perception can therefore lead to better design of speech coding [8] and potentially of recognition systems [9]. The field of psychoacoustics has made significant progress toward characterizing human auditory perception, particularly the time-frequency capabilities of the human ear [10, 11, 12]. A brief review is included here to motivate some of the techniques used both in speech transmission (Part I) and speech recognition (Part II).

1.2.1 Physiological mechanisms of speech perception

Acoustic pressure waves pass through the nearly passive outer and middle ears to excite the basilar membrane within the snail-shaped cochlea in the inner ear. The stiffness of the membrane decreases along its length, from the beginning (base) at the stapes to the end (apex). This non-uniform waveguide results in a frequency-to-place transduction on the basilar membrane, which works as follows. As a traveling horizontal acoustic wave moves inside the cochlea, its velocity diminishes (due to the reduced stiffness), decreasing the wavelength of the membrane disturbance and concentrating the energy per unit length over an increasingly smaller region [1]. This energy is then rapidly dissipated in the large amplitude deformation of the basilar membrane at that location. The position i on the basilar membrane where such large deformation occurs depends on the frequency ω_i of the acoustic wave. High frequency stimuli concentrate and dissipate energy close to the base, while low frequency stimuli travel further toward the apex, completing the frequency-to-place transduction.

Motion of the basilar membrane is then transduced by the bending of tiny hair

cells implanted on the membrane. The base of each hair cell is innervated with peripheral auditory nerves whose firing rate varies monotically but not linearly with the amplitude of the hair cell bending. The relationship between hair cell bending $y_i(t)$ at position i on the basilar membrane and the average rate of nerve firing $r(t, \omega_i)$ is often approximated by a power-law:

$$r(t, \omega_i) \approx y_i(t)^{\alpha}. \tag{1.2}$$

The average rate of nerve firing provides the impression of loudness, which can be thought of as *perceptual magnitude*. The power-law relationship with $\alpha < 1$ between hair cell bending and nerve firing enables humans to listen to a wide range of sound intensities. Often, the power-law model of loudness for humans assumes $\alpha = 0.33$, so that if the intensity of the signal is increased by 9 dB, the loudness is approximately doubled. This power-law relationship between loudness and intensity, commonly referred to as *Stevens' law* [13, 14], has been extensively studied [15] and is sometimes included in the computation of Automatic Speech Recognition (ASR) features (see Section 1.4).

1.2.2 Critical bands and masking phenomena

The human inner ear behaves as a bank of band pass filters with non-uniform bandwidths. These filters, which perform frequency selectivity and spectral analysis, are referred to as critical bands and their bandwidths as critical bandwidths. A distance of one critical band is often referred to as one *bark*. The bandwidth of the critical bands is constant up to 500 Hz, after which it increases exponentially.

The division of the audibility frequency range in critical bands accounts for the frequency domain phenomenon of *masking*. With *simultaneous masking*, a low level signal (the maskee) can be made inaudible by a simultaneously occurring stronger signal (the masker), if the masker and maskee are close enough to each

other in frequency. A masking threshold below which any signal will not be audible can be measured. The masking threshold depends on the sound pressure level (SPL) and the frequency of the masker. We distinguish between two types of simultaneous masking, *tone-masking-noise* and *noise-masking-tone*. In both cases, the underlying reason for masking is the same. The presence of a strong noise or a strong tone masker creates an excitation of sufficient strength on the basilar membrane at the critical band location to effectively block the transmission of the weaker signal [8]). The global masking threshold is the log sum of 1) the hearing threshold (defined as the minimum amount of energy needed in a pure tone signal to be detected by a listener in a quiet environment [16]), 2) the tone-masking-noise threshold and 3) the noise-masking-tone threshold. The global masking threshold is also referred to as the Noise Masking Curve (NMC) or the level of Just Noticeable Distortion (JND). The signal is completely masked if the Signal-to-Mask Ratio (SMR) is negative on a logarithmic scale.

Masking properties and critical bands theories are extensively used in speech and audio coding, reducing the bit rate necessary to perform perceptually transparent coding (see Chapter 2. The success of audio codecs such as the MPEG standard [17, 18] can be attributed to the applications of the signal masking theory. Conceptually, the masking property tells us that we can permit greater amounts of noise in the vicinity of the formant regions.

Another perceptual frequency scale is the *Mel* frequency scale, which is characterized as follows: 1) 1000 mels corresponds to 1000 Hz, and 2) a tone at $x/2$ mels is half as high as a tone at x mels for experimental subjects. The bark and Mel scales are proportional to each other; 1 bark is approximately equal to 100 mels. The Mel scale is extensively used in speech recognition by taking a bank of bandpass filters linearly spaced in the Mel frequency domain to perform spectral

analysis (see Section 1.4.1).

Evaluation of the speech spectrum on a perceptual frequency scale (Mel or bark) and evaluation of the auditory spectrum has led to the development of two perceptually motivated ASR features (see Section 1.4).

1.3 Speech coding overview

The subject of speech coding [19, 20, 21] has been an area of research for several decades. Speech coders are present in our everyday life and their use is often taken for granted. For example, speech coding is present in most digital telephone systems and in every cellular application.

1.3.1 Speech coding strategies

Speech coders, whose goal is to represent the analog speech signal in as few binary digits as possible, can be described as belonging to one of three fundamentally different coding classes: *waveform coders*, *vocoders*, and *hybrid coders*.

A *waveform* coder attempts to mimic the waveform as closely as possible by transmitting actual time- or frequency-domain magnitudes. For example, in Pulse Coded Modulation (PCM), the input speech itself is quantized. In differential PCM (DPCM) and adaptive DPCM (ADPCM), the prediction residual is quantized. In addition, Subband Coders (SBC) are also waveform coders. Speech quality produced by waveform coders is generally high, although at high bit rates. Chapter 2 presents two embedded waveform speech coders, a perceptually-based subband coder and an embedded ADPCM coder.

Vocoders, or *parametric coders*, analyze the waveform to extract parameters that in some cases represent a speech production model. The waveform

is synthetically reproduced at the receiver based on these quantized parameters. Vocoder types include formant, homomorphic vocoders, as well as Linear Prediction Coding (LPC) and Sinusoidal Transform Coding (STC). Vocoders can generally achieve higher compression ratios than waveform coders; however, they provide more artificial speech quality.

In *hybrid* coders, the high compression efficiency of vocoders and high-quality speech reproduction capability of waveforms are combined to produce good quality speech at medium-to-low bit rates. The so-called analysis-by-synthesis coders, such as the Coded Excited Linear Prediction (CELP), GSM, and Mixed Excitation Linear Prediction (MELP) are all hybrid coders. These coders are used in Chapter 5 to evaluate the effect of speech coding on speech recognition.

1.3.2 Linear predictive coding

As mentioned in Table 1.1, long-term and short-term correlations in the speech signal imply redundancies that can be exploited by speech coders to achieve compression. One solution for removing the redundancy in the speech signal is to use signal prediction and to transmit only the residual signal. If the samples of a digital speech signal are assumed to be Gaussian random variables, *linear prediction* (LP) of a speech sample from earlier speech samples is optimal in the least-square sense [7, 22]. Two types of linear predictors are typically used.

Long-term predictors (LTP): Long-term predictors are associated with the harmonic structure of speech spectra, as dictated by the periodicity of the vocal cord vibrations for voiced sounds. The purpose of the LTP filter is to extract this pitch redundancy from the signal.

Short-term predictors (STP): Short-term predictors are associated with the formant structure of speech spectra, as dictated by the vocal tract. Assuming

the linear production model (Eq. 1.1), the vocal tract can be modeled by an auto-regressive (AR) all-pole model of order p such that the speech signal $S(z)$ can be synthesized as $S(z) = U(z)H(z)$, where $U(z)$ represents the excitation signal, and the synthesis filter $H(z) = \frac{1}{A(z)}$ is $H(z) = \frac{1}{1-\sum_{k=1}^{p} \alpha_k z^{-k}}$.

The linear prediction coefficients $\{\alpha_i\}_{i=1}^{p}$ are computed by minimizing the mean-square error of the prediction error. Typically, p is chosen equal to 10 for a 4 kHz signal, to represent three formants and a spectral tilt. For an 8 kHz bandwidth signal, it is common to take $p = 16$.

STP filters can be viewed differently depending on whether we consider the filter at the encoder or decoder. At the encoder, the filter can be regarded as an *analysis* filter, used to remove redundancy from the speech signal. At the decoder, the inverse filter can be thought of as the *synthesis* filter, which models the vocal tract. Its transfer function describes the envelope of the speech signal.

Linear prediction coefficients α_i must be quantized and transmitted to the receiver, for signal reconstruction (waveform coders) or speech synthesis (parametric coders). However, direct quantization of α_i coefficients may lead to instability in the all-pole filtering operation and audible distortion. Quantization is usually performed on transformed, yet mathematically equivalent, versions of the linear prediction coefficients. Parameters typically used include the reflection coefficients (RC), which also have a physical interpretation related to the lossless tube model for speech production [23], and the log area ratios (LAR) which present better quantization properties. The most commonly used representation is the *line spectral frequencies* (LSFs) introduced by Itakura [24, 25].

LSFs are obtained by decomposing the polynomial $A(z)$ into two polynomials, with even and odd symmetry. This is accomplished by taking a difference and sum between $A(z)$ and its conjugate function, $P(z) = A(z) + z^{-(p+1)}A(z^{-1})$ and

$Q(z) = A(z) - z^{-(p+1)}A(z^{-1})$. Because of the symmetry, the roots of both $P(z)$ and $Q(z)$ are on the unit circle and their angles correspond to the frequencies ω_{p_n} and ω_{q_n} $(1 \leq n \leq p/2)$, which are precisely the LSFs. In terms of speech production, LSFs can be thought of as the zeros and poles of the impedance of the lossless tube model of the vocal tract as seen from the glottis $Z_g(z) = \frac{Q(z)}{P(z)}$.

LSFs are the customary representation of LP coefficients due to the useful characteristics they present such as: 1) the pole frequencies ω_n are approximately equal to the formant frequencies; 2) if $A(z)$ is minimum-phase (roots inside the unit circle), then ω_{p_n} and ω_{q_n} also fall onto the unit circle and are interlaced $(0 < \omega_{p_1} < \omega_{q_1} < \omega_{p_2} < \omega_{q_2} < \ldots < 1)$; 3) LSFs are correlated with each other, so intra-frame prediction and vector quantization are possible; 4) LSFs vary smoothly over time, so inter-frame prediction is possible; and 5) LSFs can effectively be interpolated.

For the converse, *i.e.* the reconstruction of the LPC filter from the LSFs, we see that $A(z) = \frac{1}{2}[P(z) + Q(z)]$. Note that the LSFs will be used in Chapter 5 for quantizing the linear prediction estimate of perceptual spectra.

1.3.3 Evaluation of speech coders

Speech and audio coders can be evaluated in terms of five attributes: bit rate, speech quality [26], delay, complexity, and robustness to acoustic noise and channel errors. Speech quality can be measured subjectively and objectively. Subjective measurements are obtained from listening tests, whereas objective measurements are computed from the original and decoded speech signals.

The most widely reported subjective measure is the Mean Opinion Score (MOS), which is obtained by averaging test results of listeners who are asked to rate their impressions on a five point-scale. The typical MOS subjective qualities

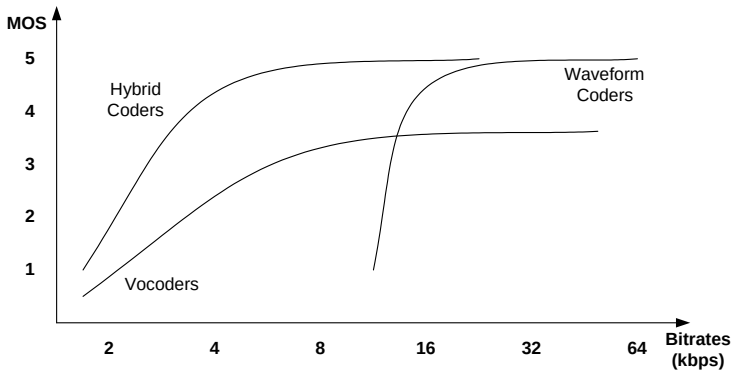


Figure 1.2: Quality comparison of speech coding schemes (after [2]).

versus coding bit rates are shown in Figure 1.2 for different coding strategies.

Widely used objective measures include mean squared error-based measures. A popular measure is the signal-to-noise ratio (SNR), which is a long-term measure of the accuracy of reconstructed speech. Temporal variations can be better detected and evaluated using a short-time SNR for each segment of speech. The segmental SNR (SEGSNR) is then defined as the average of the short-time SNRs. Modified versions of SEGSNR will be used to evaluate the quality of coders presented in Chapter 2. Other objective evaluations utilize knowledge about the human auditory system to derive perceptually motivated objective measures, such as ITU P.861 EMBSD (Enhanced Modified Bark Spectral Distortion) [27] and P.862 PESQ (Perceptual Evaluation of Speech Quality) [28].

1.4 Speech recognition overview

Figure 1.3 shows the structure of speech recognizers based on Hidden Markov Models (HMM). During training, each element in the dictionary (word, syllable, phoneme) is modeled by a transition network (Markov model) with a small

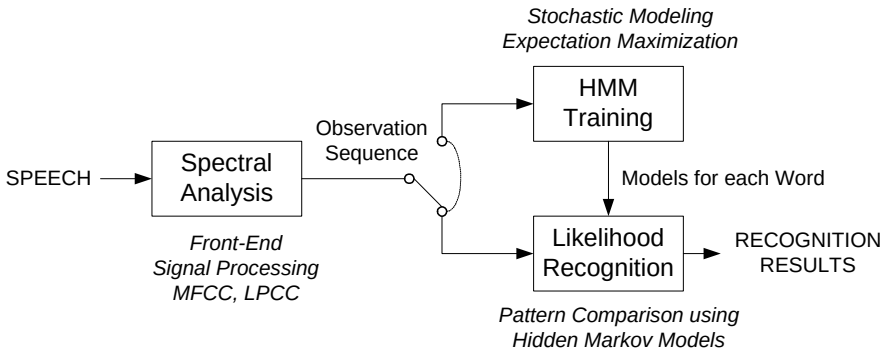


Figure 1.3: Structure of a speech recognizer based on HMM models.

number of states. Each state corresponds, in an indeterminable sense, to a set of temporal events in the spoken word [29]. During *recognition*, the likelihood that the observed sequence is generated from a state sequence for each vocabulary word is computed, and the word with the highest accumulated probability is selected. Typically, the sequence of observations consists of short-time spectral estimates, also called *features* ($\boldsymbol{O} = \boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T$), which are computed by the front-end component of the recognizer.

1.4.1 Front-end signal processing

Typical feature vectors for ASR include the Mel Frequency Cepstral Coefficients (MFCC) and the Linear Prediction Cepstral Coefficients (LPCC). LPCCs can be extracted from standard or perceptual linear prediction models.

Mel Frequency Cepstral Coefficients (MFCC)

The human ear resolves frequencies non-linearly across the audio spectrum. Empirical evidence suggests that designing a front-end to operate in a similar non-linear manner improves recognition performance.

A simple filterbank designed to provide non-linear resolution on the Mel frequency scale can be used. With MFCCs, the filters used are triangular, and they are equally spaced along the Mel scale. To implement this filterbank, the window of speech data is analyzed using a Fourier transform, and the magnitude coefficients are correlated with each triangular filter and accumulated. Each bin holds a weighted sum representing the spectral magnitude in that filterbank channel. The magnitudes of the power estimates from each channel are finally compressed using a logarithmic function. The resulting spectral estimates reflect two of the most studied aspects of auditory signal processing: frequency selectivity and magnitude compression.

Because the spectral estimates are somewhat smooth across filter number and highly correlated, each frame is roughly decorrelated using the Discrete Cosine Transform (DCT) to obtain the Mel-Frequency Cepstral Coefficients (MFCCs) c_i calculated from the log filterbank amplitudes m_j as follows,

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{M} m_j \cos\left(\frac{\pi i}{N}(j - 0.5)\right) \qquad 1 \leq i \leq N \tag{1.3}$$

where M is the number of filterbank channels and N the number of cepstral coefficients. MFCCs are the acoustic features of choice for many speech recognition applications.

Linear Prediction Cepstral Coefficients (LPCC)

As mentioned earlier, in linear prediction (LP) analysis, the vocal tract transfer function is modeled by an all-pole filter with transfer function $H(z) = \frac{1}{1-\sum_{k=1}^{p} \alpha_k z^{-k}}$, where p is the number of poles. The filter coefficients α_k are chosen to minimize the mean square filter prediction error summed over the analysis window.

Cepstral parameters can be obtained by taking the inverse Fourier transform

of the log spectrum. In the case of linear prediction cepstra, the spectrum is the linear prediction spectrum, which can be obtained from the Fourier transform of the filter coefficients. However, it can be seen that the cepstra can be more efficiently computed using a simple recursion [6]:

$$c_n = \alpha_n + \sum_{k=1}^{n-1} \frac{k}{n} \, c_k \, \alpha_{n-k}. \tag{1.4}$$

Perceptual Linear Prediction Cepstral Coefficients (P-LPCC)

Classic LP analysis techniques are used to obtain a smoothed spectral envelope of the speech spectrum $P(\omega)$. However, a main disadvantage of the LP all-pole model in speech analysis is that the LP computed spectrum $H(\omega)$ approximates $P(\omega)$ equally well at all frequencies of the analysis band. This property is inconsistent with human hearing (Section 1.2). Above about 500 Hz, the spectral resolution of hearing decreases with frequency. Furthermore, hearing is highly sensitive in the middle frequency range of the audible spectrum.

The perceptual linear predictive (PLP) speech analysis technique, introduced in 1990 by Hermansky [30], models three properties of human audition to derive an estimate of the auditory spectrum: 1) the critical-band resolution, 2) the equal-loudness curve, and 3) the intensity-loudness power law (Eq. 1.2).

The auditory spectrum is then approximated by an all-pole autoregressive linear prediction model. Typically, a low order prediction ($p = 5$–6) is effective in suppressing speaker-dependent details of the auditory spectrum, which is much less than the order of most other spectral representations.

The three types of ASR features presented above will be analyzed in Chapter 5 for their potential applications to remote speech recognition.

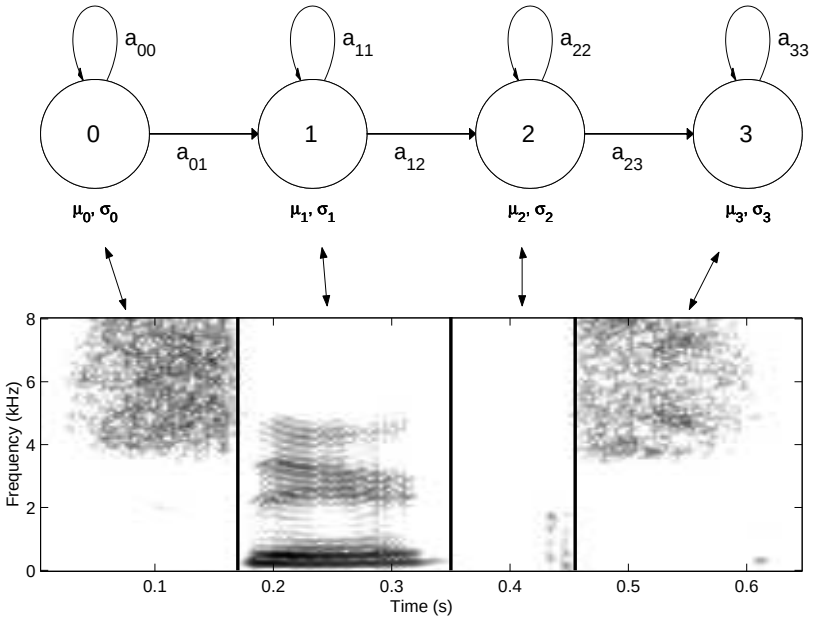


Figure 1.4: A schematic representation of a hidden Markov model.

1.4.2 Hidden Markov Models

Hidden Markov models (HMM) are used to provide a characterization of the non-stationary stochastic process represented by the sequences of feature vectors. In HMM based speech recognition, it is assumed that the sequence of observed speech vectors for each element of the dictionary (word, syllable or phone) is generated by a Markov model with a small number of states, as shown in Figure 1.4 for a four state HMM model of the digit '6'.

A Markov model is a finite state machine which changes state once every time unit. The transition from state i to state j is governed by the discrete probability $\boldsymbol{a}_{ij}$. For most continuous density HMM-based speech recognition systems, statistics of each state for each model are represented using multivariate Gaus-

sian mixture densities. The probability $b_j(\boldsymbol{o}_t)$ of observing the N_F-dimensional feature vector $\boldsymbol{o}_t$ in the j^{th} state is then

$$b_j(\boldsymbol{o}_t) = \sum_{m=1}^{N_M} w_m \frac{1}{\sqrt{(2\pi)^{N_F}|\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\boldsymbol{o}_t - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\boldsymbol{o}_t - \boldsymbol{\mu})}, \tag{1.5}$$

where N_M is the number of mixture components, w_m is the mixture weight, and the parameters of the multivariate Gaussian mixture are its mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

The model is hidden since only the observation sequence $\boldsymbol{O}$ is known during the recognition stage and the underlying state sequences are hidden. Instead of associating a state with every observation, the model state specifies the *statistics* of the observed feature vectors for a specific temporal segment of the sound.

1.4.3 Three uses of an HMM

Three problems can be solved using the formalism of an HMM [31]: 1) recognition; given a model λ for each word and an observation sequence $\boldsymbol{O}$, what is the most likely word spoken?, 2) segmentation; given $\boldsymbol{O}$ and λ, what is the state sequence Q which maximizes $P(\boldsymbol{O}, Q|\lambda)$?, and 3) training; given $\boldsymbol{O}$ and an initial estimate of λ, how can λ be modified to increase $P(\boldsymbol{O}|\lambda)$?

Segmentation: A significant complication in speech recognition is that speech is non-stationary and statistics of the different speech segments change considerably across the word. Within the word, there may be temporal segments where the statistics are nearly stationary, but the durations of these segments will also change with different speaking styles and rates. This motivates the *alignment* or *segmentation* problem.

In general, there are two related approaches to solve the temporal alignment problem with HMM speech recognition. The first is an application of

dynamic programming or Viterbi decoding, and the second is the more general forward/backward algorithm. The Viterbi algorithm [32] (essentially the same algorithm as the forward probability calculation except that the summation is replaced by a maximum operation) is typically used for segmentation and recognition and the forward/backward for training.

The Viterbi algorithm finds the state sequence Q that maximizes the probability

$$P^* = \max_{\text{all } Q} P(Q, \boldsymbol{O}|\lambda). \tag{1.6}$$

In order to calculate P^* for a given model λ, we define the metric $\phi_j(t)$, which represents the maximum likelihood of observing the features $(\boldsymbol{o}_1, \ldots \boldsymbol{o}_t)$ given that we are in state j at time t. Based on dynamic programming, this partial likelihood can be computed efficiently using the following recursion

$$\phi_j(t) = \max_i \{\phi_i(t-1)a_{ij}\}\, b_j(\boldsymbol{o}_t). \tag{1.7}$$

The maximum likelihood $P^*(\boldsymbol{O}|\lambda)$ is then given by $P^*(\boldsymbol{O}|\lambda) = \max_j \{\phi_j(T)\}$.

The recursion (1.7) forms the basis of the Viterbi Algorithm (VA) whose idea is that there is only one "best" path to state j at time t.

Recognition: Speech recognition (given that one has already somehow trained the models λ_i) translates to the problem of which model maximizes the likelihood $P(\boldsymbol{O}|\lambda)$. Probabilities $P(\boldsymbol{O}|\lambda_i)$ can be computed using the forward or the Viterbi algorithm. The models can be either whole words for limited vocabulary speech recognition or sub-words (syllables, phonemes, tri-phones) for large vocabulary continuous speech recognition.

HMM Training: When training an HMM, a set of exemplars corresponding to a particular model is used to provide iterative improvements for both the estimates of the multi-variate distributions of the feature vectors, and the

state-transition probabilities. There is no known way to analytically solve for the model parameter set that maximizes the probability of the observation sequence. However, we can choose λ such that its likelihood, $P(\boldsymbol{O}|\lambda)$, is locally maximized using an iterative procedure, such as the Baum-Welch (also known as the expectation-maximization (EM) method).

New model parameters are obtained by averaging across the training set for each model. The contribution of each state transition and each observed feature are weighted by the probabilities of having been at that state during the time of that feature. Given the set of re-estimated models, the algorithm iterates, re-aligning the original data to the updated models. This iterative process converges to a local maximum. Probabilities of state occupation are found using the forward/backward algorithm. By summing all possible previous paths, the partial forward probabilities of observing the first M frames of the exemplar and ending at a specific state can be inductively computed from the $M-1$ forward probabilities. A similar iterative process is used to obtain backward probabilities of observing the last L frames. Combining the forward and backward probabilities provides an estimate for the probability of state occupation.

1.5 Digital communication overview

Figure 1.5 illustrates the basic elements of a digital communication system. This model can be traced back to Shannon's original paper on information theory [33, 34]. The source encoder removes redundancy from the representation of the signal and is followed by a *separate* channel encoder that adds controlled redundancy for channel error protection. Digital data at the output of the channel encoder is passed to the modulator, which serves as an interface to the channel by mapping digital information sequences to analog waveforms. The demodulator, channel

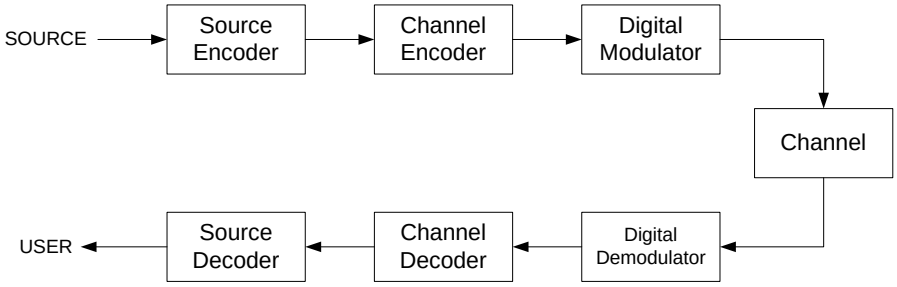


Figure 1.5: Basic elements of a digital communication system.

decoder and source decoder at the receiver perform the inverse operations so a reconstructed signal can be presented to the user.

The motivation for splitting source and channel coding is twofold. First, for many types of channels, optimal performance can be achieved with separate source and channel coders. This is usually referred to as the "source-channel separation theorem" [33, 35] which is, however, only valid for stationary ergodic sources, and provided that the source and channel encoders are allowed to operate on blocks of arbitrarily large length. The second motivation is increased flexibility. A channel code designed to minimize the influence of channel errors can be used to protect any incoming data. This results in high flexibility since data can be outputs from any source coder and the channel coder can be designed independently.

However, the need for keeping the complexity of realistic systems reasonably low implies that only small block lengths are feasible. Furthermore, speech is not stationary ergodic. In such cases, the most favorable source and channel encoding schemes might be the result of a *jointly* designed source and channel encoder, minimizing the average distortion between the source signal and its reproduction after source and channel decoding.

1.5.1 Source coding

The task of the source coder is to code the source signal so that the receiver can recover the signal with as little distortion as possible, under the constraint that the source coding rate cannot exceed the channel capacity. Signal compression can be done by removing two types of information from the original signal: the redundant and perceptually irrelevant parts of the signal.

Speech coders examine properties of the human speech production model (Section 1.1) to analyze redundancies in the speech signal, and the human auditory system (Section 1.2) to determine inaudible parts in the signal (see Chapter 2).

1.5.2 Modulator and demodulator

The modulator serves as an interface to the communication channel. The modulator may simply map each binary digit into one of two possible waveforms. Alternatively, the modulator may transmit k-bit blocks at a time by using $M = 2^k$ possible waveforms $s_m(t)$.

At the receiving end of the digital communication system, it is convenient to subdivide the receiver into two parts: the signal demodulator and the detector. The function of the *signal demodulator* is to process the received channel noise corrupted waveform $r(t) = s_m(t) + n(t)$ and to reduce each waveform to a scalar or a vector that represents an estimate of the transmitted data symbol (binary or M-ary). The *detector*, which follows the demodulator, may decide on whether the transmitted bit (for binary transmission) is a "0" or a "1". In such case, the detector has made a **hard decision**. Hard decision corresponds to binary quantization of the demodulator output ($Q = M$). More generally, we may consider a detector that quantizes the demodulator output to $Q > M$ levels. In

the extreme case where no quantization is performed ($Q = \infty$), we say that the detector has made a **soft decision**.

Soft decision is superior by about 2–3 dB to hard decision for coded digital modulation over the AWGN channel. Over multi-path fading channels, hard decision decoding typically suffers a loss in diversity when compared to soft decision decoding, which may result in significant performance degradation [36]. The distinction between hard and soft decision decoding for linear block codes will be evaluated in Chapter 6.

1.5.3 Channel coding

The function of the channel encoder is to introduce some redundancy in the information sequence, which can be used at the receiver to overcome the effects of noise and interference encountered in the transmission of the signal through the channel. The encoding process generally involves taking k information bits and mapping each k-bit sequence into a unique n-bit sequence. The amount of redundancy introduced is measured by the ratio k/n, also called the *code rate*. The effectiveness of channel coding is given by the *coding gain*, the attainable savings in energy per information bit required to achieve a given error probability to uncoded transmission [37]. In chapters 3 and 6, we use three types of channel codes: block coders, convolutional codes and trellis codes.

Block codes

The encoder for a block code divides the information sequence into message blocks of k information symbols. The encoder maps each k-symbols message independently into an n-symbols message called a *codeword*. If q is the size of the symbol alphabet, there are q^k different possible codewords at the encoder output

corresponding to the q^k different possible messages. This set of q^k codewords of length n is called an (n,k) block code. For a code to be useful, one must have $k \leq n$. When $k < n$, $n - k$ redundant symbols (bits if $q = 2$) are added to each message to form a codeword, providing the code with the capability of combating channel noise [38].

Besides the code rate parameter k/n, an important parameter of a block code is the *Hamming weight* of each codeword, which is the number of nonzero elements that it contains. By listing the various Hamming weights, w, of the codewords, and the number of codewords, A_w, at each weight, we obtain the *distance spectrum* or equivalently the *weight distribution* of the code. Certainly, the most important aspect of the distance structure of a block code is the *minimum Hamming distance* between any two codewords, denoted $d_{\min}$. The guaranteed error-correcting capability of a block code is $t = \lfloor \frac{d_{\min}-1}{2} \rfloor$. If the code is used strictly for error detection purposes, the guaranteed error detection capability of the code is $d_{\min} - 1$. Hybrid modes of operation employing concurrent error correction and detection are also possible.

Block codes can be efficiently described if they are constructed with a certain algebraic or geometric structure. Linear codes have an algebraic structure such that the sum of two codewords is also a codeword. Linearity facilitates analysis of the code's error protection capabilities. Every codeword has similar distance properties, so the error protection capability need only be evaluated for a single codeword (*e.g.* $d_{\min}$ is the Hamming weight of the minimum weight non-zero codeword).

Soft decision based error detection of block codes for remote speech recognition applications will be studied in Chapter 6.

Convolutional codes

Unlike block codes, convolutional codes are not restricted to transmitting codewords in blocks. In a rate k/n convolutional code, k information bits enter at the same time in a shift register structure with ν memory elements and generate n output symbols. The number of previous symbols on which the present state and output are dependent is the constraint length ν of the code. The number of states in the convolutional code is q^{ν}, where q is the size of the symbol alphabet, usually 2 (binary). There is a one-to-one correspondence between the information sequence, the register state sequence in the shift register, and the code output sequence.

The trellis of a convolutional encoder is a representation of all possible shift register state transitions over time. Each path through the trellis corresponds to one of the possible state sequences. The minimum Hamming weight between two paths in the trellis is called the *free distance* of the trellis, d_{free}.

The convolutional decoder estimates the path through the trellis that was followed by the encoder. There are a number of techniques for decoding convolutional codes. The most important is the Viterbi algorithm which performs maximum-likelihood (ML) decoding. Given a received sequence of symbols, a path length is associated to each branch in the trellis, which is the log-likelihood of the branch transition given the observation of the symbols received from the channel. ML decoding is then reduced to a shortest path search through the trellis. The depth of the shortest path search in the trellis is called the traceback depth.

Trellis coded modulation

Trellis Coded Modulation (TCM) was introduced in 1982 by Ungerboeck [39].

It combines both coding and modulation to achieve significant coding gains without compromising bandwidth efficiency. TCM schemes [40] employ redundant non-binary or quaternary modulation in combination with a finite state encoder, which decides the selection of the modulation signals to generate coded signal sequences. TCM uses signal set expansion to provide redundancy, and to design jointly coding and signal mapping functions so as to directly maximize the free distance (minimum Euclidean distance) between coded signals. At the receiver, the signals are decoded by a soft decision ML sequence decoder.

Rate-compatible, punctured, and bit-interleaved versions of trellis and convolutional codes providing unequal error protection will be investigated in Chapter 3.

1.6 Adaptive multi-rate speech transmission

In speech communication systems, a major challenge is to design a system that provides good speech quality over a wide range of channel conditions. For rate-constrained systems, one solution consists of allowing the transceivers to monitor the state of the communication channel and to dynamically allocate the bitstream between source and channel coders accordingly. For low SNR channels, the source coder operates at low bit rates, thus allowing powerful forward error control. For high SNR channels, the source coder uses its highest rate resulting in high speech quality. An adaptive algorithm selects the best source-channel coding combination out of a collection of available source and channel coders operating at different rates based on estimates of channel quality.

Speech coders whose operating bit rate is allowed to vary, thereby adapting the rate to channel conditions, are called adaptive multi-rate (AMR) speech coders

(e.g. [41, 42, 43, 44, 45]). Embedded source coders, for which the bitstream of the source encoder operating at low bit rates is embedded in the bitstream of the coder operating at higher rates, form one class of AMR source encoders.

Multi-rate speech coding is not new. Techniques like voice activity detection (VAD) or entropy-matching coding are proposed to decrease average coding bit rates. However, few AMR systems describing both source and channel coding have been presented. Some AMR systems that combine different types of variable rate CELP coders for source coding with RCPC and cyclic redundancy check (CRC) codes for channel coding were presented as candidates for the European Telecommunications Standards Institute (ETSI) GSM AMR codec standard. In [46], UEP is applied to perceptually-based audio coders (PAC). The bitstream of the PAC is divided into two classes and punctured convolutional codes are used to provide different levels of protection, assuming a BPSK constellation.

Channel coders whose redundancy is allowed to vary, thereby adapting the coding rate after the transmitter acquires information about channel conditions, are called variable-rate channel coders. Rate-compatible coders, for which the bit or symbol stream of the channel encoder operating at low redundancy is embedded in the bit or symbol stream of the channel encoder operating at high redundancy, form one class of variable-rate channel encoders.

Rate-compatible channel codes, such as Hagenauer's rate-compatible punctured convolutional codes (RCPC) [47], are a collection of codes providing a family of channel coding rates. By puncturing bits in the bitstream, the channel coding rate of RCPC codes can be varied instantaneously, providing unequal error protection (UEP) [48] by imparting on different segments different degrees of protection. Cox et al. [49] illustrate how RCPC codes can be used to build a

speech transmission scheme for mobile radio channels. Their approach is based on a subband coder with dynamic bit allocation proportional to the average energy of the bands. RCPC codes are then used to provide UEP. A scheme combining multi-rate embedded source and channel coding to provide speech transmission over an extended range of channel conditions was also described in [50, 51].

We combine in the first part of the dissertation embedded AMR source coding and rate-compatible channel coding to design codecs which make maximum use of the available channel bandwidth using bit-prioritized embedded source coders and a new type of rate-compatible channel encoder [52].

1.7 Remote speech recognition over error-prone channels

In remote speech recognition applications (sometimes also referred to as distributed speech recognition or DSR), the server performs the complex task of speech recognition and transmits the recognized information back to the client. This enables low power/complexity devices to be speech recognition-enabled at low cost. Applications for remote speech recognition include voice activated web portals, menu browsing, voice-operated personal digital assistants (PDAs) or alternatives to keyboards for the next generation of pocket PCs.

Figure 1.6 illustrates the general block diagram for remote speech recognition, with the client (top branch) separated from the server (lower branch) by the communication channel, typically wireless or packet-based. The goal is to provide high recognition accuracy over a wide range of channel conditions with low bit rate, delay and complexity for the client. We investigate source coding, channel coding, channel decoding, frame erasure concealment and speech recognition techniques suitable for remote speech recognition systems over error prone

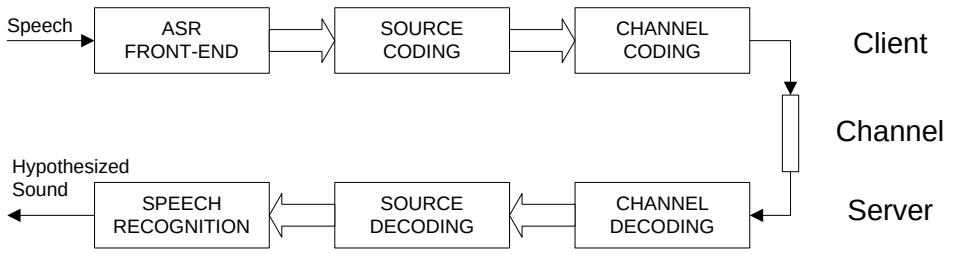


Figure 1.6: Block diagram of a remote speech recognition system.

channels.

Wireless communications is a challenging environment for remote speech recognition. The communication link is characterized by time-varying, and sometimes low signal-to-noise ratio (SNR), channels. Previous studies have suggested alleviating the effect of channel errors by adapting acoustic models [53] and automatic speech recognition (ASR) front-ends [54] to different channel conditions, or by modeling GSM noise and holes [55]. Other studies analyzed the effect of random and burst errors in the GSM bitstream for remote speech recognition applications [56, 57]. Finally, [58] and [59] evaluate the reliability of the decoded feature to provide robustness against channel errors.

Similarly, packet switched communication networks also constitute a difficult environment for distributed speech recognition (DSR) applications. The communication link in IP based systems is characterized by packet losses, mainly due to congestion at routers. Packet loss recovery techniques including silence substitution, noise substitution, repetition and interpolation are presented in [60, 61, 62].

To our knowledge, the effect of channel transmission on remote recognition systems based on quantized ASR features is a topic not yet extensively covered in the literature. Hence, our analysis and the proposed techniques present a significant improvement toward gaining robustness against channel noise.

1.8 Dissertation road map

Figure 1.7 is a road map of the dissertation, showing the inter-dependency of the chapters, and suggests that this work can be conceptually divided into four parts: source coding (left) and channel coding (right) for two types of applications, adaptive-multirate (AMR) speech transmission (top) and remote speech recognition (bottom).

Chapter 1 provides the reader with the concepts of digital speech processing and digital communications that will be used throughout this dissertation.

Chapters 2 and 3, respectively, present source and channel coding techniques that will be used in the design of spectrum efficient AMR transmission systems, presented in Chapter 4.

Chapters 5–7 are concerned with the development of source and channel coding solutions for low bit rate remote speech recognition systems. Chapter 5 presents solutions for quantizing ASR features. Chapter 6 analyzes the effect of channel errors on recognition accuracy and presents channel coding and decoding techniques adapted to remote recognition. Chapter 7 introduces techniques alleviating the effect of channel erasures on recognition accuracy, and combines source and channel coding to create channel-robust low bit rate systems. Readers may go directly to Chapter 5 if the design of remote speech recognition systems is of primary interest. Finally, Chapter 8 presents a summary of the dissertation, recapitulates the contributions made and provides future directions for research.

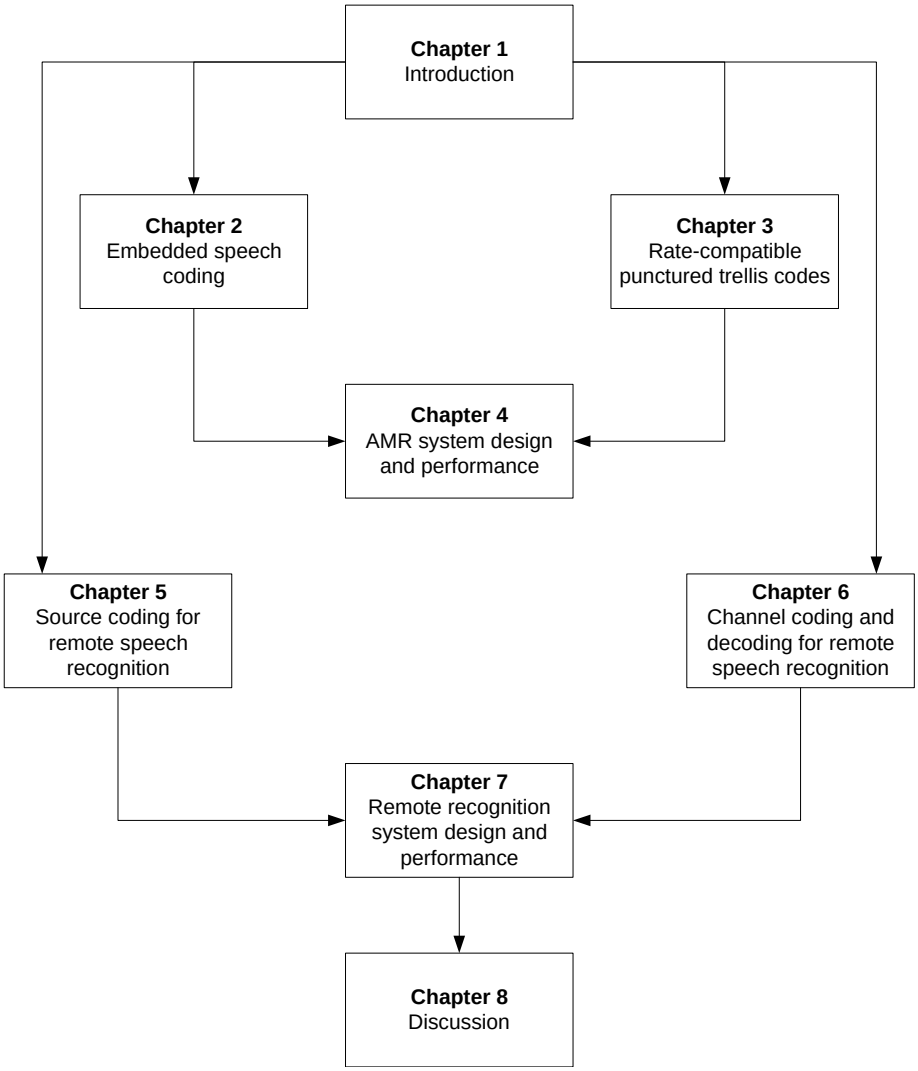


Figure 1.7: Dissertation road map.

Part I

Speech transmission using rate-compatible trellis codes and embedded source coding

Part I of the dissertation is organized as follows. Chapter 2 describes a perceptually-based embedded subband coder and the G.727 embedded ADPCM coding standard, and analyzes their bit error sensitivities against transmission errors. Chapter 3 introduces rate-compatible punctured trellis codes (RCPT) as a new tool for providing efficient rate-compatible unequal error protection with large constellation sizes. A code design strategy for RCPT is given and its performance in comparison with RCPC and RCPC-BICM codes is presented. Chapter 4 presents the design of an AMR source-channel coding scheme for the embedded subband and G.727 ADPCM coders of Chapter 2.

CHAPTER 2

Embedded speech coding

This chapter presents two types of speech coders whose properties (variable bit rate and embedibility) are useful for implementing embedded adaptive multi-rate speech transmission systems. The first coder is a perceptually based subband coder (Section 2.2) and the second coder is the ITU embedded ADPCM G.727 standard (Section 2.3). Both coders are analyzed to illustrate how AMR systems (Chapter 4) can be based on a variety of source coding schemes. Characteristics displayed by these two coders are introduced in Section 2.1.

2.1 Desired properties for speech coding

Variable bit rate coding: In variable bit rate (VBR) coding [41], different speech segments are encoded at different bit rates. Variable rate coding can be effective for storage, packet voice, multiple access channel applications and transmission over time varying channels. The two typical approaches to VBR coding are in the variable update rates and the variable bit allocations for the parameters of different speech segments. In VBR coding, a tradeoff is made between speech quality and the coding bit rate.

Embedibility: Embedded source coding allows for partial reconstruction of the source at higher distortion, given that only a portion of the bitstream is available at the decoder. It is advantageous since the source rate can be modified simply by truncating the bitstream. The distinction between VBR and embedded source coding is that embedded schemes have the capability of bit dropping outside the encoder and decoder blocks. This allows for bit reductions at any point in the network without the need of coordination between the transmitter and the receiver.

Perceptually-based coding: Source compression (image, speech or audio) employs two intrinsic properties of the original signal. On one hand, the signal contains some redundancies; on the other hand, not all the signal is relevant to the human ear or eye. Reduction of the redundancy or mutual correlation is the purpose of, for instance, any linear predictor or vector quantizer [63]. Reduction of the information content of the signal by dropping the "irrelevant" part of the signal can only be made with the use of some human auditory or visual perception model (Section 1.2).

In perceptually-based speech and audio coding [64], the human auditory perception model is used and the irrelevant signal information is identified during signal analysis by incorporating several psychoacoustic principles, such as absolute hearing thresholds, masking and critical band frequency analysis.

Dynamic bit allocation: The concept of dynamic bit allocation was introduced by Ramstad in 1982, in the context of a subband coder [65]. The bit allocation scheme presented was based on the energy of different subbands, and the number of bits allocated to each subband was directly proportional to the energy of the given subband.

Dynamic bit allocation for a perceptual coder is a two-tiered process. First, an algorithm estimating the relative importance of each band is used. The metric used for the perceptual importance of each band is the signal-to-mask ratio (SMR) presented in Section 1.2. Then, the bit allocator determines the bit allocation according to the different SMR values and overall total target bit rate.

If the number of bits required by the SMR values is larger than the number of bits that can be allocated given the target bit rate, some sub-optimal bit allocation is needed. Two strategies are the reverse water-filling bit allocation and proportional bit allocation [66]. However, reverse water-filling may not guarantee a minimal bit allocation for the bands with the lowest, yet significant, SMRs. The proportional bit allocation scheme is chosen as the one minimizing the effect of a non-sufficient bit allocation due to the overall bit rate constraint.

Dynamic perceptual bit prioritization: The dynamic bit allocation scheme can be easily modified to provide perceptual bit prioritization. For every speech frame, the bit allocator assigns different numbers of bits for the representation of different parameters or subbands in such a way that the perceptually more important parameters or subbands are quantized with more bits. Assume now that we start with no bit being allocated and that the bit allocator can only allocate one bit. If the bit allocator works properly, the first bit allocated is likely to be the perceptually most relevant one and therefore the most sensitive to channel errors. If one maintains the same progressive allocation, the order with which the bits are allocated is assumed to be a good ordering of bits in the bitstream with decreasing bit error sensitivities.

This ordering is dynamic since it is not always the case that the same bits representing the same parameters or subbands remain the most important. This

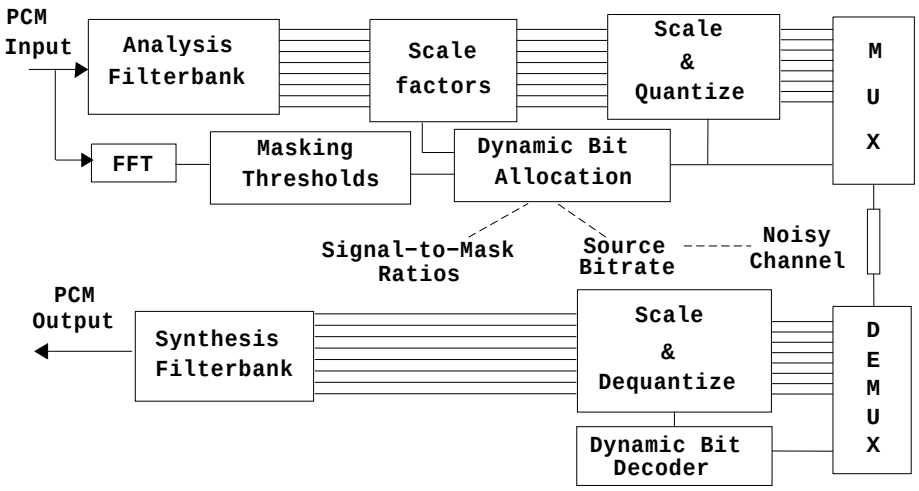


Figure 2.1: Block diagram of the perceptually-based subband speech codec.

prioritization is obviously perceptually-based.

2.2 Perceptually-based embedded subband speech coder

2.2.1 Description of the coder

The embedded subband coder shown in Figure 2.1 is a modified version of the coder presented by Tang *et al.* in [66, 67]. The speech is first divided into 20 ms frames. An Infinite Impulse Response (IIR) Quadrature Mirror Filterbank (QMF) [68] divides each frame into 8 equal subbands that are then individually encoded [69]. For each frame, dynamic bit allocation according to the perceptual importance of each subband is performed. The MPEG psycho-acoustic model [17] estimates the signal to mask ratio (SMR) required in each band to mask the quantization noise. The dynamic bit allocation (which is the side-information of the coder and is transmitted with the coded bits) translates the SMR prescribed by the model into a bit assignment to scalar quantize the subband samples.

Dynamic bit allocation based on the perceptual characteristics of the signal has two advantages: it minimizes audible distortion by shaping the quantization noise with respect to the speech spectrum, and it allows the same coder to operate at different bit rates. In the case of the subband coder, dynamic bit allocation is progressive and allocates bits of high perceptual importance first and the ones with the least perceptual importance last. This provides a tool for bit prioritization, necessary for UEP.

Figure 2.2 shows an example of progressive bit allocation for the case of a coder operating at 32 kbps for a 4 kHz wide speech signal. Each frame is composed of 160 samples, divided into 8 subbands with 20 samples per subband. Each block shown in Figure 2.2 represents the allocation of one bit to all 20 samples in a subband (1 block=1 kbps). The first 3 blocks (3 kbps) are dedicated to the transmission of the bit allocation (3 bits/band), the frame gain (4 bits) and the bands' gains (4 bits/band), for a total of 60 bits per frame. The allocated bitstream is prioritized using 20-bit segments, selecting the blocks in Figure 2.2 from top to bottom and from left to right. The allocation order of each block is indicated by the number in its center. The coder, robust against acoustic noise, offers embedded variable bit rate source coding with reasonable to excellent speech quality in the range of 8–32 kbps.

2.2.2 Bit error sensitivity analysis

In Chapter 3, we will show how rate-compatible punctured trellis codes can be used in order to provide UEP. For this purpose, we derive the maximum BER tolerable for each bit in the bitstream below which the effect of channel errors is inaudible.

The notion of determining the relative importance of bits for further UEP

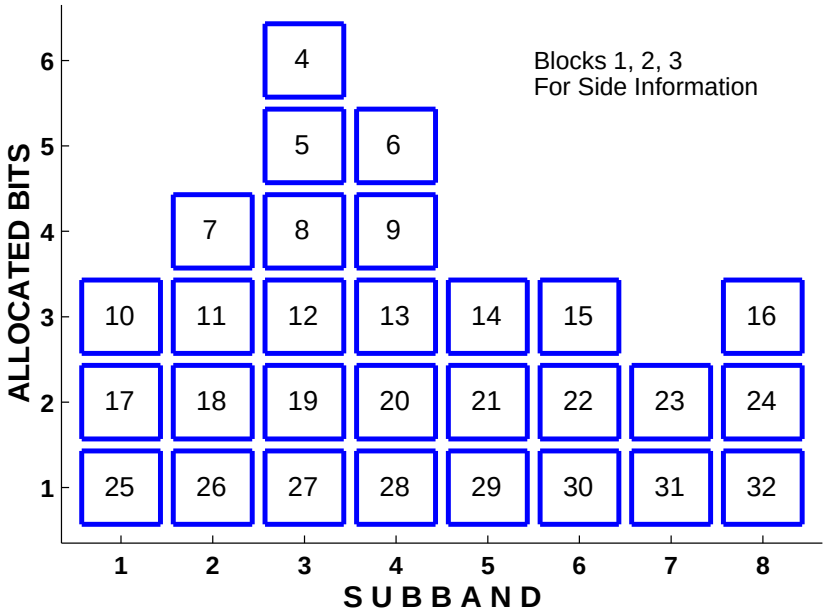


Figure 2.2: Example of bit allocation and bit prioritization for the subband coder operating at 32 kbps. Each block represents the allocation of one bit to each subband sample (1 kbps). The first three blocks (3 kbps) are reserved for the transmission of the side information (bit allocation and the different gains). The priority of each block is indicated by the number in its center. Note that the coder operating at m kbps would consist of the first m allocated blocks.

was pioneered by Rydbeck and Sundberg [70, 71]. One can define the bit error sensitivity (BES) of a given bit in the bitstream as the relative increase in speech distortion due to transmission errors at that particular bit position. Typically, BES is computed by measuring the segmental SNR after setting bits in errors [70].

In the perceptually-based subband coding scheme, the signal-to-mask ratio (SMR) of each subband is computed. The SMR indicates the perceptual importance of each band. We refine the BES analysis by computing the increase in speech distortion due to setting bits in error at different BERs using a distortion metric that takes into account the masking properties of auditory perception. We define the perceptual spectral distortion (SD_P) measure between the original spectrum $A(f)$ and the reconstructed spectrum $\hat{A}(f)$ as follows:

$$SD_p(\hat{A}(f), A(f)) = \sqrt{\sum_{i=1}^{N} \text{SMR(i)} \int_{f=f_i^l}^{f=f_i^u} 10 \log \frac{|\hat{A}(f)|^2}{|A(f)|^2} \, df}, \qquad (2.1)$$

where N is the number of bands and f_i^l, f_i^u, and SMR(i) represent the lower frequency, the upper frequency, and the weighting function, respectively, for the i^{th} subband. SMR(i) is defined as $\text{SMR(i)} = \max_{f \in [f_i^l, f_i^u]} SMR(f)$.

Figure 2.3 illustrates the BES for the coder operating at its maximum rate, *i.e.* 32 kbps. The sensitivity of each block against channel errors is computed by averaging the BES of the 20 bits in that block. Individual BES are estimated by systematically setting for each frame the particular bit position in error with a probability of error equal to the BER of interest and keeping all other bit positions error-free. Speech material used consists of 8 English sentences (4 male and 4 female talkers) from the TIMIT [72] database. The dotted horizontal line represents the maximum tolerable distortion due to channel errors. At this distortion level, informal listening tests indicated that speech distortion introduced by channel impairment is practically transparent to the listener, in the sense that

an increment in distortion due to channel transmission is indistinguishable from the distortion introduced by the source coder.

Figure 2.3 shows that the sensitivity to transmission errors of the first three blocks is very high even for BER=10^{-3}, and may be beyond scale for larger BERs. Those blocks correspond to the side information (bit allocation and band gains) and forward error correcting codes should assure that they are sufficiently protected. A second observation is that at this rate, the last bits in the bitstream are relatively insensitive to channel errors. Even with a BER as high as 10^{-1}, distortion is below the sensitivity threshold. These bits barely need protection against channel impairment. Finally, observe that the almost monotically decreasing nature of Figure 2.3 justifies *a posteriori* the perceptual and dynamic bit allocation algorithm.

2.3 Embedded ADPCM G.727 speech coder

The ITU G.727 embedded ADPCM standard coder [73] is one of the few speech coding standards that provides both variable bit rate and embedibility. It is constructive to consider AMR system design using the ADPCM coder to verify the applicability of the scheme to an existing standard and a non-perceptually-based speech coder.

2.3.1 Description of the coder

Embedded ADPCM algorithms are a family of variable bit rate coding algorithms operating on a sample-by-sample basis that allow for bit dropping after encoding. As with the subband coder, the decision levels of the lower rate quantizers are subsets of those of the quantizers at higher rates. This allows for bit reduc-

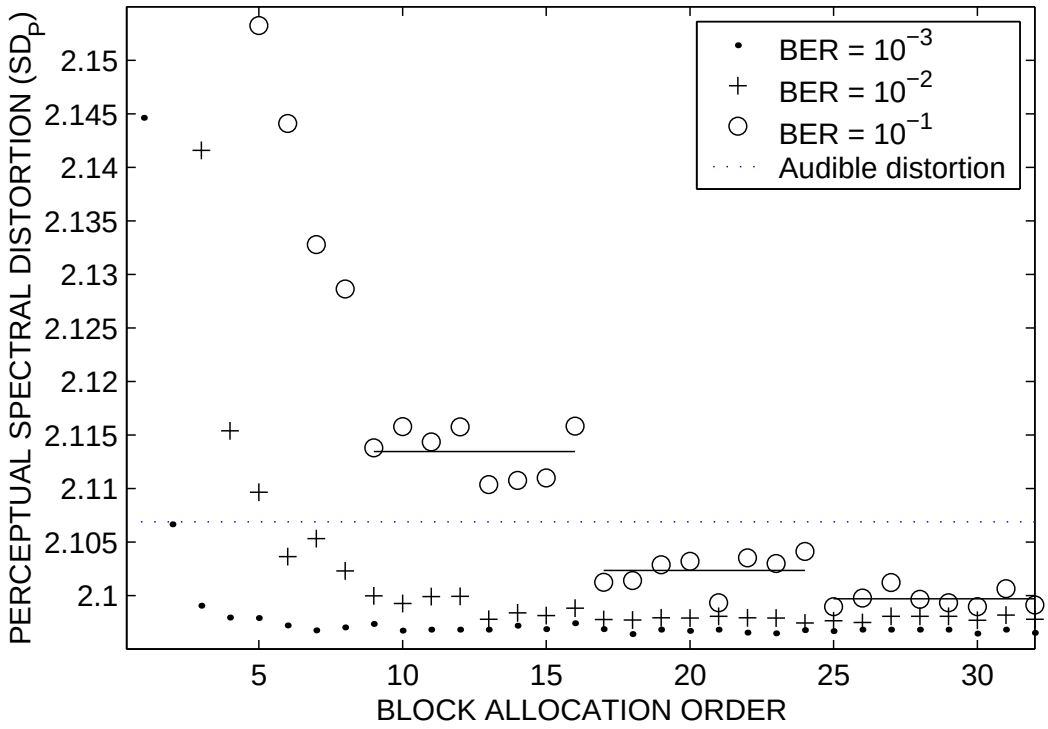


Figure 2.3: Bit error sensitivity analysis of the perceptually-based subband coder operating at 32 kbps. Note that sensitivities tend to reach plateaus of 8 blocks, which typically correspond to the allocation of one block to each subband. Eight English sentences are used to generate these plots.

tion at any point in the network without the need for coordination between the transmitter and the receiver.

Simplified block diagrams of the G.727 embedded ADPCM encoder [74] and decoder are shown in Figure 2.4. Embedded ADPCM algorithms produce code words that contain enhancement and core bits. The feed-forward (FF) path of the codec utilizes both enhancement bits and core bits, while the feed-back (FB) path uses core bits only. With this structure, enhancement bits can be discarded or dropped during network congestion. Embedded ADPCM algorithms are referred to by (n, m) pairs where n refers to the FF (enhancement and core) bits and m refers to the FB (core) bits. For example, the (5,2) coder operates at 40 kbps (5 bits/sample) while the (4,2), (3,2) and (2,2) pairs represent the 32 kbps, 24 kbps and 16 kbps algorithms, respectively, embedded in the 40 kbps coder.

2.3.2 Bit error sensitivity analysis

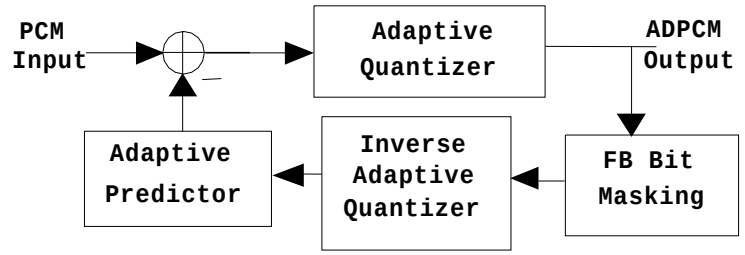
ADPCM coders do not provide any SMR information. In this case, we use the spectral distortion metric (SD) introduced in [75],

$$SD(\hat{A}(f), A(f)) = \sqrt{\frac{1}{W_0}\int |W_B(f)|^2 10 \log \frac{\left|\hat{A}(f)\right|^2}{|A(f)|^2}\, df}, \tag{2.2}$$

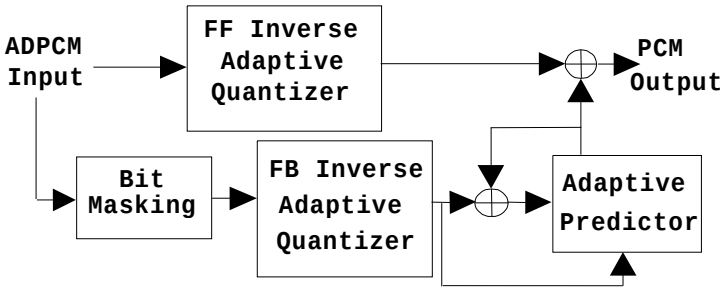
where $A(f)$ and $\hat{A}(f)$ are the original and quantized speech spectra, W_0 is a normalization constant and W_B is a hearing sensitivity weighting function defined by

$$W_B(f) = \frac{1}{25 + 75(1 + 1.4(f/1000)^2)^{0.69}}. \tag{2.3}$$

Figure 2.5 illustrates the effect of transmission errors on the 5 different bit positions in a 5 bits/sample ADPCM encoder (5,2). Also represented in Figure 2.5 is the level of distortion for which the incremental distortion introduced



(a) Encoder



(b) Decoder

Figure 2.4: Simplified diagrams of the embedded ADPCM G.727: (a) encoder and (b) decoder.

by channel impairment is inaudible. This result is obtained by informal listening tests. In the (5,2) ADPCM encoding pair, the two first bits (FB bits) are fed-back into the adaptive predictor, resulting in error propagation. Therefore, their sensitivity to channel inaccuracy is high. The three last bits (FF bits) are less sensitive to transmission errors, and tolerate transmission error rates up to around 10^{-2} (in contrast with the perceptual SBC coder at 32 kbps which could tolerate BERs up to 10^{-1}).

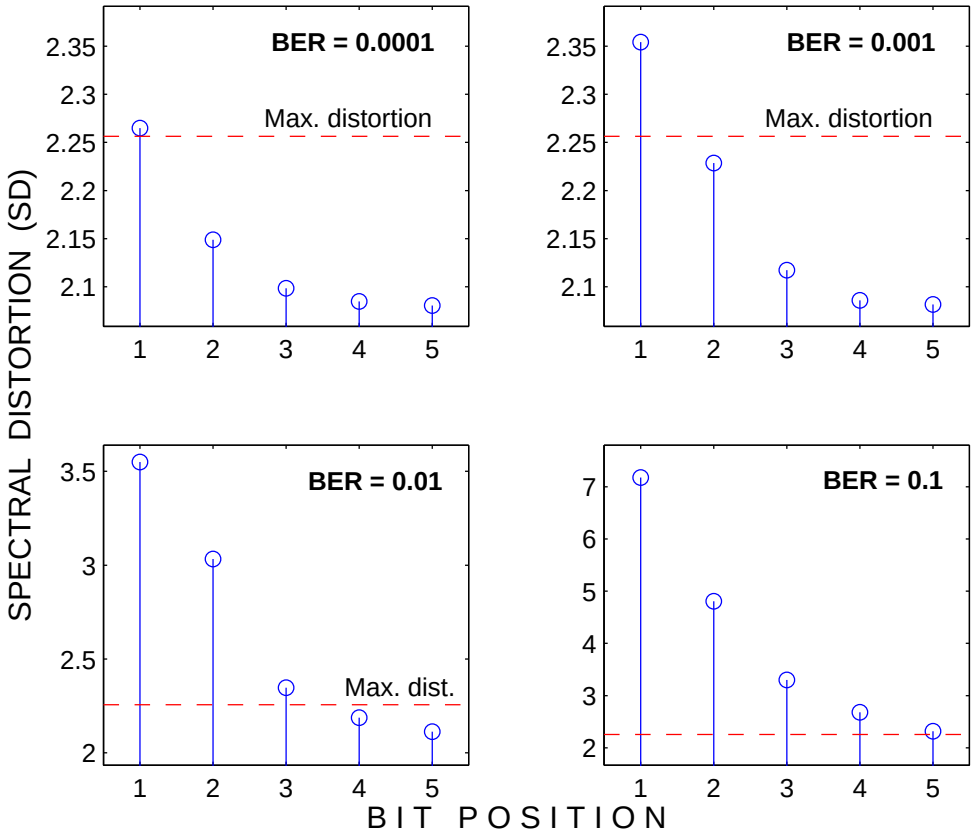


Figure 2.5: Bit error sensitivity analysis for the embedded ADPCM (5,2) speech coder operating at 40 kbps (5 bits/sample). Bit error rates analyzed range from 10^{-4} to 10^{-1}.

2.4 Summary

This chapter presented two types of speech coders whose properties (variable bit rate and embedibility) are useful for implementing embedded adaptive multi-rate speech transmission systems.

The first coder is a perceptually-based subband coder which incorporates knowledge of the human auditory system to produce a bandwidth efficient bitstream with a very wide range of perceptual sensitivities to channel errors.

The subband coder is compared to an existing standard (the ITU G.727 embedded ADPCM standard) to analyze how unequal error protection can also be applied to non-perceptually based embedded speech coders.

The bit error sensitivities of both coders are evaluated. This information is used in the design of bandwidth efficient AMR schemes in order to apply adequate channel protection to each part of the bitstream (unequal error protection) by means of rate-compatible channel encoders, which will be presented in Chapter 3.

Reliability of the overall scheme will be guaranteed over a wide range of channel conditions by dynamically allocating bits between source and channel coding depending on channel conditions, as will be seen in Chapter 4.

CHAPTER 3

Rate-compatible punctured trellis codes

In this chapter, we present a novel unequal error protection channel encoding scheme by analyzing how puncturing of symbols in a trellis and the rate-compatibility constraint (progressive puncturing pattern) can be used to derive rate-compatible punctured trellis codes (RCPT). While conceptually similar to RCPC codes, RCPT codes are specifically designed to operate efficiently on large constellations (for which Euclidean and Hamming distances are no longer equivalent) by maximizing the residual Euclidean distance after symbol puncturing. Large constellation sizes, in turn, lead to higher throughput and spectral efficiency on high SNR channels.

3.1 RCPC, RCPT and RCPC-BICM codes

Punctured convolutional codes were introduced in 1974 by Cain, Clark and Geist [76], mainly as a lower complexity alternative to high rate convolutional coding. Punctured convolutional codes reduce the complexity of the decoder for high rate codes [77].

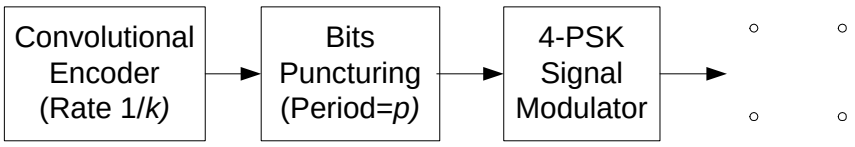
Hagenauer added the rate-compatibility restriction to derive the concept of

Rate Compatible Punctured Convolutional (RCPC) codes as a special case of punctured convolutional codes [47, 78, 79]. Rate compatibility arises when more severe puncturing can only be obtained by puncturing bits that were not punctured at the lower rate code. This means that all the bits of the high rate code are used by the low rate code, or equivalently, once a bit is punctured for a given puncturing, it must also be punctured for a more severe puncturing level. For convolutional codes, it was shown that rate-compatible codes can be as good as the best known conventional codes of the same constraint length [47].

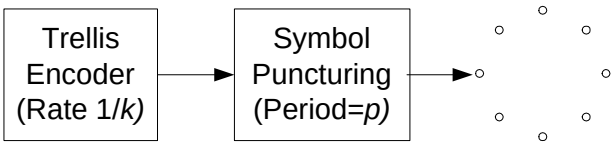
In RCPC (Figure 3.1(a)), the effect of periodic puncturing is to remove periodic subsequences of *bits* before signal mapping and transmission. As the number of punctured bits increases, the information rate per *bit* of the convolutional encoder increases and its performance degrades.

In the proposed rate-compatible punctured trellis codes (RCPT) (Figure 3.1(b)), *symbol*-wise periodic puncturing of trellis codes, introduced in [80, 81], provides an alternative to bit puncturing. If progressive symbol puncturing is required, rate-compatible punctured trellis codes are obtained. The effect of periodically puncturing bits or symbols is to remove periodic subsequences of bits (before signal mapping) or constellation points, respectively, before transmission. As the number of punctured bits or symbols increases, the information rate per transmitted symbol increases and the BER performance of the code degrades.

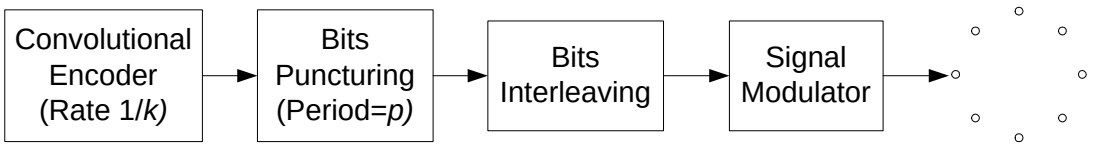
With rate-compatible punctured codes, all codes, except the one with the lowest rate, are derived by puncturing bits from the convolutional encoder (RCPC) or symbols from the trellis encoder (RCPT) with the lowest rate. Rate-compatibility also provides embedibility in the bit or symbol stream. One of the main advantages of rate-compatible codes is that they allow the use of the same decoding structure for multiple code rates, since the decoding trellis remains unchanged



(a) RCPC



(b) RCPT



(c) RCPC-BICM

Figure 3.1: Schematic representation of the different rate-compatible punctured encoding schemes: (a) RCPC, (b) RCPT and (c) RCPC-BICM.

through puncturing. Only the branch metrics in the Viterbi decoder need to be updated depending on the puncturing pattern.

We also introduce a third class of rate-compatible punctured codes, based on a bit-interleaved version of RCPC codes, which can operate on large constellation size even when using RCPC codes. The scheme combines Hamming distance based convolutional codes with large constellation sizes by introducing bit-interleaved coded modulation (BICM) between the convolutional encoder and the signal mapper to guarantee that successive bits in the bitstream (and mapped into symbols) are independent from one another. However, since the scope of this chapter is to design rate-compatible channel coders providing unequal error protection, we modify BICM in such a way as to make the coder rate-compatible

using progressive puncturing of bits after convolutional encoding. The resulting coder is a rate compatible punctured bit-interleaved convolutional code, referred to as RCPC-BICM, and illustrated in Figure 3.1(c). Operating on any types of constellation, the rates of RCPC-BICM and RCPT codes can be compared.

In [82] and [47], Lee and Hagenauer present convolutional codes and rate-compatible puncturing patterns leading to good RCPC codes. In the next section, we discuss the design of a trellis code and the selection of progressive puncturing patterns defining efficient RCPT codes. These codes will be compared to RCPC and RCPC-BICM.

3.2 RCPT code design

With RCPT, a puncturing pattern that removes q symbols out of every p symbols (p is the puncturing period) is a $p-q$ puncturing pattern. The average per-symbol information rate R associated with a $p-q$ puncturing pattern applied to a rate k/n code and a 2^n constellation size is given by

$$R = \frac{pk}{p-q}, \tag{3.1}$$

where $0 \leq q < p$.

In a sequence of progressive puncturing patterns, let $\tilde{a}^q$, a vector of p binary elements, be a pattern with q punctured symbols. A "1" in $\tilde{a}^q$ indicates that the symbol is transmitted, and a "0" indicates that the symbol is not transmitted (punctured). To provide rate-compatibility, once a symbol is punctured at a given rate, it must also be punctured at any higher rate; *i.e.* $\tilde{a}^{q+1}$ can only be formed by replacing one remaining "1" of $\tilde{a}^q$ with a "0". Note that in order to avoid negative redundancy ($R > n$), one must satisfy $q \leq \lfloor p(1-k/n) \rfloor$.

The trellis used in the soft Viterbi decoding of the received symbols has the

same structure throughout all of the puncturing patterns. Puncturing any symbol before transmission can be represented in the receiver by setting all branch metrics associated with the corresponding non-received symbol to zero. The same decoder can be used with all coding rates, and the rate can change during decoding, as with RCPC.

RCPT codes are a particular case of the symbol-punctured trellis codes for periodic erasures introduced in [80, 81]. For RCPT codes, the puncturing vector $\tilde{a}^q$ must also satisfy the condition for rate-compatibility, *i.e.* progressive puncturing.

The periodic distance vector for trellis codes is first defined. Let the normalized symbol-wise squared Euclidean distance between the i^{th} symbols of two trellis events be $d_i^2(x \rightarrow \hat{x}) = (\hat{x}_i - x_i)^2/\varepsilon_x$, where x_i and $\hat{x}_i$ are the correct and incorrect constellation points associated with the i^{th} symbols of a trellis error event, respectively, and ε_x is the average constellation energy. The periodic puncturing of symbols scales the distances with the same index modulo p by the binary scale factor $\tilde{a}_i$. Define the periodic squared distance $\tilde{d}_i^2$ for any given index i and puncturing period p as the sum of the square of the distances scaled by the same factor a_i,

$$\tilde{d}_i^2(x \rightarrow \hat{x}) = \sum\nolimits_{m=0}^{\infty} d_{i+mp}^2. \tag{3.2}$$

The p values of $\tilde{d}^2 = [\tilde{d}_1^2 \cdots \tilde{d}_p^2]$ form the periodic distance vector.

The usual criterion for minimizing the BER for trellis codes under AWGN is a large free Euclidean distance. The minimum Euclidean distance remaining after puncturing q symbols out of every p symbols ($p - q$ puncturing) using the puncturing pattern $\tilde{a}^q$ is referred to as the residual Euclidean distance RED_q, and is computed as an inner product:

$$RED_q(\tilde{a}(q)) = \min_{\tilde{d}^2} \left\langle \tilde{a}(q)^2, \tilde{d}^2 \right\rangle. \tag{3.3}$$

For a specified $\tilde{a}$, the pairwise error probability for the trellis error event $x \rightarrow \hat{x}$ is

$$P(x \rightarrow \hat{x}) = Q\left(\sqrt{\frac{\varepsilon_x \langle \tilde{a}^2, \tilde{d}^2 \rangle}{2N_0}}\right), \tag{3.4}$$

where $Q(t) = \frac{1}{\sqrt{2\pi}} \int_t^\infty e^{-u^2/2}\mathrm{du}$ and $\tilde{d}^2$ is the periodic distance vector for $x \rightarrow \hat{x}$.

Note that if two output sequences after puncturing are identical, the RED is zero. Note also that when designing a trellis code for periodic symbol puncturing, the necessary condition for rate-compatibility limits the number of puncturing families $\{\tilde{a}\}$ to consider, where $\{\tilde{a}\}$ is the set of all puncturing families.

In practice, finding the best code and puncturing patterns to minimize the BER under different puncturing patterns would require extensive simulations, or at least a union bounding. However, asymptotic coding gains for trellis codes are linear in the minimum Euclidean distance of the code expressed in dB. Thus, RED is a good (but not exact) indicator of BER under puncturing.

RCPT code design is a multi-criterion problem since we have to minimize the BER (maximize RED) at all rates (puncturing patterns) simultaneously. The best performance for a particular puncturing level will often be obtained at the expense of suboptimal performance for another puncturing level. We refer to a trellis code as undominated if no trellis codes of the same complexity performs better on every channel in the family. Typically, there will be several undominated trellis codes. Such undominated solutions are called Pareto optimal. To select among the Pareto optimal codes, we choose equal weighting of asymptotic coding gains as a sensible way to resolve the multi-criterion problem. The design criterion is thus the maximization of J_{dB}, the logarithmic sum of all RED values of interest,

$$J_{dB} = \sum_{q=0}^{\left\lfloor p(1-\frac{k}{n}) \right\rfloor} 10 \log_{10}(RED_q). \tag{3.5}$$

The limits of the summation represent the puncturing patterns of interest, which range from no puncturing, $q = 0$, to puncturing all redundancy added by the channel encoder, $q = p\ (1 - k/n)$. We emphasize that other reasonable approaches exist to choose a final code from the set of Pareto optimal codes. This objective function gives equal weight to the asymptotic SNR requirements of each puncturing pattern. We ran an exhaustive search with this objective function to find the best candidate over all Pareto optimal codes and progressive puncturing families.

The Viterbi decoding complexity of a trellis code depends both on the number of memory elements ν (number of states is $S = 2^{\nu}$), and on the traceback depth $\mathrm{L_D}$ of the decoding process. For standard trellis or convolutional codes, $\mathrm{L_D}$ is computed as the trellis depth at which all unmerged error events have more Euclidean distance than the minimum Euclidean distance of the trellis code [83]. The traceback depth for a specific puncturing pattern dropping q symbols, written $(\mathrm{L_D})_{\mathrm{q}}$, is the trellis depth at which all unmerged incorrect paths exceed the residual Euclidean distance $\mathrm{RED_q}$.

Catastrophic behavior occurs when an infinite number of bit errors result from a finite Euclidean distance vector event, *i.e.* the encoder state diagram has a loop that has zero output Hamming weight and nonzero input Hamming weight. Even if the original encoder is not catastrophic, periodic puncturing of symbols may lead to catastrophic behavior. A technique for determining catastrophic behavior under periodic symbol erasures is presented in [81, 84]. Our search used this technique to rule out combinations of codes and puncturing families that were catastrophic at any rate of interest. Recently, a more efficient technique for identifying catastrophic behavior was presented in [85].

Whether or not codes generally exist using our design procedure depends on

RCPT Generator Matrix = [32 11 27] $\nu = 4$, $p = 8$ symbols, rate = 1/3			
q	**Rate**	**Puncturing**	$\mathbf{RED}_q^2$
0	1.000	11111111	12.60
1	1.142	01111111	8.34
2	1.333	01110111	6.34
3	1.600	01010111	4.58
4	2.000	01010101	4.58
5	2.667	00010101	1.17

Table 3.1: Characteristics of the 8-PSK, 16-states ($\nu = 4$), rate-1/3 and period-8 RCPT codes.

whether codes exist that can be maximally punctured without becoming catastrophic. We do not have proof of the existence of such codes, but we have examined several scenarios and never found a case where such codes did not exist. As an example, a rate-1/3 RCPT code designed for an 8-PSK constellation with puncturing period $p = 8$ symbols and with 4 memory elements ($\nu = 4$ or 16 states) was found using the J_{dB} criterion and is presented in Table 3.1. The generator matrices are given in octal notation (*e.g.* 43 stands for $1+D+D^5$). Euclidean distances throughout these results assume a two-dimensional constellation with a unit average energy. As expected, RED typically decreases as more symbols are punctured, increasing the information rate.

Another example of RCPT code, designed for a 16-QAM constellation, is shown in Table 3.2. It is a rate-1/4, 64-state and period-8 code. For the curve with information rate of 1 bit per symbol, [80] shows that there is a penalty of about 1 dB at BER=10^{-5} between the RCPT code and the best known code with information rate of 1 bit per symbol (the feedforward rate-1/2 maximum

RCPT Generator Matrix = [43 175 155 103] $\nu = 6$, $p = 8$ symbols, rate = 1/4			
q	**Rate**	**Puncturing**	$\mathbf{RED}_q^2$
0	1.000	11111111	12.4
1	1.142	01111111	8
2	1.333	01110111	7.6
3	1.600	01010111	4.8
4	2.000	01010101	4.4
5	2.667	00010101	1.2

Table 3.2: Characteristics of the 16-QAM, 64-states ($\nu = 6$), rate-1/4 and period-8 RCPT codes.

Hamming distance convolutional code [G=133 171] used with Gray-labelled 4-PSK). However, [81] shows another example of a 64-state, period-5, rate-1/3 RCPT code used on an 8-PSK constellation [G= 171 46 133] where for the same information rate of one bit per symbol, there is no penalty associated with the rate compatibility constraint.

In general, punctured trellis codes are competitive with stand-alone codes for information rates of 1 and 2 bits per symbol while providing greater rate flexibility. For relatively high target BER, appropriate for speech transmission, the punctured codes are also competitive at 3 information bits per symbol.

As observed in [80] and more carefully examined in [81], the determining factor for the loss imposed on trellis code performance by a rate-compatibility constraint is constellation size. Specifically, for a rate of K bits per symbol, if the constellation is significantly larger than 2^{K+1} points, as with a 16-QAM constellation for $K = 1$, the rate-compatible code will have some performance loss

as compared to a single-rate code using a 2^{K+1}-point constellation and a standard set-partitioning code. Note that the larger constellation is required for the rate-compatible code to support higher information rates. An 8-PSK constellation represents a good tradeoff, giving a relatively wide range of rates with negligible performance loss from a set-partitioning code at $K = 1$.

For any periodic puncturing vector and channel noise variance, periodic transfer function bounds producing asymptotically tight bounds on BER can also be computed [81]. However, we used simulation results to obtain the low-SNR BER performance necessary for our system design.

3.3 Comparison of rate-compatible codes

Tables 3.3 through 3.5 compare 64-state RCPT, RCPC-BICM and RCPC codes by providing the information rate per symbol R, the puncturing vector $\tilde{a}^q$, the residual squared Euclidean distances RED^2, the traceback depth ($\mathrm{L_D}$), and the number of nearest neighbors (N) for different puncturing levels. The number of nearest neighbors is computed as the sum of minimum Euclidean distance error events starting at each phase normalized by the number of phases (*i.e.* the period p). Table 3.3 presents a 64-state, rate-1/3, 8-PSK RCPT code with period $p = 9$, describing performance and decoding complexity at each puncturing pattern in its family. Note that RCPT codes are able to operate after all redundancy has been removed (case g) with the same free Euclidean distance as uncoded transmission, but it does not perform as well as uncoded 8-PSK modulation in simulation. This means that if there are no channel errors, the decoder is capable of recovering exactly the original bit sequence since the non-zero RED after extreme puncturing is sufficient to distinguish different trellis events.

RCPT Rate = 1/3 8-PSK constellation G = [165 142 127] p = 9 symbols					
ID	**Rate**	**Puncturing Vector**	**RED**^{2}	**L$_D$**	**N**
a	1.00	111111111	16	24	1
b	1.12	111111110	11.2	25	0.22
c	1.29	111101110	9.17	26	0.44
d	1.50	110101110	6.93	31	0.33
e	1.80	110101100	4.34	31	0.66
f	2.25	100101100	1.76	27	0.22
g	3.00	100101000	0.59	36	1
8	3.00	Uncoded 8-PSK	0.59	0	2

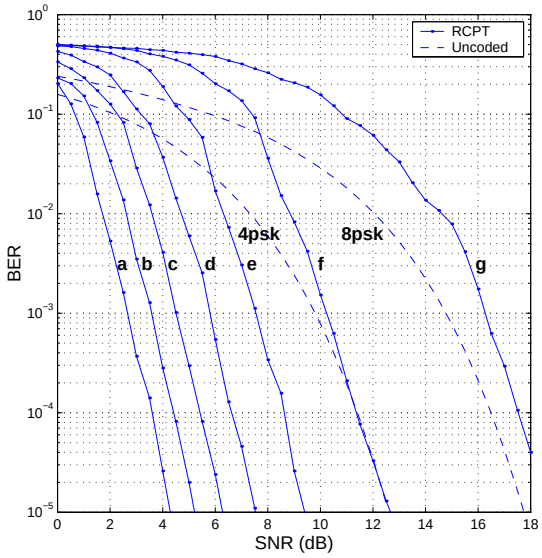
Table 3.3: Characteristics of the 8-PSK, 64-states ($\nu = 6$), rate-1/3 RCPT codes.

RCPC-BICM Rate = 1/3 8-PSK constellation G = [155 127 117] p = 3 × 9 bits					
ID	**Rate**	**Puncturing Matrix**	**RED**^{2}	**L$_D$**	**N**
A	1.00	111111111 111111111 111111111	8.79	21	3
B	1.12	111111111 111101111 111101101	7.03	22	0.22
C	1.29	111101101 111101101 111101101	5.86	21	0.33
D	1.50	101101101 101101101 101101101	5.86	28	11.1
E	1.80	101101001 101101001 101101001	4.10	34	2.33
F	2.25	101001001 001101001 101001001	2.34	45	0.22
–	3.00	Catastr.			
8	3.00	Uncoded 8-PSK	0.59	0	2

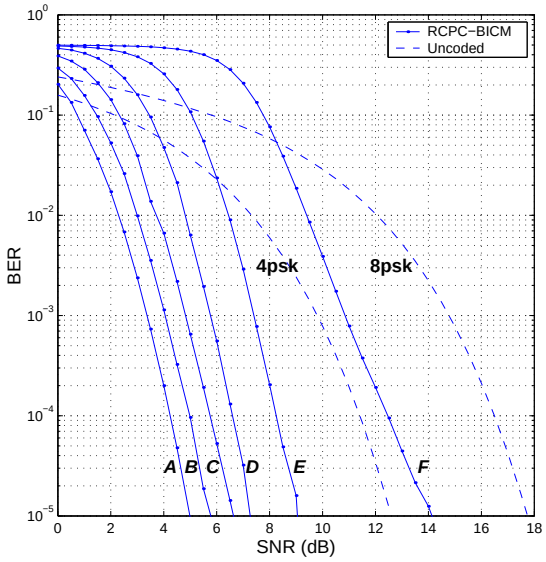
Table 3.4: Characteristics of the 8-PSK, 64-states ($\nu = 6$), rate-1/3 RCPC-BICM codes.

RCPC Rate = 1/2 4-PSK constellation G = [133 171] p = 2 × 8 bits					
ID	**Rate**	**Puncturing Matrix**	**RED**^{2}	$\mathbf{L_D}$	**N**
A	1.00	11111111 11111111	20	27	11
B	1.14	11111111 11101110	14	27	0.5
C	1.33	11111111 10101010	12	34	0.5
D	1.60	11111111 10001000	8	48	0.75
E	1.78	11110111 10001000	6	92	0.5
–	2.00	Catastr.			
4	2.00	Uncoded 4-PSK	1	0	2

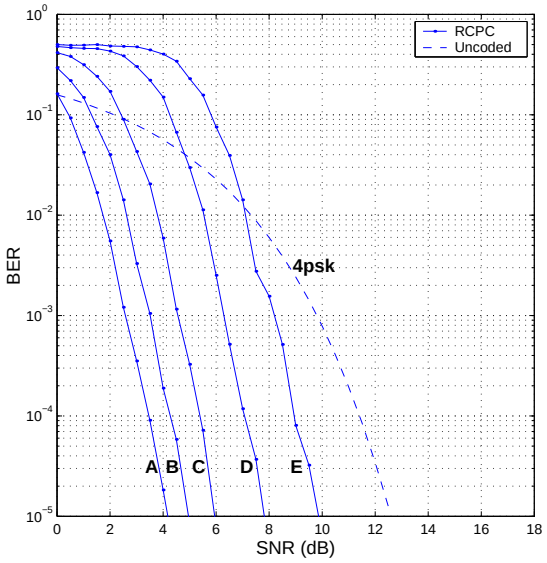
Table 3.5: Characteristics of the 4-PSK, 64-states ($\nu = 6$), rate-1/2 RCPC codes.



(a) RCPT



(b) RCPC-BICM



(c) RCPC

Figure 3.2: Bit error rate curves for the (a) RCPT, (b) RCPC-BICM and (c) RCPC encoding schemes presented in Tables 3.3–3.5 over an AWGN channel. Traceback depth used is 41.

For comparison, Table 3.5 also summarizes the performance of two 64-state RCPC codes [47]. Since RCPC codes are designed especially for Hamming distances, we use the 4-PSK constellation and consider rate-1/2 convolutional codes. The RCPC system consistently provides a slightly better Euclidean distance at a slightly higher information rate as compared to RCPT for its rate family. However, this Euclidean distance advantage appears negligible in simulations at the BERs of interest, apparently because the RCPT code has a smaller number of nearest neighbors. Furthermore, the Euclidean distance advantage of RCPC is small compared to the disadvantage of the rate limitation at high SNR imposed by the 4-PSK constellation as compared to the 8-PSK constellation used for RCPT.

We also consider bit-interleaved coded modulation (BICM) codes [86, 87] that can use Hamming-distance-based convolutional codes with any constellation. In order to use RCPC with an 8-PSK constellation, we modify BICM to make the coder rate-compatible using progressive puncturing of bits after convolutional encoding. The resulting coder is a rate compatible punctured bit-interleaved convolutional coder, referred to as RCPC-BICM. Table 3.4 shows performance of a 64-state RCPC-BICM with a rate-1/3 convolutional encoder, a puncturing period of $p = 3 \times 9$ bits and an 8-PSK constellation. The information rates are the same as for those for the RCPT codes. RCPT generally provides a better Euclidean distance than RCPC-BICM. The residual Euclidean distance for RCPC-BICM is computed from the residual Hamming distance (RHD) of the convolutional encoder after puncturing as follows:

$$RED_q^2 = RHD_q \cdot (2\sin(\pi/8))^2. \tag{3.6}$$

Raw BER versus SNR curves of the RCPT, RCPC-BICM and RCPC codes with six memory elements for a standard AWGN channel are presented in Figure 3.2. However, in the design of AMR transmission systems, we are concerned

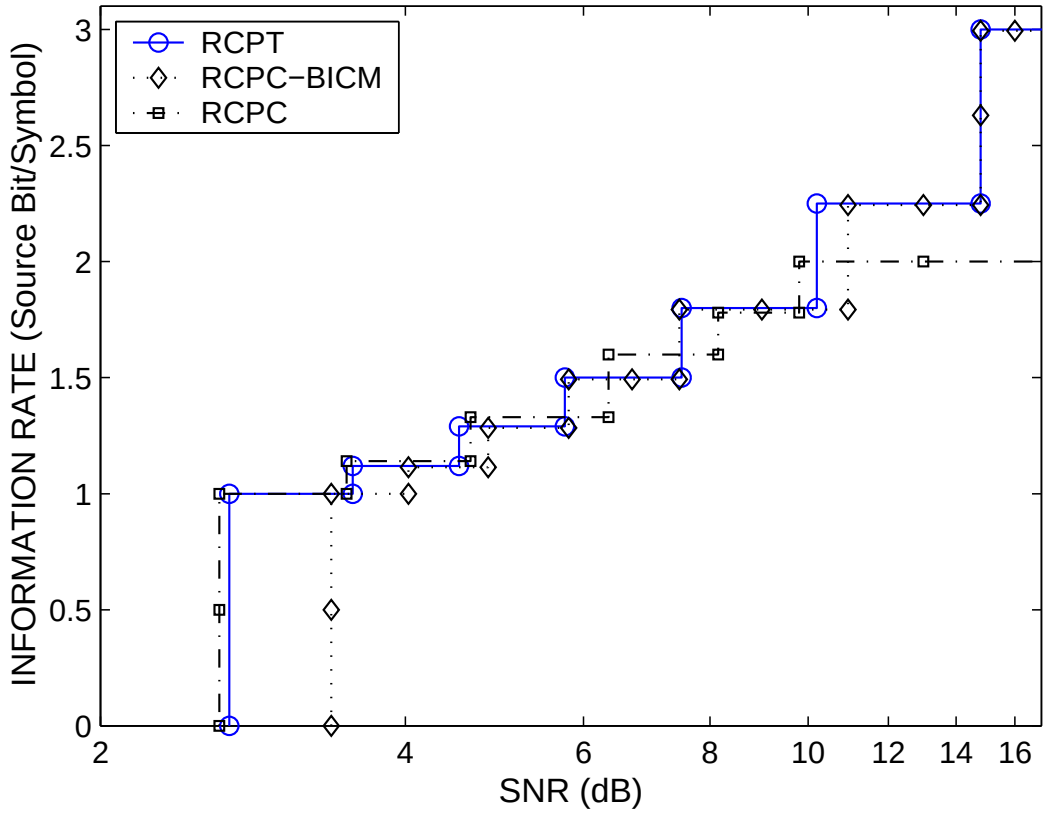


Figure 3.3: For a required BER level of 10^{-3} and an AWGN channel, the figure illustrates the achievable information rates (in source bits/transmitted symbols) for RCPT, RCPC-BICM and RCPC as a function of the channel SNR. The information rates and SNRs can also be found in Figure 3.2.

with how much source coding information can be transmitted using the different coding schemes depending on the BER requirements and the channel SNR. Figure 3.3 illustrates the different achievable information rates (source bits/transmitted symbols) as a function of the AWGN channel SNR, assuming that a BER of 10^{-3} is required. Note that the curves with rates equal to 2 and 3 are obtained using uncoded 4- and 8-PSK, respectively. Two observations are made: 1) RCPT and RCPC-BICM operate at the same rates, but RCPT outperforms since transitions to larger information rates occur at lower SNRs; 2) despite RCPC's better free Euclidean distances, RCPT and RCPC behave similarly at low SNRs. RCPT benefits at high SNR from its larger constellation size and exhibits larger information rates. Although Figure 3.3 considers only BER=10^{-3}, similar behavior is observed at different BERs. In summary, RCPT offers a wider efficient operating range than RCPC since it is specifically designed for a larger constellation, permitting larger throughput. In addition, RCPT combines coding and modulation, allowing for improved performance with respect to RCPC-BICM.

3.4 RCPT codes for fading channels

Since Ungerboeck introduced trellis coded-modulation [39], it is generally accepted that modulation and coding should be combined for improved performance. This fact is a basis for the design of RCPT codes. However, the authors of [86] showed that with bit-interleaved coded modulation, the code diversity, and hence the reliability of coded modulation over a Rayleigh fading channel, can be further improved by achieving bit-wise interleaving at the encoder output (and using an appropriate soft-decision bit metric as input to the Viterbi decoder), as opposed to doing symbol-wise interleaving. This has the effect of making the

code diversity equal to the smallest number of distinct bits rather than of distinct channel symbols along any error event.

In summary, for Rayleigh fading channels, RCPC-BICM codes would be superior since the residual diversity after puncturing would be equal to the RHD. However, this assumes ideal interleaving, which in turn requires interleaver depths of at least the coherence time of the channel. This might not be tolerable for speech coding applications where the overall transmission delay must be kept at a minimum.

The design of good RCPT codes for fading channels with no interleaver can be based on periodic versions of the fading channel design metrics of Effective Code Length (ECL) and Code Product Distance (CPD) [88], respectively called the Periodic Effective Code Length (PECL) and the Code Periodic Product Distance (CPPD) defined in terms of the periodic distance vector. These metrics first appeared in [89]. The PECL is essentially a measure of the diversity provided by the code. The CPPD measures how evenly Euclidean distances are distributed to the branches of the code [90]. The marginal pairwise sequence error probability (probability of error event) in a trellis with soft decision is bounded for high SNR by the Chernoff bound

$$P(b \to \hat{b}) \leq (\frac{\gamma^2 \varepsilon_x}{4\sigma_n^2})^{-PECL} \cdot \frac{1}{CPPD} \tag{3.7}$$

where $\varepsilon_x = E[x^2]$ is the average power of the transmitted symbol, γ^2 is the time-average power of the received signal before envelope detection and σ_n^2 is the noise variance. This probability decreases exponentially with PECL and is inversely proportional to CPPD. The reader shall refer to [90] for the computation of PECL and CPPD. As for AWGN, we again have a multi-criterion optimization problem and we refer to Section 3.2 for techniques solving the multi-criterion problem.

Tables 3.6 and 3.7 present the performance of RCPT and RCPC codes with

RCPT Generator Matrix = [32 11 27] $\nu = 4$, $p = 8$ symbols, rate = 1/3							
q	**Rate**	**Puncturing**	**RED^2**	**L_D**	**N**	**PECL**	**CPPD**
0	1.000	11111111	12.58	21	1.000	5	37.40
1	1.142	01111111	8.34	23	0.125	4	0.94
2	1.333	01110111	6.34	23	0.250	3	0.47
3	1.600	01010111	4.58	22	0.125	2	2.32
4	2.000	01010101	4.58	24	1.500	2	2.32
5	2.667	00010101	1.17	24	1.750	1	0.33

Table 3.6: Characteristics of the 8-PSK, 16-states ($\nu = 4$), rate-1/3 RCPT codes for Rayleigh fading channels.

RCPC Generator Matrix = [23 35] $\nu = 4$, $p = 2 \times 8$ bits, rate = 1/2							
q	**Rate**	**Puncturing**	**RED^2**	**L_D**	**N**	**PECL**	**CPPD**
0	1.000	11111111 11111111	14	2.000	15	5	128
2	1.125	11111111 11101110	10	0.250	15	4	32
4	1.286	11111111 10101010	8	0.500	17	4	16
6	1.500	11111111 10001000	6	2.000	39	3	8
7	1.800	11110111 10001000	4	0.125	48	2	4

Table 3.7: Characteristics of the 4-QAM, 16-states ($\nu = 4$), rate-1/2 RCPC codes for Rayleigh fading channels.

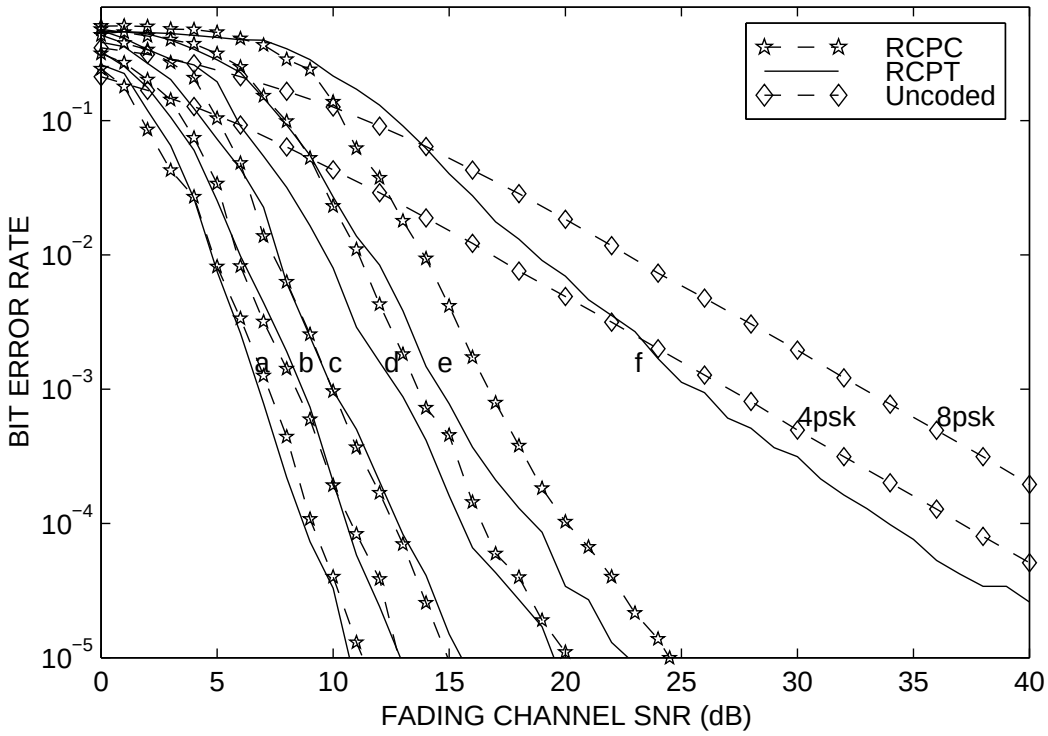


Figure 3.4: Bit error rate curves for the RCPC and RCPT encoding schemes with 4 memory elements (Tables 3.6 and 3.7) under independent Rayleigh fading channels. Traceback depth used is 128.

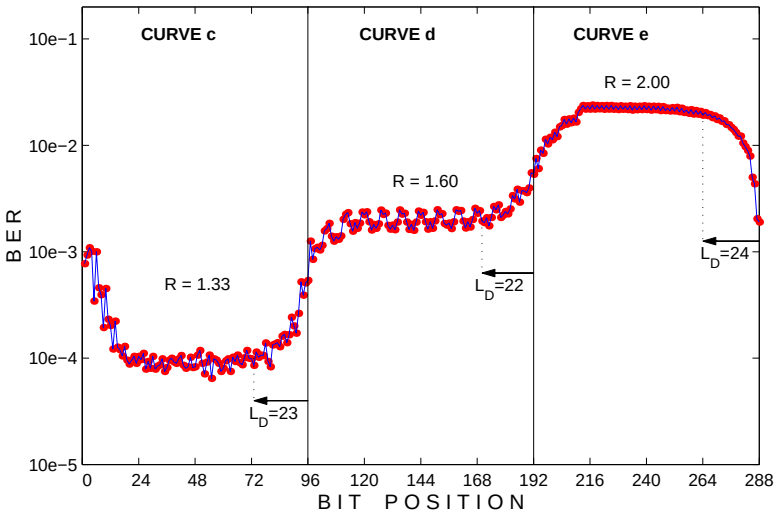
4-memory (16 states) at comparable information rates. Figure 3.4 shows the performance of these codes under Rayleigh fading channels with different channel SNRs.

3.5 Traceback depth and frame size

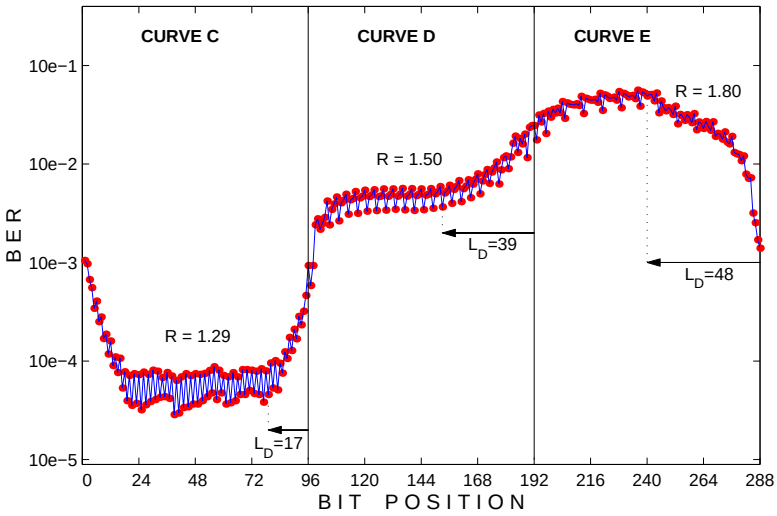
It is interesting to study the effect of frame size on the transition between RCPT code rates within a frame. Figure 3.5(a) illustrates the three levels of protection offered by the c, d, and e curves of the RCPT code of Table 3.1 operating on an AWGN channel at 7 dB and their corresponding information rates per symbol R. Figure 3.5(b) illustrates the three levels of protection offered by the C, D, and E curves of the RCPC code of Table 3.7 operating on an AWGN channel at 6 dB and their corresponding information rates per symbol R. Each code rate is used for 96 bits, which corresponds to twice the longest traceback depth.

Note that the effect of the traceback depth in the trellis is visible at transition zones. When transitioning from a $p-q$ puncturing to a $p-(q+1)$ puncturing, branch metrics degrade and performance starts degrading $(\mathrm{L_D})_{\mathrm{q}}$ bits before the transition (*i.e.* as soon as the Viterbi decoder must trace back through symbols with increased puncturing). State metrics (or path metrics) explain the behavior of the BER versus bit position curve after switching. The superior quality of the state metrics at the end of the $p-q$ puncturing pattern enhances performance at the beginning of the $p-(q+1)$ puncturing pattern, and it takes approximately another $(\mathrm{L_D})_{\mathrm{q+1}}$ symbols for the quality of these path metrics to fully degrade to the steady quality of the $p-(q+1)$ pattern. The transition zone length is then approximately $(\mathrm{L_D})_{\mathrm{q}}$+$(\mathrm{L_D})_{\mathrm{q+1}}$ symbols.

In Table 3.3, we see that RCPT codes have smaller traceback depths than



(a) Unequal error protection illustration for curves c, d, and e of the RCPT codes presented in Table 3.1. Operating AWGN channel SNR is 7 dB.



(b) Unequal error protection illustration for curves C, D, and E of the RCPC codes presented in Table 3.7. Operating AWGN channel SNR is 6 dB.

Figure 3.5: Unequal error protection illustration using (a) RCPT and (b) RCPC codes. Traceback depth used is L=96. Each level of protection is 96 bits long.

RCPC and RCPC-BICM for the most severe puncturings. This means that RCPT codes can usually operate with smaller frame sizes and buffering delays.

3.6 Summary

We presented in this chapter three different types of rate-compatible channel coding techniques (RCPC, RCPT and RCPC-BICM) allowing for unequal error protection, which can be used in conjunction with embedded speech coders to provide bandwidth efficient rate-compatible source-channel coding.

We have introduced the novel concept of rate-compatible punctured trellis (RCPT) codes whereby unequal error protection is provided by puncturing symbols in the symbolstream. Rate-compatibility was obtained by performing progressive puncturing of symbols.

We showed that in general, punctured trellis codes remain competitive with stand-alone codes for information rates of 1 and 2 bits per symbol while providing greater rate flexibility.

RCPT codes were designed in order to maximize the Euclidean distance between trellis error events at each puncturing level, whereas RCPC codes maximize Hamming distance at each puncturing level. This allows for the RCPT codes to operate (without bit interleaving) on larger constellation sizes. At high channel SNRs, the larger information throughput offered by the increased constellation sizes can provide higher speech quality.

Compared to RCPC codes with bit-interleaved coded modulation (RCPC-BICM) which also operate on large constellations, it was shown that RCPT codes, which combine modulation and coding, have a better performance.

CHAPTER 4

AMR system design and performance

This chapter presents the design procedure for embedded adaptive multi-rate speech communication systems combining variable bit rate embedded source coding and rate-compatible channel coding. Sections 4.1 and 4.2 present the AMR scheme for the perceptually-based embedded source coder of Section 2.2 and the embedded ADPCM coder of Section 2.3, respectively.

4.1 AMR system design for the subband speech coder

We design a source-channel coding system that leads to high speech quality over a wide range of channel conditions in three steps. First, for each part of the bitstream and for every SNR, rates of protection needed to obtain BERs that have corresponding inaudible distortions are determined. Second, we determine the maximum source coding bit rate that can satisfy these BER conditions given the average redundancy inferred by the rates of protection required. Finally, the puncturing architecture of the coded bitstream is derived so that the final source-channel coded bitstream equals 20 kbps for a 4-PSK RCPC scheme or 30 kbps for both 8-PSK RCPT and RCPC-BICM schemes (*i.e.* 10 kbaud/s for

Rate [kbps]	**bits/ frame**	**RCPT**	**RCPC-BICM**	**RCPC**
10	200	a_{200}	A_{200}	A_{200}
12	240	$\mathrm{a}_{60}\mathrm{b}_{40}\mathrm{c}_{140}$	$A_{60}B_{40}C_{140}$	$\mathrm{A}_{60}\mathrm{B}_{40}\mathrm{C}_{140}$
14	280	$\mathrm{b}_{60}\mathrm{c}_{80}4_{140}$	$B_{60}C_{80}4_{140}$	$\mathrm{B}_{60}\mathrm{C}_{80}\mathrm{D}_{140}$
16	320	$\mathrm{b}_{20}\mathrm{c}_{60}\mathrm{d}_{140}4_{100}$	$B_{20}C_{60}D_{140}4_{100}$	$\mathrm{C}_{80}\mathrm{D}_{140}\mathrm{E}_{40}4_{60}$
18	360	$\mathrm{c}_{60}\mathrm{d}_{120}4_{120}8_{60}$	$C_{60}D_{120}4_{120}8_{60}$	$\mathrm{D}_{80}\mathrm{E}_{160}4_{120}$
20	400	$\mathrm{d}_{80}\mathrm{e}_{340}8_{200}$	$D_{80}E_{340}8_{200}$	4_{400}
22	440	$\mathrm{e}_{320}8_{120}$	$E_{320}8_{120}$	
24	480	$\mathrm{e}_{200}\mathrm{f}_{160}8_{120}$	$E_{200}F_{160}8_{120}$	
26	520	$\mathrm{e}_{120}\mathrm{f}_{160}8_{240}$	$E_{120}F_{160}8_{240}$	
28	560	$\mathrm{f}_{320}8_{240}$	$F_{320}8_{240}$	
30	600	8_{600}	8_{600}	

Table 4.1: Unequal error protection puncturing architecture for RCPT, RCPC-BICM, and RCPC codes of Tables 3.3–3.5 applied to the subband coder. The notation x_n indicates that n bits are protected using the x curve.

the three schemes). Table 4.1 summarizes the puncturing architecture for the channel encoders and for different source coding bit rates, assuming we use the subband coder of Chapter 2 and the RCPT, RCPC-BICM and RCPC codes given in Tables 3.3–3.5. In addition to the various puncturing patterns, we make use of both uncoded 4-PSK and 8-PSK curves. In Table 4.1, the notation a_m, b_n, c_p, indicates that the first m bits in the prioritized bitstream are protected using the puncturing pattern a, the following n bits with the puncturing pattern b, and the last p bits with the puncturing pattern c. Note also that the number of bits protected with any given level of protection is generally at least twice the traceback depth of the level of protection considered.

Figure 4.1 shows the quality of the different source-coder/RCPT-channel-

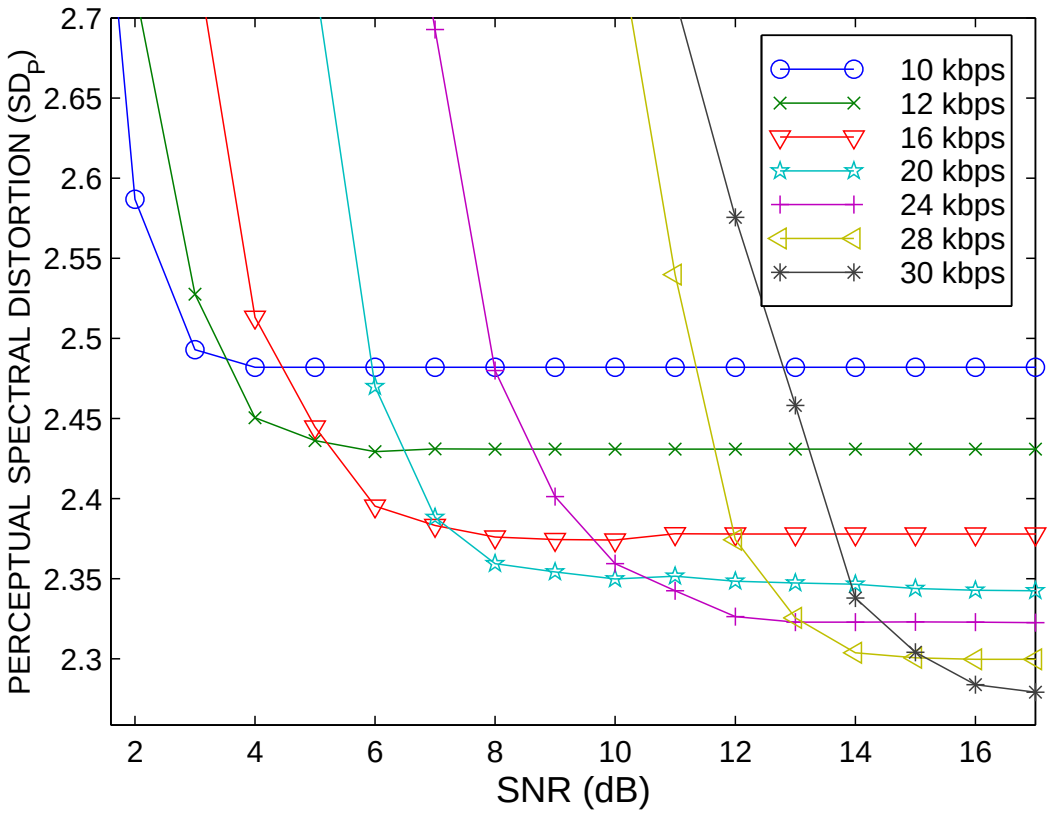


Figure 4.1: Perceptual spectral distortion (SD_P) for the subband coder with RCPT at different bit rates over an AWGN channel. Speech material used is 8 English sentences (4 males and 4 females) from the TIMIT database.

coder pairs simulated with $L_D = 32$ on independent AWGN channels (for clarity, only a few of the source coding bit rates are shown). As expected, no single source coding rate systematically outperforms the others. At low SNR, the 10 kbps source coder with maximum channel protection performs best, while at high SNR, the coders with large source coding bit rates provide the least speech distortion.

For good performance over a wide range of channel conditions, we select for

every SNR the source-channel system that provides the best speech quality. The overall distortion-SNR curve is simply the minimum of all the curves at each SNR. This is an operational rate-distortion [91] curve for this system [92, 93].

Figure 4.2 compares the minimum perceptual distortions obtained for every SNR using the subband coder with RCPC, RCPT and RCPC-BICM codes over an AWGN channel. For each channel SNR and for each channel coding scheme, the smallest spectral distortions obtained by running the system for all the possible source and channel rates are selected. Perceptual speech distortion decreases with increasing SNR and is kept limited even at very low SNR; this would not be true for a scheme with fixed source bit rate and no rate-compatible channel encoder. Furthermore, each rate-distortion curve in Figure 4.2 is lower than if equal error protection were used. Both results illustrate how AMR speech transmission systems are able to provide good speech quality over a wide range of channel conditions.

Note that RCPT and RCPC provide comparable speech quality at intermediate SNRs. However, RCPT produces slightly less distortion at high SNRs, due to its higher per-symbol information rate. Indeed, with a 4-PSK constellation, RCPC only allows up to 20 kbps joint source-channel bit rates, while the 8-PSK constellation of RCPT and RCPC-BICM permits 30 kbps overall bit rates. This effect is noticeable only at high SNRs, where the mutual information of an 8-PSK constellation exceeds the maximum mutual information of a 4 PSK constellation. At intermediate SNRs the per-symbol information rates obtained from both coders are similar. Note also that at low SNRs, RCPT results in less distortion when compared to RCPC-BICM. In summary, RCPT allows for both larger bit rates at high SNRs by using large constellations, and good code performance at low SNRs by combining coding and modulation. It should

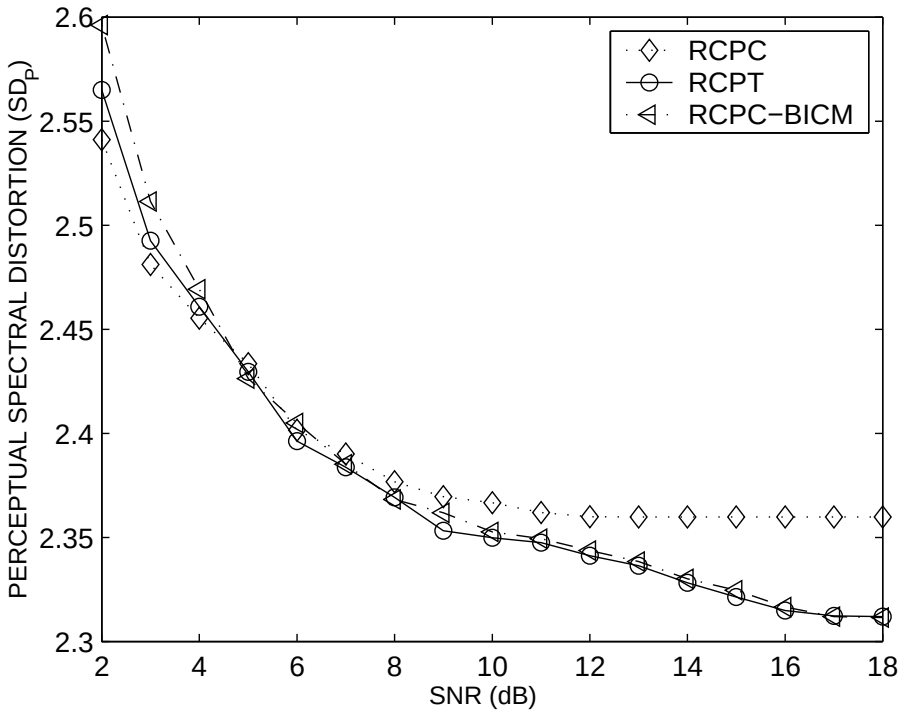


Figure 4.2: Comparison of the operational rate-distortion curves for the complete AMR systems using RCPC, RCPT and RCPC-BICM over an AWGN channel. Speech material used is 8 English sentences (4 males and 4 females) from the TIMIT database.

also be stated that the source-channel coding rate combination that minimizes speech distortion for any channel SNR is exactly the one that was specifically designed for that SNR. This justifies *a posteriori* our AMR system design procedure whose criterion, in the tradeoff between source and channel distortions, was to keep channel distortions just below the audibility threshold.

AMR communication systems work under the assumption that slow channel tracking permits switching to the best AMR operating mode for each channel condition. When channel quality is underestimated, speech quality can be im-

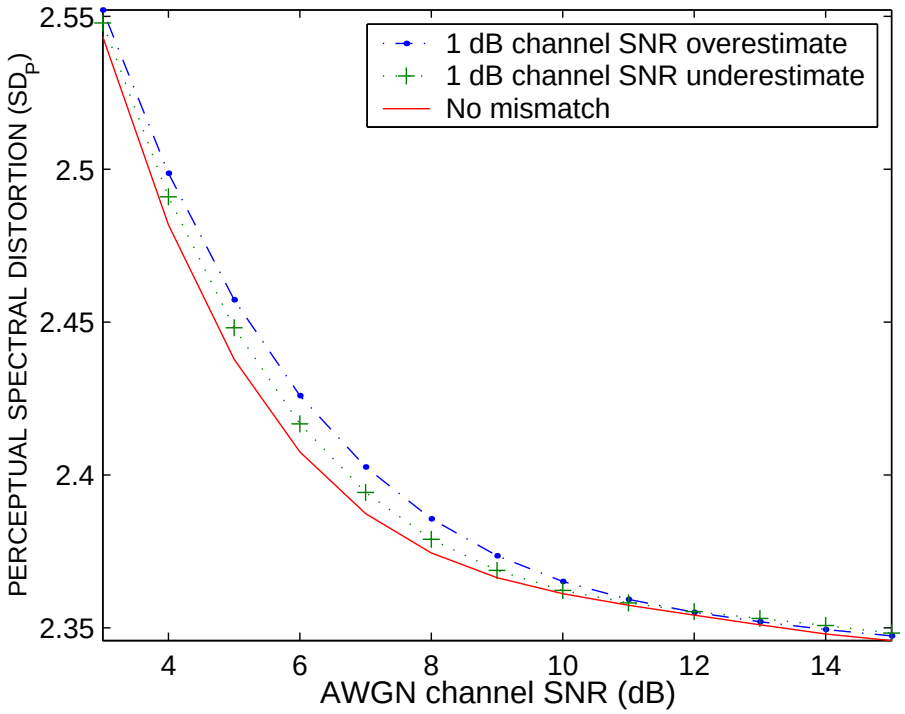


Figure 4.3: Effect of channel mismatch on the subband source coder-RCPT channel coder AMR system performance.

proved by switching to a source coding rate appropriate to the true channel characteristics. When channel quality is overestimated, channel coding protection is not sufficient to protect the bitstream against channel errors, resulting in degraded speech quality. Figure 4.3 shows the effect of channel mismatch on the AMR system performance. It can be seen that overestimating channel quality leads to an erroneous bitstream whose corresponding speech distortion is higher than when using an underperforming source coder as a result of channel quality underestimation.

Note that the means by which the switch between rates is conveyed from the transmitter to the receiver and the related issues such as signaling, channel

estimation error, feedback delay, and channel coherence time are important but are considered beyond the scope of this dissertation.

4.2 AMR system design for the G.727 ADPCM coder

In the previous sections, we showed how an embedded and perceptual coder can be coupled with rate-compatible channel coders in order to build a channel-adaptive speech transmission system with UEP. This section shows that rate-compatible channel coding techniques providing UEP are not specific to perceptual coders and can also be applied to other embedded source coding schemes such as the ITU standard G.727 codec [74].

In order to define the puncturing architecture that provides different levels of protection for bits in the bitstream, we follow the same three steps used for designing an AMR system for the subband coder.

However, a major difference between the two coders is that for the subband coder, bits are grouped in frames, while the G.727 coder operates on a sample by sample basis. As we have seen in Figure 3.3, the traceback depth requirement of the Viterbi decoder requires that we apply the same level of protection to at least twice as many bits as the maximum traceback depth of the code. Therefore, we cannot change the protection requirement on a sample by sample basis. For an (n, m) embedded ADPCM encoder, one needs to frame at least $2 \cdot \max(\mathrm{L_D})$ samples together and group them into maximum n groups requiring different sensitivities. The disadvantage of this procedure is the introduction of a buffering delay in the communication link proportional to the traceback depth of the code. For instance, using the RCPT code presented in Table 3.3 and taking $2 \cdot \max(\mathrm{L_D})$ as the minimum group size, 72 samples must form a frame, which corresponds

ADPCM pair	**Bits/ Sample**	**Rate [kbps]**	**RCPT**	**RCPC-BICM**	**RCPC**
(2,2)	2	16	$a_1\ b_1$	$A_1\ B_1$	$A_1\ B_1$
(3,2)	3	24	$c_1 d_1 e_1$	$C_1 D_1 E_1$	$C_1 E_2$
(4,2)	4	32	$d_1 e_1\ f_1 8_1$	$D_1 E_1\ F_1 8_1$	8_4
(5,2)	5	40	$e_1\ f_1 8_3$	$E_1\ F_1 8_3$	

Table 4.2: Unequal error protection puncturing for RCPT, RCPC-BICM and RCPC codes of Tables 3.3–3.5 applied on the embedded ADPCM G.727 coder. The notation x_n indicates that n bits are protected using the x curve.

to a minimum buffering delay of 9 ms. If the RCPC code of Table 3.5 had been used, at least 184 samples should be grouped, which corresponds to a buffering delay of 23 ms. Note that the buffering delay increases with the mother code complexity and that RCPT codes provide smaller traceback depth requirements than RCPC.

With G.727, the source bit rate varies from 16 kbps to 40 kbps in steps of 8 kbps. For the combined source-channel coding scheme, we limit the source-channel bit rate to 45 kbps, *i.e.* the baudrate is 15 ksymbols/s with an 8-PSK constellation, for RCPT and RCPC-BICM. For RCPC and its 4-PSK constellation, in order to keep approximately the same baudrate, one limits the overall bit rate to 32 kbps.

Table 4.2 illustrates the puncturing architecture for the (n,2) ADPCM encoders with $2 \leq n \leq 5$. The subscripts represent the number of bits per sample protected with the corresponding puncturing level. This number has to be multiplied by the frame size in samples/frame in order to compute the number of successive bits being similarly protected.

Simulations combining RCPT, RCPC-BICM and RCPC codes with embed-

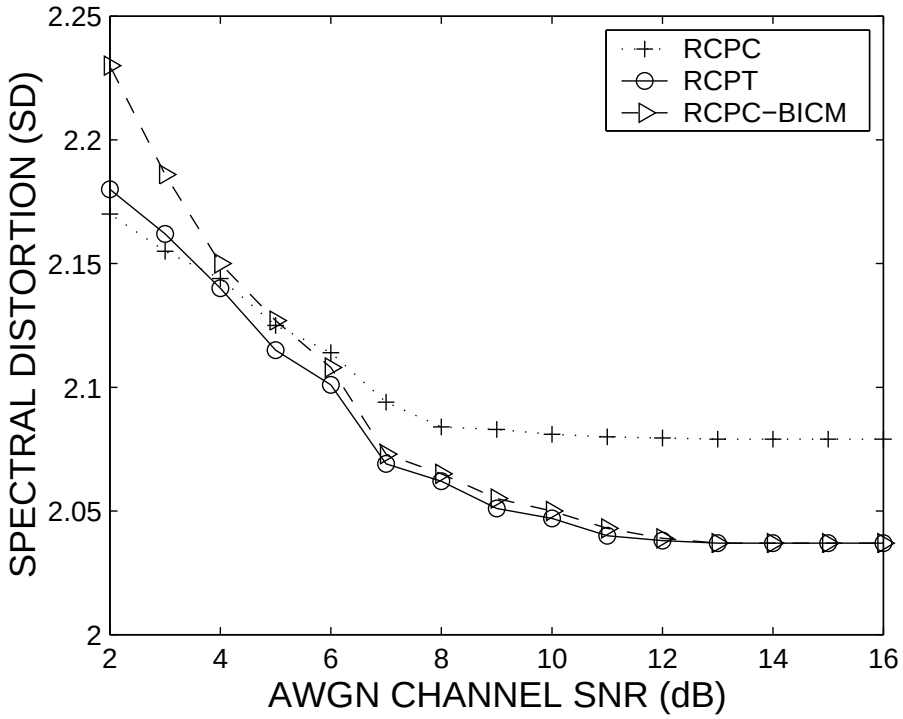


Figure 4.4: Operating distortion curves using RCPC, RCPT and RCPC-BICM with the embedded ADPCM coder.

ded ADPCM at different source coding rates and AWGN channel SNR are performed. Again, according to channel conditions, one can select the mode with the least distortion. The operational distortion-SNR curves of the AMR-UEP systems operating on an AWGN channel are shown in Figure 4.4. The operational distortion-SNR curves are monotically decreasing and operate on a wide range of channel conditions. Higher transmission rates allow RCPT and RCPC-BICM to outperform RCPC at high SNRs. At low SNRs, RCPT outperforms RCPC-BICM due to its superior residual Euclidean distance profile.

4.3 Summary

In this chapter, we presented a solution for designing bandwidth efficient adaptive multi-rate speech transmission systems using embedded speech coders and unequal error protection provided by rate-compatible channel coders.

Two AMR systems were designed: one based on the perceptually-based subband coder and the other based on the embedded ADPCM G.727 speech coding standard. In both cases, the AMR system displayed graceful degradation with decreasing channel SNR over a wide range of channel conditions. The mismatch between channel estimation and the true channel condition has also been studied for the case of the subband coder.

This chapter concludes our analysis on the design of source and channel coding techniques for speech transmission applications. In the next part, we analyze source and channel coding solutions for remote speech recognition applications.

Part II

Source and channel coding for low-bit rate remote speech recognition over error prone channels

This part of the dissertation is organized as follows. Chapter 5 presents efficient quantization techniques for PLP and MFCC features. Chapter 6 analyzes the effect of channel errors and erasures on speech recognition accuracy, and provides a description of channel encoders one can use to efficiently protect speech features against transmission errors. Chapter 6 also presents different channel decoding techniques that maximize recognition accuracy. Chapter 7 presents frame-erasure concealment techniques and frame reliability techniques combined with weighted Viterbi recognition (WVR). Chapter 7 also evaluates the performance of the PLP and MFCC based overall remote speech recognition system, including source coding, channel coding, and frame erasure concealment.

CHAPTER 5

Source coding for remote speech recognition

There are three possible approaches to source coding for remote speech recognition applications, as illustrated in Figure 5.1. The first approach (Figure 5.1(a)) bases recognition on the decoded speech signal. This method suffers from significant recognition degradation at low bit rates [94, 95, 96]. A second approach (Figure 5.1(b)) builds a recognition engine based on coding parameters of commonly used speech codecs, without re-synthesizing the speech signal [97, 98, 99, 100, 101, 102]. The third approach (Figure 5.1(c)) performs recognition on quantized ASR features [103, 104, 105, 106, 107].

Depending on the approach taken, the information transmitted to the server is either the speech coding bitstream or a sequence of spectral observations. In the former case, the client compresses speech, the server decodes it and performs feature extraction before running the recognition engine (Figure 5.1(a)). Alternatively, the recognition engine may utilize directly as input the compressed speech coding features (or a transformation thereof), bypassing the speech synthesis stage (Figure 5.1(b)). In the latter case, the front-end processing for feature extraction, which is not expensive computationally, is done at the client and the server decompresses the features for recognition (Figure 5.1(c)).

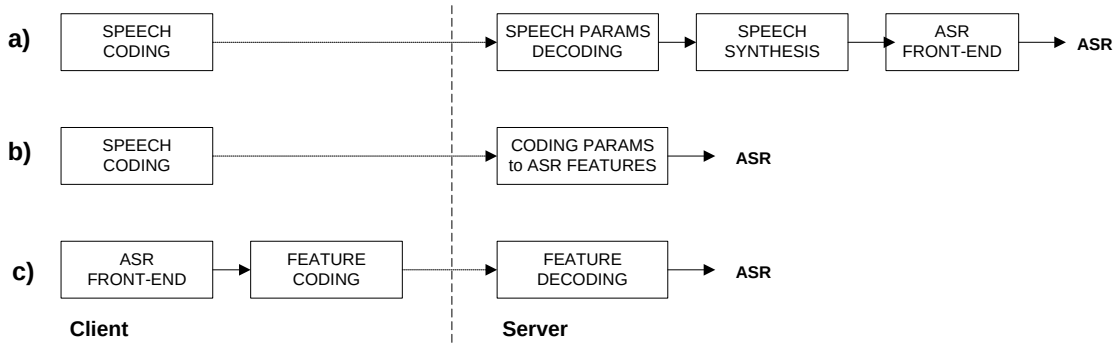


Figure 5.1: Block diagram of the different approaches for remote speech recognition: a) ASR features extracted from decoded speech, b) transformation of speech coding parameters to ASR features, c) ASR feature quantization.

5.1 Recognition based on decoded speech signals

In this section, we report on recognition results based on two low bit rate speech coding standards, the Federal Standard FS-1016 Code Excited Linear Prediction (CELP) coder [108] operating at 4.8 kbps and the Department of Defense Mixed Excitation Linear Prediction (MELP) standard [109] operating at 2.4 kbps. Recognition accuracies obtained with these two low bit rate speech coders are compared with those of two GSM speech coding standards without channel errors.

Hidden Markov Model (HMM) based isolated digit recognition experiments (with HTK 2.1 [3]) using the TI-46 speaker independent database are performed. Since the speech database contains signals sampled at 16 kHz and the speech coders studied are narrowband, speech files are first downsampled by a factor of two. Linear prediction cepstral coefficients (LPCCs), with their first and second order time-derivatives, are extracted from the decoded speech signal. Training is always performed on the original speech signal.

RATE [kbps]	**CODER**	**BER**			
		0%	2%	5%	10%
4.8	CELP	93.88	89.38	79.06	64.19
2.4	MELP	98.15	96.13	89.14	70.82

Table 5.1: Isolated digit recognition accuracy based on LPCCs extracted from MELP and CELP decoded speech signals at different BERs using the TI-46 database.

Table 5.1 summarizes speech recognition results for the MELP and CELP coders operating over binary symmetric channels (BSC) characterized by different bit error rates (BER).

Note that the MELP coder, while operating at a lower bit rate than the CELP coder, performs significantly better. We have analyzed this question further and concluded that the main problem of CELP coding for speech recognition is the mismatch between the CELP coded test set and the uncoded training set. CELP coding introduces distortions in the speech signal whose effect is that ASR features extracted from CELP coded speech signals are no longer well represented by HMM models trained on uncoded speech. This was verified by performing both training and testing on CELP coded speech, and observing that recognition improved from 93.88% to 97.45%. For the MELP and GSM coders, on the other hand, little was gained by performing both training and testing on coded speech, avoiding the mismatch problem.

Several features of the MELP coder may also explain the performance differences. First, while the MELP coder is also based on the traditional LPC parametric model, it includes five additional features [109, 75]: mixed excitation, aperiodic pulses, adaptive spectral enhancement, pulse dispersion and Fourier magnitude modeling. For the purpose of speech recognition, we believe that

CODER	MELP	CELP	GSM-HR	GSM-EFR
RATE (kbps)	2.4	4.8	5.6	12.2
Recognition	98.15	93.88	98.51	98.78

Table 5.2: Illustration of speech recognition accuracy using different speech coding standards, using the TI-46 database.

three of these features help improve recognition accuracy: 1) the mixed excitation, implemented using a multi-band mixing model, which simulates frequency dependent voicing strength; 2) adaptive spectral enhancement filter, based on the poles of the LPC filter, which enhances the formant structure of the synthetic speech; and 3) Fourier magnitude modeling, obtained by picking peaks in the Fourier transform of the residual signal, which improves accuracy of the speech production model at lower frequencies. Second, MELP coders update the synthesis parameters every pitch period using parameter interpolation. Third, for unvoiced speech frames, MELP includes a frame erasure detection mechanism and 13 bits of additional channel protection, which may explain its relative robustness to channel errors.

For a better comprehension of the merits of CELP and MELP coding for remote recognition, their recognition performance is compared to that of widely used speech coding standards in wireless applications. In particular, the same recognition experiments are carried out using the Enhanced Full-Rate (EFR) and Half-Rate (HR) GSM codecs operating at 12.2 kbps and 5.6 kbps, respectively.

Table 5.2 illustrates the merits of the different coders with respect to their bit rates in error-free transmission. Note the particularly attractive behavior of the MELP coder, whose performance approaches that of GSM at a significantly lower bit rate.

These levels of recognition accuracy for the specific task and database confirm those presented in [94, 95, 96] on the effect of speech coding and reconstruction on speech recognition accuracy. In particular, [94] confirms the poor performance of LPCCs and MFCCs extracted from CELP decoded speech signals.

5.2 Recognition based on speech coding parameters

A main advantage of performing speech coding bitstream-based feature extraction is the ability to use a speech coding standard at the client, which eliminates the need for defining an ASR quantization standard solely for remote speech recognition applications. For instance, [100] uses the IS-641 coding standard as a basis for performing speech coding bitstream-based ASR feature extraction.

We have seen that MELP coding results in good recognition accuracy with bit rates that are significantly less than the GSM speech coding standards. A remaining question is to see whether 1) better results could be obtained without re-synthesizing speech by performing ASR based on speech coding parameters (or a transformation thereof), and hence whether 2) one could further reduce the bit rates for remote recognition without significant loss in accuracy by transmitting only a portion of the MELP bitstream.

MELP, like many of the standard low bit rate speech coders, uses a 10^{th} order auto-correlation based linear predictive coding (LPC). LPCCs can be extracted from the transmitted LPC coefficients, allowing the recognition engine to utilize speech coding parameters directly.

Typically, for low bit rate LP coders, about half the bitstream is dedicated to encoding parameters representing the spectrum-shaping vocal tract impulse response, while the rest is used to characterize the excitation signal. Since removal

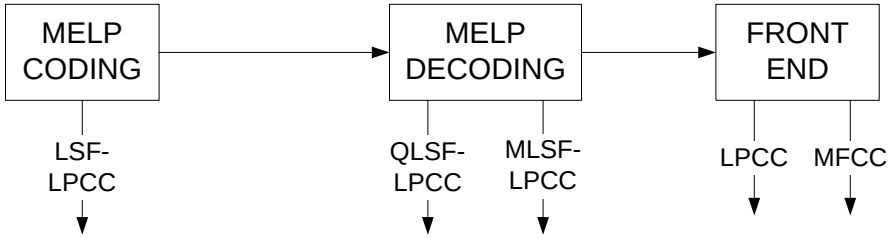


Figure 5.2: Block diagram of possible MELP-based recognition experiments.

of excitation information is usually desirable in speech recognition, only about half the bitstream is useful for recognition. Also, in speech coding, naturalness and intelligibility require precision in the rendering of the speech spectrum after quantization. On the other hand, speech recognition characteristics suggest that coarser spectrum representations should still provide good recognition results. This implies that lower bit rates may be sufficient to provide reliable recognition.

As explained in Section 1.3, a resourceful method for quantizing LPC parameters involves the use of line spectrum frequencies (LSFs) to represent the LPC spectrum. LSFs improve coding efficiency, are real, vary smoothly and guarantee stable minimum-phase filters, even after quantization. LSFs can also be linearly interpolated, allowing them to be updated faster than they are quantized.

The following experiment, illustrated in Figure 5.2, evaluates the merits of remote recognition based on speech coding by comparing recognition performance obtained using five different scenarios: 1) LPCCs iteratively computed from unquantized LSFs (LSF-LPCC), 2) LPCCs computed from MELP quantized LSFs (QLSF-LPCC), 3) LPCCs iteratively computed from MELP quantized and MELP processed LSFs using pitch and gain information (MLSF-LPCC), 4) LPCCs extracted from reconstructed speech (LPCC), and 5) MFCCs extracted from reconstructed speech (MFCC).

Implementation details of the experiments are as follows. In the first two scenarios (LSF-LPCC and QLSF-LPCC), LSFs are linearly interpolated by a fixed factor of two, from every 22.5 ms to every 11.25 ms. In the third scenario (MLSF-LPCC), LSFs are linearly interpolated every pitch period (6-15 ms) and the interpolation factor is a function of the neighboring frame gains. In the last two scenarios (LPCC and MFCC), LSFs are extracted every 10 ms, as in most speech recognition systems. Training is always performed on the original speech signal.

While the main purpose of this experiment is to analyze recognition based on the speech coding bitstream, the last two scenarios are included in order to analyze the potential degradation in recognition accuracy implied by the absence of speech synthesis at the decoder. The first scenario is introduced to analyze the effect of LSF quantization on recognition.

In MELP coding, LSFs are quantized using a multi-stage vector quantizer (MSVQ) [110]. The MSVQ codebook contains four stages of 7, 6, 6 and 6 bits, respectively. The search procedure is an M-best approximation to a full search, in which the M=8 best codevectors from each stage search are used by the next stage search. The two experiments based on quantized LSFs (QLSF-LPCC and MLSF-LPCC) are carried out with different precision in LSF quantization (from one to four MSVQ stages) and hence in bit rates.

Table 5.3 shows recognition results for each of the five possible scenarios over BSC channels with different cross-over probabilities. For the ASR features based on quantized LSFs (QLSF and MLSF), performances are subdivided depending on how many stages (MSVQ_i, $1 \leq i \leq 4$) of the MSVQ search are included in the quantization.

The results indicate that 1) MLSF does not perform as well as QLSF, mainly

Feature	Bitrate	Quant.	BER=0%	BER=2%	BER=5%	BER=10%
LPCC	2.40 kb/s	MELP	98.15	96.13	89.14	70.82
MFCC	2.40 kb/s	MELP	98.63	97.71	90.18	71.11
MLSF	0.31 kb/s	$MSVQ_1$	43.67	44.94	43.67	39.87
	0.58 kb/s	$MSVQ_2$	88.61	82.28	67.72	50.00
	0.84 kb/s	$MSVQ_3$	94.30	89.24	79.11	61.39
	1.11 kb/s	$MSVQ_4$	94.94	91.14	81.65	66.46
QLSF	0.31 kb/s	$MSVQ_1$	72.78	70.25	65.19	44.30
	0.58 kb/s	$MSVQ_2$	94.30	93.31	85.44	72.15
	0.84 kb/s	$MSVQ_3$	97.47	93.67	91.77	72.78
	1.11 kb/s	$MSVQ_4$	97.91	94.94	91.14	75.95
LSF	—	—	98.10	—	—	—

Table 5.3: Recognition performance using different MELP coding based ASR features.

because the timing of the observation sequence has been modified by the MELP decoder and no longer matches the training set which uses uniform 10 ms frame shifts; 2) LSFs quantized with 4 stages perform almost as well as unquantized LSFs (at least with no channel errors); 3) QLSF performance is similar to that of LPCC or MFCC, with a great reduction in bit rates and complexity for the server (no reconstruction of speech); and 4) unquantized LSF performance approaches but does not match that of MFCCs extracted from the decoded speech signal, which suggests that there is room for accuracy improvement, and perhaps also bit rate reduction, by quantizing ASR features directly.

Note that our analysis only used the short-term prediction approximation of the vocal tract spectrum for the purpose of recognition. However, some research results have shown improved performance by using voiced/unvoiced information [111]. This feature is used in [100], in which the adaptive and fixed

codebook gains are used as a representation of the voiced/unvoiced information for improved recognition performance.

5.3 Quantization of ASR features: PLP

Linear prediction based cepstral coefficients can be extracted from a standard linear prediction model or from a perceptual linear prediction model [30] (PLP) which models human auditory perception. One advantage of PLP is that its spectra can be represented using a low order all-pole model. This yields a low dimensional representation [112], which is advantageous for quantization. Typically, p=5–6 is sufficient to represent two formants and a spectral tilt.

In this section, we study the quantization of one particular type of ASR feature, the PLP cepstral coefficients (P-LPCCs). Quantization of LPCCs is not studied here since it can be based on the transmission of the LP coefficients α_i (or a similar representation of it such as LAR, RC or LSF), which has already been extensively studied in the literature [7, 113, 114, 115, 116, 117, 118, 119]. Quantization of MFCC features is covered in Section 5.4.

5.3.1 Quantizing P-LSFs or P-LPCCs

The transfer function of the PLP filter can be written as $A(z) = 1 - \sum_{k=1}^{p} \alpha_k z^{-k}$. From the α_i prediction coefficients, one can compute the PLP cepstral coefficients (P-LPCC) (c_n) using the following recursive formula,

$$c_n = \alpha_n + \sum_{k=1}^{n-1} \frac{k}{n} \; c_k \; \alpha_{n-k}, \tag{5.1}$$

or alternatively the perceptual line spectral frequencies P-LSFs (ω_n),

$$\omega_n = \text{roots } [A(z) \; \pm \; z^{-(p+1)} \; A(z^{-1})]. \tag{5.2}$$

Coefficient	1	2	3	4	5	6
P-LSF	0.87	0.88	0.88	0.91	0.92	0.90
P-LPCC	0.88	0.93	0.91	0.91	0.86	0.86

Table 5.4: Average (across all digits) inter-frame correlations between the six P-LSFs and P-LPCCs extracted from PLP_6 of adjacent frames, using 25 ms Hamming windows shifted every 20 ms for 5 minutes of speech.

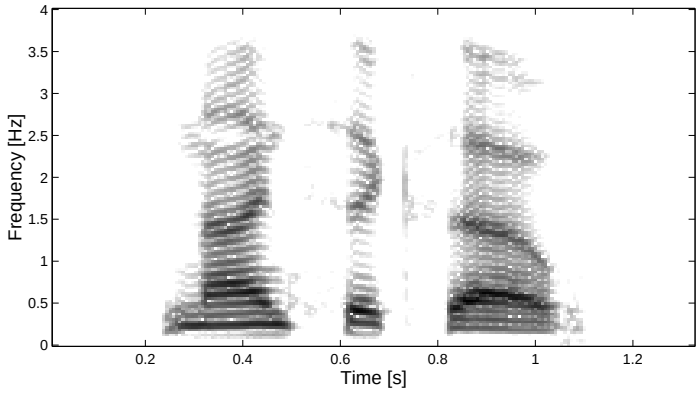
For visualization of both sets of data, Figure 5.3 illustrates the spectrogram, the PLP cepstral coefficients and the PLP line spectral frequencies for the digit string "9 6 0" pronounced by a male speaker.

While speech recognition is performed using the P-LPCCs, we analyze next whether quantizing the P-LSFs might lead to better compression ratios, knowing that P-LPCCs can also be re-computed from the P-LSFs. On one hand, quantizing P-LPCCs guarantees minimizing the Euclidean distance between quantized and unquantized P-LPCCs, assuring a close match between coded and uncoded feature vectors. On the other hand, P-LSFs typically improve coding efficiency, and can be linearly interpolated between transmitted P-LSF values, allowing the P-LSFs to be updated more often than they are quantized.

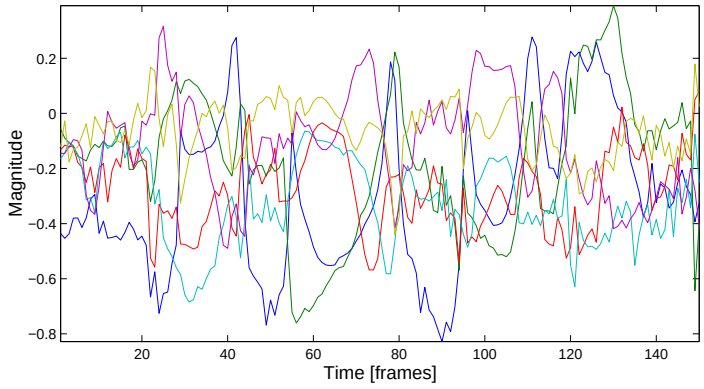
In order to determine which information should be transmitted, we analyze three properties for both P-LPCCs and P-LSFs.

Inter-frame correlation

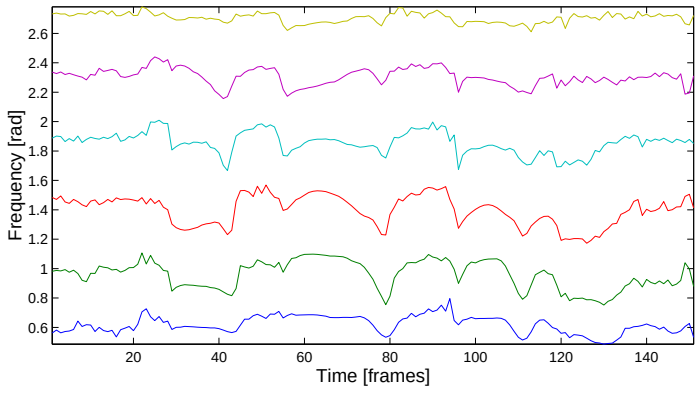
Time correlation can be exploited with predictive coding to reduce the dynamic range of the information to quantize. Five minutes of the speaker dependent TI-46 digit database are used to compute time-correlations between P-LSFs and P-LPCCs of neighboring frames (20 ms apart). Results are shown in Table 5.4



(a) Spectrogram



(b) P-LPCCs



(c) P-LSFs

Figure 5.3: Illustration of the (a) spectrogram, (b) P-LPCCs and (c) P-LSFs of the digit string "9 6 0" pronounced by a male speaker.

for a 6^{th} order PLP spectrum. We present here analysis for $p = 6$. Note that both P-LPCCs and P-LSFs display high inter-frame correlation. An encoder can exploit this time redundancy by transmitting only the residual error after prediction.

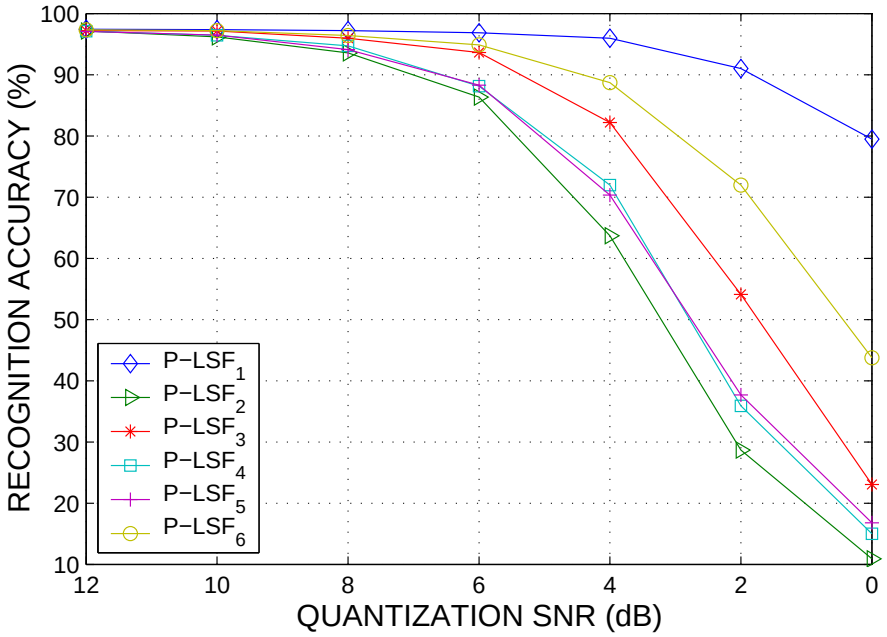
Intra-frame correlation

Statistical correlation between elements of a feature vector can be exploited by vector quantization. Tables 5.5(a) and 5.5(b) indicate the intra-frame correlations of the residual P-LPCCs and P-LSFs after first-order prediction. Large intra-frame correlations are observed for the P-LSFs, which can be efficiently exploited using vector quantization (VQ). P-LPCCs, however, are almost uncorrelated. This is a property often sought for ASR features, since it allows for the features to be treated independently without significant loss in performance or generality.

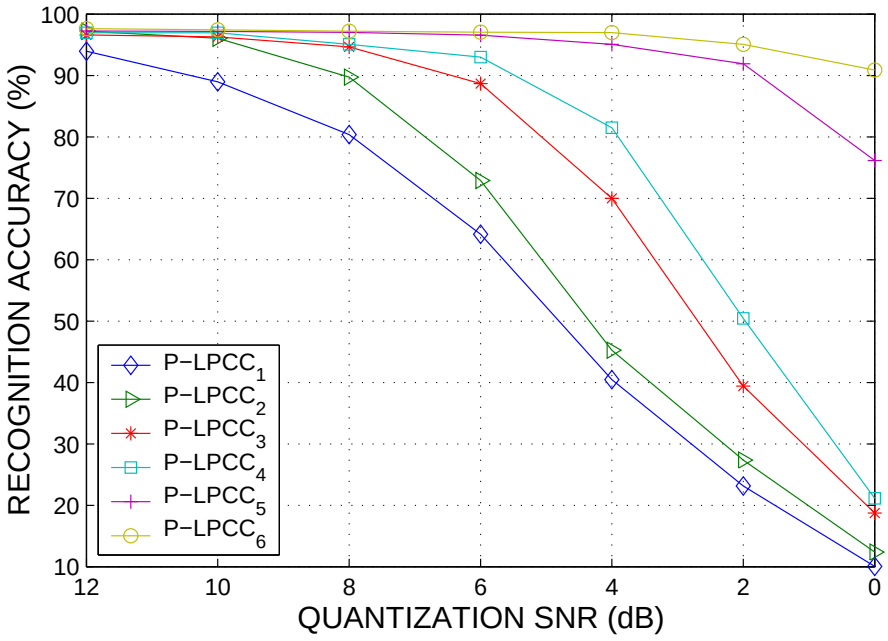
Sensitivity to quantization noise

The third factor to consider when developing quantization schemes for speech recognition applications is the sensitivity of recognition accuracy with respect to quantization error. Figure 5.4 illustrates the sensitivity of continuous digit recognition results with respect to quantization errors when separately scalar quantizing the P-LSF and P-LPCC residuals after first-order prediction. The recognition experiment consisted of performing continuous digit recognition using HTK 3.0 and the Aurora-2 database [104] with 16 states and 6 mixtures per word model. More than 3300 tokens were tested.

The results of Figure 5.4 may vary slightly from those presented in [105], where the recognition task was independent digit recognition using the TI-46 digit database (1180 male and female tokens for training, 480 for testing) and HTK 2.1 with 5 states and 3 mixtures per word model, indicating dependency



(a) Sensitivities of line spectral frequencies (P-LSFs).



(b) Sensitivities of cepstral coefficients (P-LPCCs).

Figure 5.4: Quantization error sensitivity analysis of the (a) P-LSFs and (b) P-LPCCs extracted from PLP_6.

$\rho(i,j)$	1	2	3	4	5	6
1	1.00	0.79	0.60	0.27	0.05	0.17
2	0.79	1.00	0.84	0.35	0.11	0.20
3	0.60	0.84	1.00	0.58	0.16	0.29
4	0.27	0.35	0.58	1.00	0.63	0.38
5	0.05	0.11	0.16	0.63	1.00	0.69
6	0.17	0.20	0.29	0.38	0.69	1.00

(a) Intra-frame correlation for P-LSFs

$\rho(i,j)$	1	2	3	4	5	6
1	1.00	-0.70	-0.11	-0.15	-0.16	-0.44
2	-0.70	1.00	-0.19	-0.48	-0.03	-0.29
3	-0.11	-0.19	1.00	-0.15	-0.37	0.26
4	-0.15	-0.48	-0.15	1.00	-0.10	-0.01
5	-0.16	-0.03	-0.37	-0.10	1.00	-0.04
6	-0.44	-0.29	0.26	-0.01	-0.04	1.00

(b) Intra-frame correlation for P-LPCCs

Table 5.5: Intra-frame correlation of the residual (a) P-LSFs and (b) P-LPCCs after first-order prediction.

on the database and task.

Note that the P-LPCCs are significantly more sensitive to quantization errors than P-LSFs. Sensitivities also vary with the order of the P-LSF or P-LPCC feature. The individual sensitivities will be taken into account in the training and search of the vector quantizers. For instance, one can see in Figure 5.4 that error in the quantization of the first P-LSF only marginally reduces recognition accuracy. This means that the cepstral coefficients, based on which speech recognition is performed, did not change much with a somewhat significant modification of ω_1, or that the change was not important.

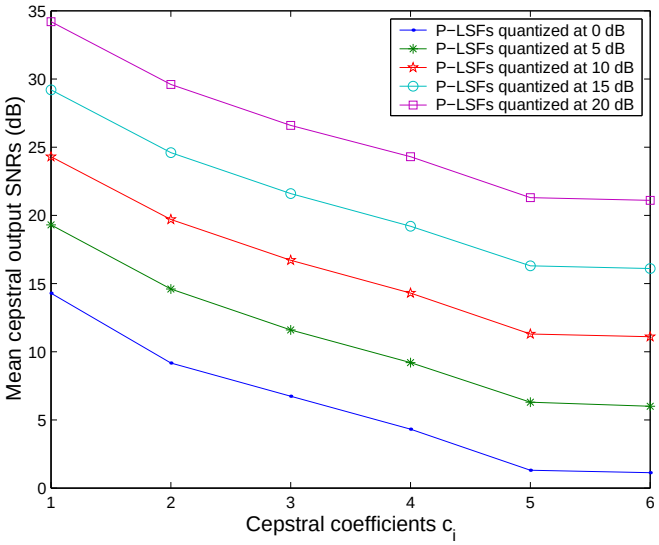
SNR (dB)	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	Mean
c_1	18.98	13.55	10.02	9.80	12.71	20.81	14.29
c_2	10.08	11.40	11.73	9.91	7.01	11.12	9.71
c_3	5.22	10.84	4.56	3.13	7.53	5.43	6.73
c_4	5.71	2.86	4.19	2.03	6.74	3.67	4.32
c_5	7.81	-3.30	0.29	1.81	-1.17	2.12	1.31
c_6	6.32	-0.32	-2.72	-3.83	-2.19	5.17	1.13
Mean	9.19	5.91	4.80	3.75	5.04	8.01	

Table 5.6: Analysis of the partial sensitivities $\frac{\partial c_i}{\partial \omega_j}$ by studying the effect on the i^{th} cepstral coefficient of quantizing the j^{th} perceptual line spectral frequency. Each P-LSF is individually quantized at 0 dB.

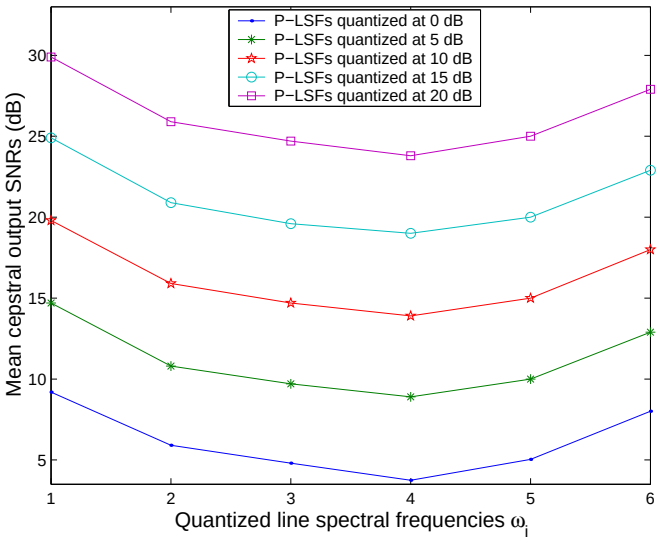
To confirm these results, the following experiment evaluating $\frac{\partial c_i}{\partial \omega_j}$ is carried out. Each P-LSF is individually disturbed (quantized) with a given SNR, and the resulting SNRs of the cepstral coefficients computed from the disturbed P-LSFs are evaluated. Table 5.6 illustrates the SNRs of the P-LPCCs obtained after individual and successive scalar quantization of the P-LSFs at 0 dB SNR.

Two major conclusions are drawn from the results shown in Table 5.6. First, as already suggested in Figure 5.4, P-LSFs with low and high order (ω_1 and ω_6) have less impact on the SNRs. Second, note that low-order cepstral coefficients are better represented than the higher-order coefficients.

Results of Table 5.6, when successively quantizing each of the P-LSFs with 0 dB, are confirmed when quantizing the P-LSFs with different resolution levels, as shown in Figure 5.5. Only the horizontal means (for the effect of P-LSF quantization on cepstral precision) and vertical means (for the effect of quantizing each individual P-LSF) are reported.



(a) Mean SNRs of each cepstral coefficient after individual P-LSF quantization at different SNRs.



(b) Mean SNRs of all cepstral coefficients after individual P-LSF quantization at different SNRs.

Figure 5.5: Sensitivity analysis of the cepstral coefficients after P-LSF quantization at different SNRs: (a) mean output SNR for each cepstral coefficient; (b) mean SNR of all cepstra depending on the quantized P-LSF.

5.3.2 Mathematical sensitivity analysis to P-LSF quantization

The goal of this section is to mathematically analyze the sensitivity of the cepstral coefficients to explain the two trends observed in Figure 5.5, *i.e.* smaller sensitivity of the low and high order P-LSFs and better representation of the low order cepstral coefficients.

The Jacobian matrix $J = \frac{\partial c_i}{\partial \omega_j}$ provides partial answers for both questions since it analyzes the effect of a small disturbance of a given line spectral frequency ω_j on the cepstral coefficients c_i. Computation of the cepstral coefficients from the P-LSFs includes 1) the transformation from the P-LSFs into LP coefficients, and 2) the computation of the cepstral coefficients from the LP coefficients. The Jacobian matrix $\frac{\partial c_i}{\partial \omega_j}$ can therefore be computed as follows:

$$\frac{\partial c_i}{\partial \omega_j} = \sum_{k=1}^{p} \frac{\partial c_i}{\partial \alpha_k} \cdot \frac{\partial \alpha_k}{\partial \omega_j}. \tag{5.3}$$

In matrix notation, this corresponds to a matrix multiplication

$$J = J_C \cdot J_A, \tag{5.4}$$

where the matrix J is defined as

$$J(i,j) = \frac{\partial c_i}{\partial \omega_j} \tag{5.5}$$

$$= \begin{bmatrix} \frac{\partial c_1}{\partial \omega_1} & \cdots & \frac{\partial c_1}{\partial \omega_p} \\ \vdots & \ddots & \vdots \\ \frac{\partial c_p}{\partial \omega_1} & \cdots & \frac{\partial c_p}{\partial \omega_p} \end{bmatrix}, \tag{5.6}$$

J_C as

$$J_C(i,k) = \frac{\partial c_i}{\partial \alpha_k} \tag{5.7}$$

$$= \begin{bmatrix} \frac{\partial c_1}{\partial \alpha_1} & \cdots & \frac{\partial c_1}{\partial \alpha_p} \\ \vdots & \ddots & \vdots \\ \frac{\partial c_p}{\partial \alpha_1} & \cdots & \frac{\partial c_p}{\partial \alpha_p} \end{bmatrix}, \tag{5.8}$$

and J_A as

$$J_A(k,j) = \frac{\partial \alpha_k}{\partial \omega_j} \tag{5.9}$$

$$= \begin{bmatrix} \frac{\partial \alpha_1}{\partial \omega_1} & \cdots & \frac{\partial \alpha_1}{\partial \omega_p} \\ \vdots & \ddots & \vdots \\ \frac{\partial \alpha_p}{\partial \omega_1} & \cdots & \frac{\partial \alpha_p}{\partial \omega_p} \end{bmatrix}. \tag{5.10}$$

We first evaluate the matrix $\boldsymbol{J_C} = \frac{\boldsymbol{\partial c}}{\boldsymbol{\partial \alpha}}$.

From Eq. 5.1, it can be seen that the vector representing the cepstral coefficients c_n can be written as a function of the linear prediction coefficients α_i as follows (for a 6th order predictive system),

$$C = \begin{bmatrix} \alpha_1 \\ \alpha_2 + \frac{1}{2}\alpha_1^2 \\ \alpha_3 + \alpha_1\alpha_2 + \frac{1}{3}\alpha_1^3 \\ \alpha_4 + \alpha_1\alpha_3 + \frac{1}{2}\alpha_2^2 + \alpha_2\alpha_1^2 + \frac{1}{3}\alpha_1^3 + \frac{1}{4}\alpha_1^4 \\ \alpha_5 + \alpha_1\alpha_4 + \alpha_3\alpha_2 + \alpha_3\alpha_1^2 + \alpha_1\alpha_2^2 + \alpha_2\alpha_1^3 + \frac{1}{5}\alpha_1^5 \\ \alpha_6 + \alpha_1\alpha_5 + \alpha_4\alpha_2 + \alpha_4\alpha_1^2 + \frac{1}{2}\alpha_3^2 + 2\alpha_3\alpha_1 + \alpha_2\alpha_3\alpha_1^3 + \\ \qquad\qquad \frac{1}{3}\alpha_2^3 + \frac{3}{2}\alpha_1^2\alpha_2^2 + \alpha_2\alpha_1^4 + \frac{1}{6}\alpha_1^6 \end{bmatrix}. \tag{5.11}$$

From Eq. 5.11, one can easily compute the Jacobian matrix J_C by taking the derivative of C with respect to the prediction coefficients α_k:

$$J_C(i,k) = \begin{bmatrix} J_{C_0} & 0 & 0 & 0 & 0 & 0 \\ J_{C_1} & J_{C_0} & 0 & 0 & 0 & 0 \\ J_{C_2} & J_{C_1} & J_{C_0} & 0 & 0 & 0 \\ J_{C_3} & J_{C_2} & J_{C_1} & J_{C_0} & 0 & 0 \\ J_{C_4} & J_{C_3} & J_{C_2} & J_{C_1} & J_{C_0} & 0 \\ J_{C_5} & J_{C_4} & J_{C_3} & J_{C_2} & J_{C_1} & J_{C_0} \end{bmatrix}, \tag{5.12}$$

where $J_C(i,k) = J_{C_{i-k}}$ $(i \geq k)$ and

$$J_{C_0} = 1 \tag{5.13}$$

$$J_{C_1} = \alpha_1 \tag{5.14}$$

$$J_{C_2} = \alpha_2 + \alpha_1^2 \tag{5.15}$$

$$J_{C_3} = \alpha_3 + \alpha_2\alpha_1 + \alpha_1^3 \tag{5.16}$$

$$J_{C_4} = \alpha_4 + 2\alpha_1\alpha_3 + \alpha_2^2 + 3\alpha_1^2\alpha_2 + \alpha_1^4 \tag{5.17}$$

$$J_{C_5} = \alpha_5 + 2\alpha_1\alpha_4 + 2\alpha_3\alpha_2 + 3\alpha_3\alpha_1^2 + 3\alpha_1\alpha_2^2 + 4\alpha_2\alpha_1^3 + \alpha_1^5. \tag{5.18}$$

The fact that the Jacobian matrix $J_C(i,k)$ is lower-triangular and Toeplitz, with increased complexity in the elements further away from the diagonal, explains the better representation of the lower-order cepstral coefficients. The increased sensitivity with the cepstral coefficient order is a direct consequence of the recursive formula in Eq. 5.1. Indeed, since the value of c_{n-1} is used to compute c_n, errors tend to propagate into the computation of the high order cepstra; hence, the quantization quality decreases as the cepstral order increases.

For the second part of the sensitivity computation, we need to evaluate the Jacobian matrix $\boldsymbol{J_A} = \frac{\partial \boldsymbol{\alpha}}{\partial \boldsymbol{\omega}}$.

The P-LSF set is determined from the LP coefficients by first forming the polynomials

$$P(z) = A(z) \; + \; z^{-(p+1)} \; A(z^{-1}) \tag{5.19}$$

and

$$Q(z) = A(z) \; - \; z^{-(p+1)} \; A(z^{-1}). \tag{5.20}$$

The line spectral frequencies are defined as the angular frequencies ω_i of the roots of $P(z)$ and $Q(z)$, which can be shown to be interlaced. Let the P-LSFs be denoted by $\omega_1, \omega_2, \cdots, \omega_p$, so the roots of $P(z)$ correspond to the odd indices, and the roots of $Q(z)$ correspond to the even indices [120].

Since we try to find the sensitivity of time domain parameters (the coefficients of the linear prediction filter α_n) with respect to frequency domain parameters (the angular frequencies of the roots of $P(z)$ and $Q(z)$), we first need to express the impulse response of the linear prediction filter $A(z) = 1 - \sum_{k=1}^{p} \alpha_k z^{-k}$ in terms of the P-LSF frequencies. This is done by analyzing the transfer function $A(z)$ along the unit-circle to obtain the Fourier transform $A(\omega)$ of the time-domain impulse response $a(n)$ of the linear prediction filter,

$$A(\omega) = \frac{1}{2}(P(\omega) + Q(\omega)). \tag{5.21}$$

Given the symmetry of the polynomials, the roots of $P(z)$ and $Q(z)$ are complex conjugate and can be combined to yield

$$P(w) = (1 + e^{-jw}) \prod_{i \text{ odd}} (1 - 2\cos\omega_i e^{-jw} + e^{-2jw}) \tag{5.22}$$

and

$$Q(w) = (1 - e^{-jw}) \prod_{i \text{ even}} (1 - 2\cos\omega_i e^{-jw} + e^{-2jw}) \ . \tag{5.23}$$

To simplify the notation and to keep the time-frequency domain interpretation, the following Fourier transforms (shown with the symbol $\Leftrightarrow$) are defined

$$\tilde{p}_0(n) = \delta(n) + \delta(n-1) \qquad \Leftrightarrow \qquad \tilde{P}_0(\omega) = 1 + e^{-jw}, \tag{5.24}$$

$$\tilde{q}_0(n) = \delta(n) - \delta(n-1) \qquad \Leftrightarrow \qquad \tilde{Q}_0(\omega) = 1 - e^{-jw}, \tag{5.25}$$

for the terms independent of the P-LSFs, and

$$\begin{aligned} \tilde{p}_i(n) &= \delta(n) - 2\cos\omega_{2i-1}\delta(n-1) + \delta(n-2) \ \Leftrightarrow \\ \tilde{P}_i(\omega) &= 1 - 2\cos\omega_{2i-1}e^{-jw} + e^{-2jw}, \end{aligned} \tag{5.26}$$

together with

$$\begin{aligned} \tilde{q}_i(n) &= \delta(n) - 2\cos\omega_{2i}\delta(n-1) + \delta(n-2) \ \Leftrightarrow \\ \tilde{Q}_i(\omega) &= 1 - 2\cos\omega_{2i}e^{-jw} + e^{-2jw} \ , \end{aligned} \tag{5.27}$$

for the terms depending on ω_i.

One can then re-write $P(z)$ and $Q(z)$ as

$$p(n) = \bigwedge_{k=0}^{p/2} \tilde{p}_k(n) \qquad \Leftrightarrow \qquad P(\omega) = \prod_{k=0}^{p/2} \tilde{P}_k(\omega), \tag{5.28}$$

and

$$q(n) = \bigwedge_{k=0}^{p/2} \tilde{p}_k(n) \qquad \Leftrightarrow \qquad Q(\omega) = \prod_{k=0}^{p/2} \tilde{Q}_k(\omega), \tag{5.29}$$

where the symbol $\bigwedge$ stands for a series of convolutions

$$\bigwedge_{k=0}^{L} v_k(n) = v_0(n) * v_1(n) * \cdots * v_L(n), \tag{5.30}$$

which is the time-domain correspondence of a product in the frequency domain.

Together, Eqs. 5.24 through 5.30 create the link between the P-LSFs and the prediction filter, such that the partial derivatives can be obtained using

$$\frac{\partial \alpha_n}{\partial \omega_i} = \begin{cases} \frac{1}{2}\frac{\partial p(n)}{\partial \omega_i}; & i \text{ odd} \\ \frac{1}{2}\frac{\partial q(n)}{\partial \omega_i}; & i \text{ even} \end{cases} \qquad \Leftrightarrow \quad \frac{\partial A(\omega)}{\partial \omega_i} = \begin{cases} \frac{1}{2}\frac{\partial P(\omega)}{\partial \omega_i}; & i \text{ odd} \\ \frac{1}{2}\frac{\partial Q(\omega)}{\partial \omega_i}; & i \text{ even} \end{cases} . \tag{5.31}$$

The elements of the Jacobian matrix are then computed in the time and frequency domain using the following formulae:

$$\begin{aligned} j_i(n) &= \frac{\partial \alpha_n}{\partial \omega_i} \\ &= \begin{cases} [\sin(\omega_i)\delta(n-1)] * \left[\bigwedge_{k=0,\ k\neq\frac{i+1}{2}}^{p/2} \tilde{p}_k(n)\right]; & i \text{ odd} \\ [\sin(\omega_i)\delta(n-1)] * \left[\bigwedge_{k=0,\ k\neq\frac{i}{2}}^{p/2} \tilde{q}_k(n)\right]; & i \text{ even} \end{cases} \\ &\qquad\qquad \Updownarrow \\ J_i(\omega) &= \frac{\partial A(\omega)}{\partial \omega_i} \\ &= \begin{cases} \sin(\omega_i)e^{-jw}\prod_{k=0,\ k\neq\frac{i+1}{2}}^{p/2} \tilde{P}_k(\omega); & i \text{ odd} \\ \sin(\omega_i)e^{-jw}\prod_{k=0,\ k\neq\frac{i}{2}}^{p/2} \tilde{Q}_k(\omega); & i \text{ even} . \end{cases} \end{aligned} \tag{5.32}$$

Note that since

$$\bigwedge_{k=0,\ k\neq\frac{i+1}{2}}^{p/2} \tilde{p}_k(n) = \frac{p(n)}{\tilde{p}_i(n)} \tag{5.33}$$

for odd i, and

$$\bigwedge_{k=0,\ k\neq\frac{i}{2}}^{p/2} \tilde{q}_k(n) = \frac{q(n)}{\tilde{q}_i(n)} \tag{5.34}$$

for even i, the values of $j_i(n)$ can be found by simple polynomial division. Furthermore, the computation can be simplified further if we notice that the symmetry in the coefficients of $p(n)$ and $q(n)$ imply that

$$\frac{\partial \alpha_n}{\partial \omega_i} = \frac{\partial \alpha_{p+1-n}}{\partial \omega_i} \tag{5.35}$$

for odd i, and

$$\frac{\partial \alpha_n}{\partial \omega_i} = -\frac{\partial \alpha_{p+1-n}}{\partial \omega_i} \tag{5.36}$$

for even i.

The important factor in Eq. 5.32 is the term $\sin(\omega_i)$, which indicates the trend observed when analyzing the effect of quantizing the individual P-LSFs on recognition accuracy. Indeed, the term $\sin(\omega_i)$, which is smaller for the P-LSFs with low and high order and larger for the intermediate P-LSF frequencies, shows that there may be less sensitivity of the prediction coefficients with respect to the low and high order frequencies than with the middle P-LSF frequencies.

Numerical example

Figures 5.6 through 5.8 illustrate the three Jacobian matrices, J_C, J_A and J, respectively, computed by taking the mean of these matrices over 150 speech frames. Figure 5.6 plots only the first column of the Jacobian matrix J_C, since it is sufficient to fully determine the lower-triangular Toeplitz matrix J_C. Note the following:

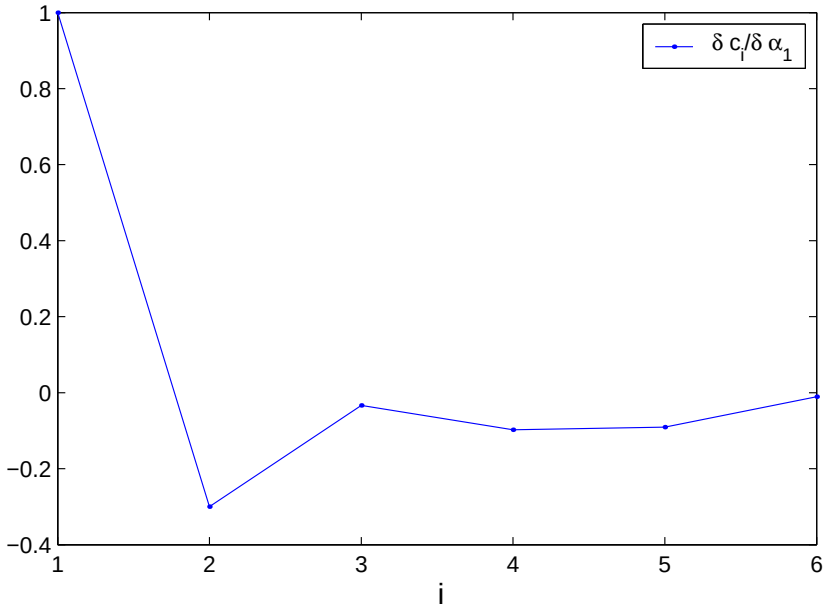


Figure 5.6: Illustration of the first column ($\frac{\partial c_i}{\partial \alpha_1}$) of the Jacobian matrix J_C.

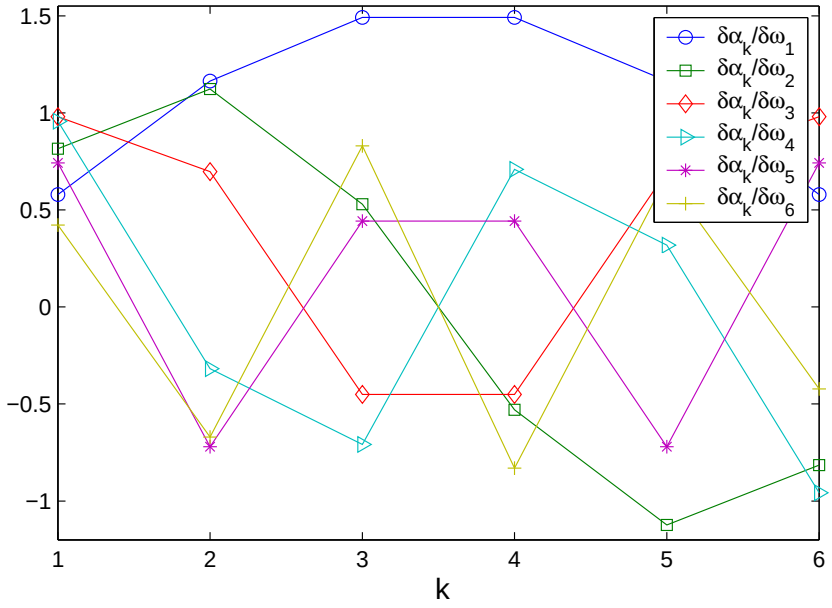


Figure 5.7: Illustration of the Jacobian matrix $J_A = \frac{\partial \alpha_k}{\partial \omega_j}$.

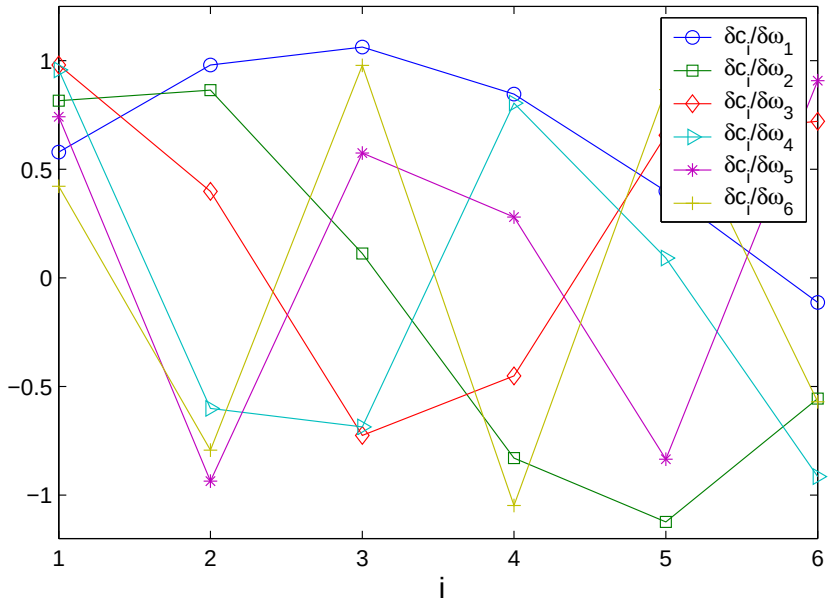


Figure 5.8: Illustration of the Jacobian matrix $J = \frac{\partial c_i}{\partial \omega_j}$.

1. The Jacobian matrix J_C is predominantly diagonal with the element of the first diagonal equal to one. This means that the Jacobian matrix J_C is almost equal to the identity matrix ($J_C \approx I$) and therefore $J \approx J_A$.

2. Observe the symmetry for the even columns of J_A and the anti-symmetry for the odd columns of J_A.

3. The illustration of the Jacobian matrix J_A seems to be displaying sinusoids whose frequencies are equal to the line spectral frequency ω_i in question. This phenomenon can be explained if we analyze the situation in the frequency domain. We have seen that the i^{th} column J_A is $J_{A_i}(\omega) = sin(\omega_i) \cdot \frac{P}{\tilde{P}_i(\omega)}$, which can be thought of as a spectrum that contains all the zeros of $P(\omega)$ at the even LSFs, except the one at frequency ω_{2i} which has been divided out. As a result, the spectrum $J_{A_i}(\omega)$ displays a large peak at the position of the missing zero (ω_{2i}) and resembles a dirac impulse. The

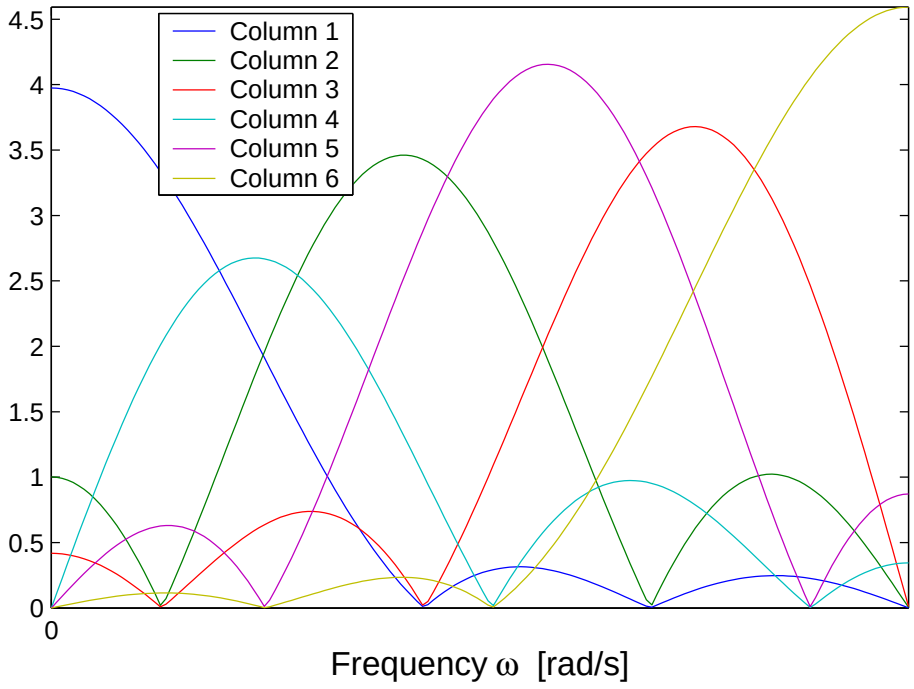


Figure 5.9: Spectra of the different columns of $J_A = \frac{\partial \alpha_k}{\partial \omega_j}$.

time domain counterpart of this spectrum is a sinusoid whose frequency is proportional to the LSF studied. For confirmation of this claim, Figure 5.9 illustrates the spectra for each column of the Jacobian matrix J_A. These results ($J_C \approx I$ and the sinusoidal behavior of J_A) are confirmed by analyzing the first six columns of the time-domain Jacobian matrix J of a high order (p=40) linear prediction system (Figure 5.10).

4. Figure 5.8 shows that the sensitivity of the first cepstral coefficients c_1 (which is assumed to be the most important for recognition) is about twice as high for ω_3 and ω_4 ($\frac{\partial c_1}{\partial \omega_3} \approx 1$) than for ω_1 and ω_6 ($\frac{\partial c_1}{\partial \omega_6} \approx 0.5$). This could help explain the behavior of Figure 5.4, which show higher sensitivity for mid-frequency range P-LSFs.

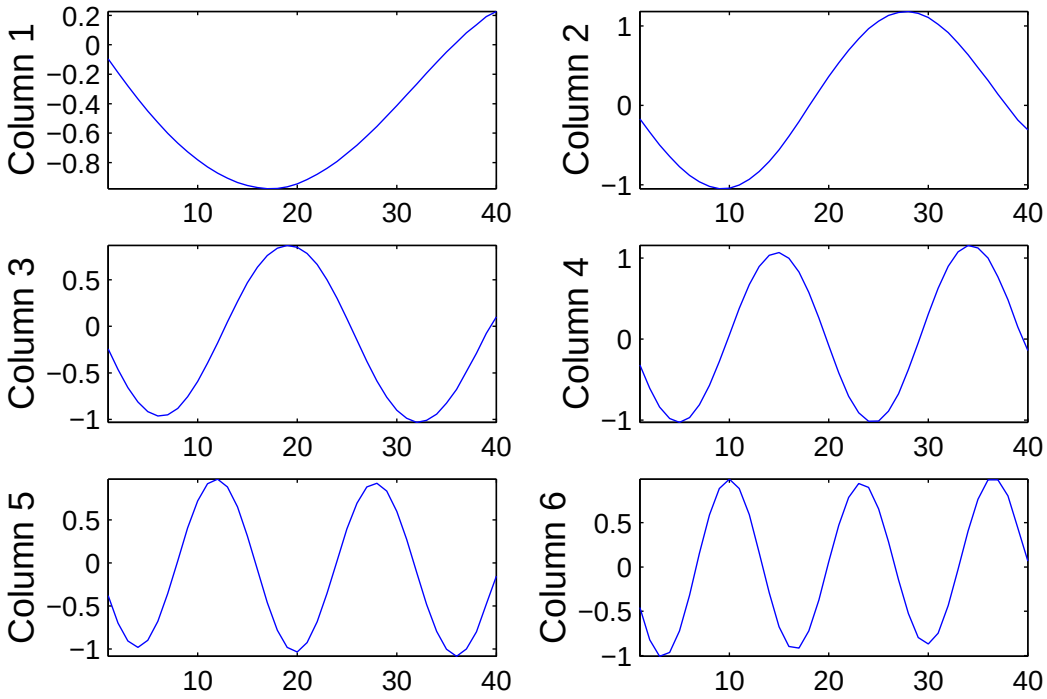


Figure 5.10: Spectra of the first six columns of $J_A = \frac{\partial \alpha_k}{\partial \omega_j}$ when using high order linear prediction.

5. Finally, observe that typically $|\frac{\partial c_i}{\partial \omega_j}| \leq 1$. This explains the results of Figure 5.4, which shows that the cepstral coefficients are less sensitive to quantization in the P-LSF domain than in the cepstral domain.

5.3.3 Quantization using perceptual line spectral frequencies

In the previous section, we showed that an efficient representation of the PLP spectrum for quantization could exploit the high inter- and intra-frame correlation of the line spectral frequencies of the perceptual linear prediction system. Furthermore, it was shown that quantizing P-LSFs typically yields a better representation of the low-order cepstral coefficients, which are more important for speech recognition. Finally, it was shown that error sensitivity of the P-LSFs to quantization noise is uneven for the different P-LSFs and more significant for the P-LSFs in the mid-frequency range. Appropriate weighting can then be performed when designing the vector quantizer and during the search procedure.

In this section, we design two types of quantization for coding the P-LSFs of the PLP spectrum, depending on the targeted application.

In the first application, we assume an error-free transmission. Since there is no risk for error propagation with error-free communication, predictive vector quantization (PVQ) can be used. We suggest the following quantization scheme for the P-LSFs of the 6^{th} order PLP spectrum: 1) remove the mean (DC component); 2) compute the residual P-LSFs after a first order moving average prediction whose coefficient is chosen to minimize the signal variance after prediction; 3) vector quantize the residual vector using different one stage vector quantizers operating at 3, 4, 5 and 6 bits depending on channel conditions. The search cost function to be minimized is weighted depending on the error sensitivity of each P-LSF. The P-LSFs are transmitted every 20 ms and interpolated every 10 ms.

Bits/frame	3	4	5	6	7
Source bit rate (bps)	150	200	250	300	350
Recognition accuracy (%)	81.07	97.15	98.33	98.61	98.74

Table 5.7: Continuous digit recognition accuracy using the Aurora-2 database after quantization of the P-LSFs of PLP_6 using first order predictive weighted VQ.

This results in a total bit rate of only 150 to 300 bits/second. Table 5.7 shows recognition results for continuous digit strings using the Aurora-2 database at different bit rates.

In order to appreciate the reduction in bit rate and the recognition improvement of P-LSF quantization with respect to speech coding based recognition, Figure 5.11 summarizes the results obtained in Tables 5.2, 5.3, and 5.7.

In the second application, we assume that we have to deal with error-prone communication channels, where errors may occur even after forward error correction (FEC). In such situations, it seems preferable to use a source coding scheme that does not use prediction to avoid memory mismatch between the encoder and the decoder. For the same reason, interpolation is avoided. Given the constraint that no prediction should be used, the dynamic range of the vector to quantize is larger, which in turn calls for more bits per vector. On the other hand, if one wants to keep the bit rate and simplicity of the full VQ scheme with one stage, it is necessary to lower the dimension of the feature vector. For this reason, we use for this application a 5^{th} order PLP spectrum (PLP_5), and the quantization scheme operates as follows. The five P-LSFs are computed and quantized every 10 ms using vector quantizers operating at 7 to 10 bits per frame. The receiver decodes the P-LSFs and computes the PLP coefficients (α_n) from the P-LSFs,

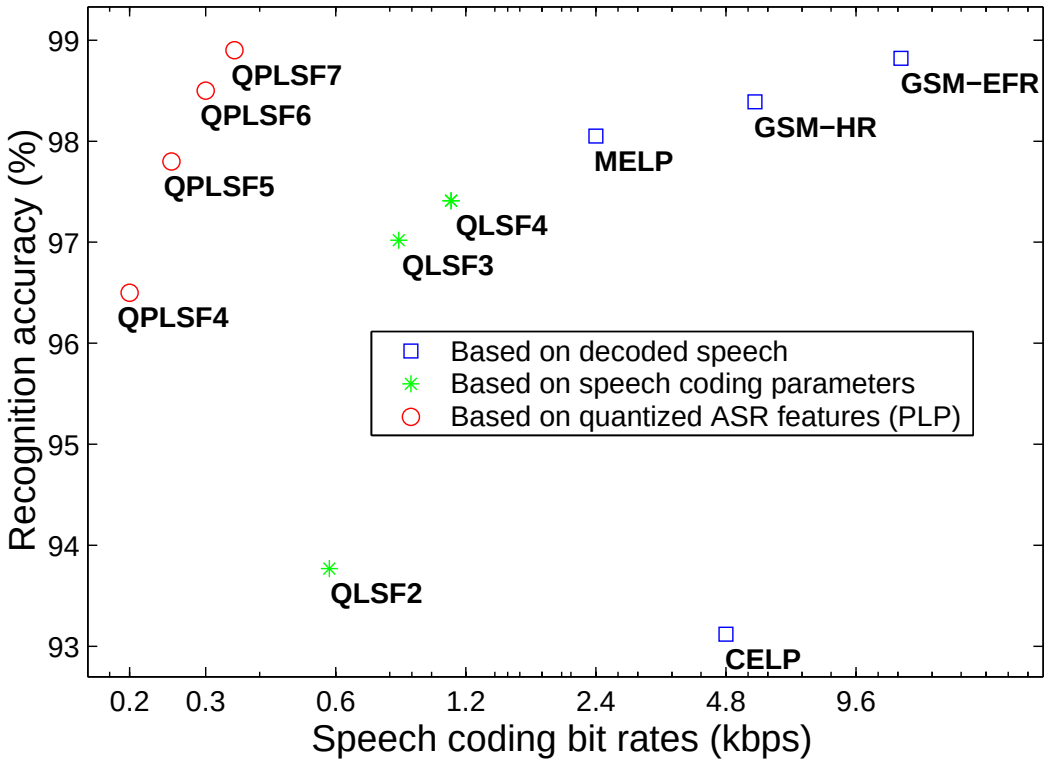


Figure 5.11: Illustration of speech recognition accuracy using different speech coding standards (squares), MELP based remote recognition using MSVQ quantization of the line spectral frequencies (stars), and quantized P-LSFs (circles), using the Aurora-2 database.

Bits/frame	7	8	9	10
Source bit rate (bps)	700	800	900	1000
Recognition accuracy(%)	97.07	98.05	98.31	98.81

Table 5.8: Continuous digit recognition accuracy using the Aurora-2 database after non-predictive vector quantization of the P-LSFs of PLP_5.

and the cepstral coefficients (c_n) from α_n according to Eq. 5.1. Table 5.8 reports continuous digit recognition results at different bit rates for this quantization scheme.

5.4 Quantization of ASR features: MFCC

Several recent papers addressed the issue of quantizing MFCC features. In [103, 104, 121], split vector quantization of MFCCs is shown to provide good recognition accuracy at about 2-4 kbps. A similar technique is used in [106] to provide recognition at 4 kbps. In the ETSI standard [104], MFCCs are also split vector quantized and transmitted at 4.8 kbps. Finally, [122] exploits redundancy of MFCC parameters using a 2-D Discrete Cosine Transform (DCT).

Before describing our own techniques for quantizing the MFCC feature vectors, we first suggest an alternative strategy for transmitting MFCCs to the client by quantizing a mathematically equivalent set of values, in the same way we quantized in the previous section the P-LSFs instead of the P-LPCCs. This time, the mathematically equivalent feature is the inverse DCT of the MFCCs,

$$\tilde{m}_i = \sum_{j=1}^{N} c_j \cos\left(\frac{\pi i}{N}(j-0.5)\right) \qquad 1 \le i \le N. \tag{5.37}$$

The resulting N coefficients will be referred to as inverted cepstra (ICP).

Since MFCCs are obtained in the first place by taking the DCT of the log Mel

Coeff.	0	1	2	3	4	5	6	7	8	9	10	11	12
ICP	0.93	0.93	0.93	0.92	0.92	0.92	0.92	0.91	0.91	0.91	0.92	0.90	0.90
MFCC	0.93	0.88	0.83	0.81	0.77	0.75	0.70	0.67	0.65	0.65	0.64	0.61	0.65

Table 5.9: Average (across all digits) inter-frame correlations between the 13 MFCCs and ICPs, using 25 ms Hamming windows shifted every 10 ms.

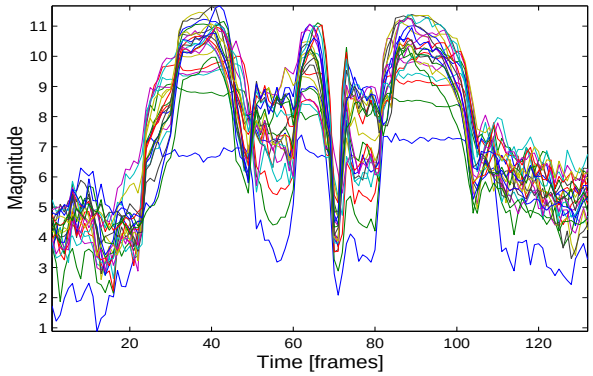
filter outputs (Eq. 1.3), ICPs conceptually represent again a log Mel spectrum, but with less coefficients than the original Mel filterbank outputs ($N < M$). Typically, for a narrowband speech signal, $N = 13$ MFCCs (including the energy term c_0) are computed from a set of $M = 23$ Mel filter outputs.

Figure 5.12 illustrates the log Mel filterbank outputs, the MFCC and the inverted cepstra for the digit string "9 6 0" pronounced by a male speaker. Figure 5.12(c) hints that inverted cepstra display interesting properties that can be exploited when compressing the signal, such as large inter- and intra-frame correlation. Quantitative results are shown in Table 5.9 for the inter-frame correlation and in Table 5.10 for the intra-frame correlation.

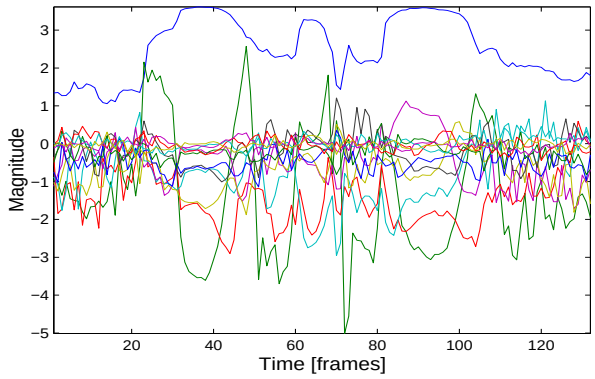
Table 5.9 indicates that ICPs are more time correlated than MFCCs since they are directly related to the amount of energy in each Mel frequency band; energy variations are dictated by the relatively slow process of articulation. Since the time-correlation is significant, efficient source compression will be obtained by performing predictive coding. Each parameter can be individually predicted using a first order linear prediction scheme.

Table 5.10 reports the mean of the N diagonals of the intra-frame correlation matrix instead of the full intra-vector correlation matrix after first-order prediction given its size.

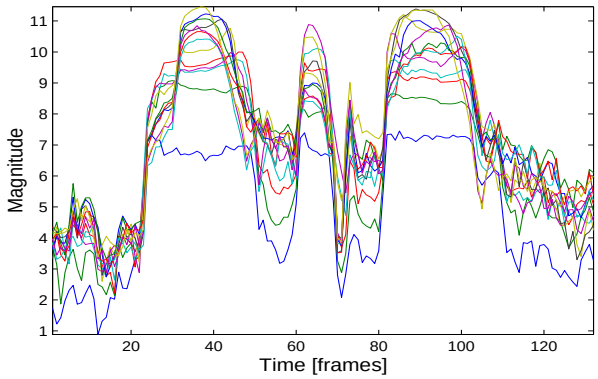
Note that the large intra-frame correlation of the inverted cepstra is a di-



(a) Log Mel filter output



(b) Mel Frequency cepstral coefficients



(c) Inverted cepstra (ICP)

Figure 5.12: Illustration of the (a) log energy outputs of the Mel filterbank (M=23); (b) Mel frequency cepstral coefficients (N=13); and (c) inverted cepstra (N=13) for the digit string "9 6 0" pronounced by a male speaker.

Diag. #	0	1	2	3	4	5	6	7	8	9	10	11	12
ICP	1.00	0.86	0.77	0.73	0.68	0.65	0.62	0.59	0.55	0.52	0.45	0.38	0.32
MFCC	1.00	0.07	0.04	0.05	0.03	0.06	0.03	0.03	0.04	0.08	0.09	0.08	0.06

Table 5.10: Mean of the elements of the i^{th} diagonal of the average (across all digits) intra-frame correlation matrix $\rho(i, j)$ for the 13 ICPs and MFCCs obtained after first order prediction.

rect consequence of the fact that neighboring Mel frequency bands are bound to display analogous behavior. On the other hand, the small intra-vector correlation displayed by the MFCCs is the result of taking the DCT of the log-Mel filterbank output in the first place, which is done to guarantee statistical independence. This has the dual advantage of 1) feeding the HMM recognizer with non-redundant information and 2) permitting the use of diagonal covariance instead of full covariance matrices for multivariate Gaussian modeling.

In the section on PLP quantization, we have seen that correlations displayed by P-LSFs could be converted into coding gains when compared to quantizing the actual P-LPCC recognition features. Such analysis is repeated in the next section for MFCCs. We investigate whether correlations displayed by ICPs can yield to high recognition accuracy at reduced bit rates.

5.4.1 MFCC quantization

Typically, an MFCC speech recognition feature vector consists of 12 MFCCs (c_1,...,c_{12}), to which might be added a log-energy component ($\log(E)$). MFCCs are computed every 10 ms using a 25 ms analysis window. This overlap results in a high correlation between adjacent frames, as illustrated in Table 5.9. This correlation can be exploited in speech recognition systems. Specifically, the client

SNR	E	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}
-5 dB	7.4	19.1	33.9	26.1	43.8	60.5	72.5	55.9	59.9	72.1	75.4	88.8	86.5
0 dB	10.3	61.1	72.1	88.4	89.1	93.6	95.9	96.0	96.5	96.9	97.2	97.6	97.6

Table 5.11: Recognition accuracy after quantization noise is added to each individual feature, one at a time. Quantization SNRs are expressed in dB.

can compute and transmit features every 20 ms, while the server interpolates the features by a factor of 2 for recognition. This results in lower bit rate and complexity at the client.

Furthermore, due to the nature of the speech signal itself, there is evidence of a remaining correlation between adjacent frames even if MFCCs are computed every 20 ms. This is captured in our coding scheme using first order predictive coding, which provides on average 4 dB of coding gain. The MFCCs can then be efficiently quantized as follows: 1) remove the mean of each feature; 2) compute the residual feature after first order linear prediction whose coefficient is chosen to minimize the signal variance after prediction; 3) split the residual vector into two subvectors and vector quantize them using different rates depending on channel conditions. Note that the cost function to be minimized during VQ training and VQ search is weighted to take into account quantization sensitivities of each individual feature. Degradations in recognition as a function of quantization noise SNRs are shown in Table 5.11 for each feature.

Table 5.12 reports continuous digit recognition accuracy when quantizing MFCCs (with and without energy) at different bit rates for each VQ split. When using MFCCs without energy, the feature vector is split [c_1-c_6] and [c_7-c_{12}]. If energy is added (MFCC_E), the feature vector is split [log(E),c_1-c_5] and [c_6-c_{12}]. Two different cases are analyzed: 1) when training and testing are done on quantized features (MFCC_E$_Q$ and MFCC$_Q$) and 2) when training is done using

bits/sec.	bits/frame	MFCC_E$_Q$	MFCC$_Q$	MFCC_E	MFCC
Unquantized	—	99.20	98.86	99.20	98.86
900	9+9	98.73	98.09	97.96	96.51
800	8+8	98.61	98.18	97.87	96.91
700	7+7	98.13	97.56	97.56	95.71
600	6+6	97.12	96.98	97.22	94.94
500	5+5	96.35	95.93	96.48	93.49
400	4+4	94.75	93.16	93.27	91.45

Table 5.12: Continuous digit recognition accuracy using the Aurora-2 database after quantizing the MFCCs using first order predictive weighted split VQ. Notation 8+8 means 8 bits for the first split and 8 bits for the second. Subscripts $_Q$ indicate that the HMM models have been trained on quantized features.

unquantized data and testing using quantized data (MFCC_E and MFCC).

Note that one can get reasonable recognition accuracies with rates as low as 700 bps. Below this, however, prediction starts degrading and recognition drops significantly. Note also that despite the additional vector dimension to quantize, MFCC_E always outperforms MFCC.

5.4.2 Inverted cepstra quantization

A similar quantization procedure (weighted split VQ after first order prediction) is used to quantize the inverted cepstra. Table 5.13 reports continuous digit recognition accuracy when quantizing ICPs (with and without energy) at different bit rates for each VQ split. The same scenarios are analyzed, with and without energy, as well as training on quantized or unquantized data.

One can see from Tables 5.12 and 5.13 that recognition accuracies are comparable at high bit rates when quantizing ICPs and CEPs, and worse for ICPs at low bit rates. The lack of superior behavior of ICPs over CEPs despite its more

bits/sec.	bits/frame	ICP_E$_Q$	ICP$_Q$	ICP_E	ICP
Unquantized	—	99.20	98.86	99.20	98.86
900	9+9	98.86	98.40	98.33	96.95
800	8+8	98.67	98.43	97.99	96.88
700	7+7	98.55	97.99	97.90	96.58
600	6+6	97.10	97.75	95.62	95.10
500	5+5	96.08	95.43	93.89	90.71
400	5+5	91.21	92.19	86.86	83.79

Table 5.13: Continuous digit recognition using the Aurora-2 database after quantizing the inverted cepstra (ICP) using first order predictive weighted split VQ. Notation 8+8 means 8 bits for the first split and 8 bits for the second. Subscripts $_Q$ indicate that the HMM models have been trained on quantized features.

attractive correlation properties can be explained by analyzing the figure of merit of each quantization scheme for each cepstral coefficient. Figure 5.13 illustrates the signal-to-noise ratio for each individual cepstral coefficient resulting from the quantization of ICPs and CEPs at different bit rates.

When quantizing the cepstral coefficients (CEP), one can see that the first energy term (c_0) and the spectrum tilt term (c_1) display higher quantization accuracy and that the remaining coefficients show about the same level of precision. Given the increased sensitivity for recognition of the low-order cepstral coefficients, this fact is actually beneficial for recognition.

While this trend holds true for inverted cepstra, one can see that the high-order cepstra, while less important for recognition accuracy, are too coarsely reconstructed to provide good recognition. Furthermore, the level of precision in the reconstruction of the low-order cepstra goes beyond what is necessary for recognition purposes. In fact, one can see that reconstruction accuracy almost monotically decreases with the cepstrum order. This can be easily interpreted if

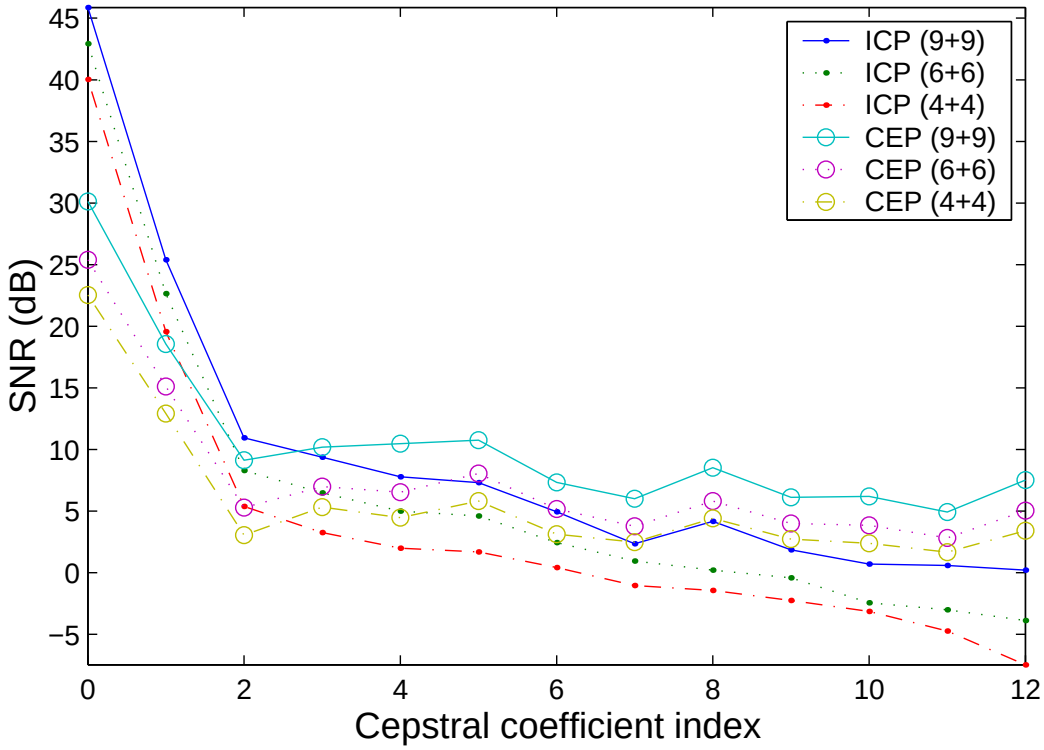


Figure 5.13: Quantization SNRs for each cepstral coefficient after predictive split vector quantization of the cepstral coefficients (o) and the inverted cepstra (.) using 9, 6 and 4 bits per split.

we recall that in this case the resulting cepstra are obtained after taking the DCT of the quantized ICPs. While c_0 averages out the noise introduced by quantizing ICPs, higher order cepstra suffer from quantization error.

5.5 Summary

We analyzed in this chapter three source coding approaches on which remote recognition can be based: 1) using features obtained from the decoded speech signal; 2) using transformed speech coding parameters; 3) using specifically quan-

tized ASR features.

Second, we presented different solutions for quantizing two types of ASR features: PLP and MFCC.

For the quantization of PLP features, transmission of the perceptual line spectral frequencies instead of the PLP cepstral coefficients constituted a sensible choice both in terms of coding gain and recognition accuracy. A mathematical and experimental analysis of the Jacobian matrices of the two non-linear operations permitting the transformation from the P-LSFs to cepstral coefficients were presented.

For the quantization of the MFCC coefficients, quantization of the cepstral coefficients and the inverted cepstra were shown to offer similar recognition performance, with a slight edge for quantization of the cepstral coefficients.

CHAPTER 6

Channel coding and decoding for remote speech recognition

This chapter is divided into four sections. Section 6.1 analyzes the effect of channel errors and erasures on speech recognition accuracy, and derives channel coding requirements specific to remote recognition applications. Section 6.2 presents channel encoders that meet these requirements. Section 6.3 presents different channel decoding techniques and illustrates the advantage of performing soft decision based error detection channel decoding. Finally, Section 6.4 discusses the performance of different channel coding and decoding systems over a wide range of channel conditions.

6.1 The effect of channel errors and erasures on remote speech recognition

In this section, we study how channel errors and erasures affect the Viterbi speech recognizer. The following two sections analyze how channel coding and decoding can be performed to deal with such errors and erasures.

6.1.1 The effect of channel errors and erasures

The emphasis in remote ASR is typically recognition accuracy and not playback. The nature of this task implies different criteria for designing channel encoders and decoders than those used in speech coding applications.

Recognition is achieved by accumulating feature vectors over time, and by selecting the element in the dictionary that most likely produced that sequence of observations. The likelihood of observing a given sequence of features given a hidden Markov model (HMM) is computed by searching through a trellis for the most probable state sequence. The Viterbi algorithm (VA) presents a dynamic programming solution to find the most likely path through a trellis (Figure 6.1). For each state j, at time t, the likelihood of each path is computed by multiplying the transition probabilities a_{ij} between states and the output probabilities $b_j(\boldsymbol{o}_t)$ along that path. The partial likelihood, denoted as $\phi_{j,t}$, is computed efficiently using the following recursion:

$$\phi_{j,t} = \max_i [\phi_{i,t-1}\ a_{ij}]\ b_j(\boldsymbol{o}_{t-1}). \tag{6.1}$$

The probability of observing the N_F-dimensional feature $\boldsymbol{o}_t$ is:

$$b_j(\boldsymbol{o}_t) = \sum_{m=1}^{N_M} c_m \frac{1}{\sqrt{(2\pi)^{N_F}|\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\boldsymbol{o}_t-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{o}_t-\boldsymbol{\mu})}, \tag{6.2}$$

where N_M is the number of mixture components, c_m is the mixture weight, and the parameters of the multivariate Gaussian mixture are its mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

Figure 6.2(a) analyzes the effect of a channel *error* in the VA. Assume first a transmission without channel errors or erasures; the best path through the trellis is given by the solid line. Assume now that a channel error occurs at time t. The decoded feature is then $\hat{\boldsymbol{o}}_t$, as opposed to $\boldsymbol{o}_t$, and the associated probabilities

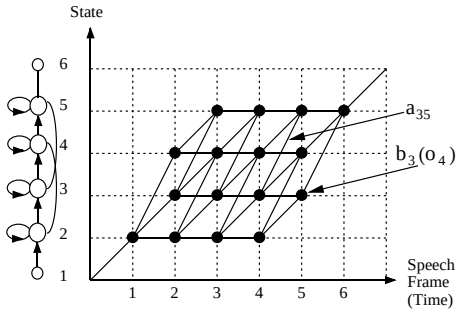
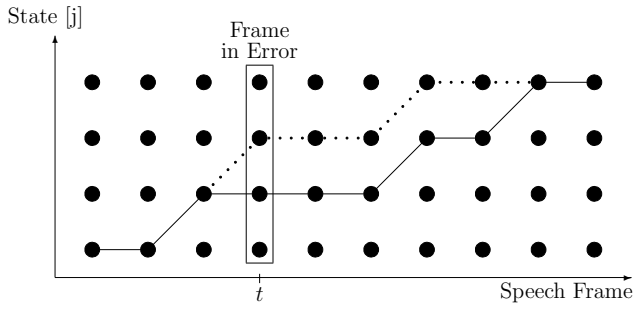


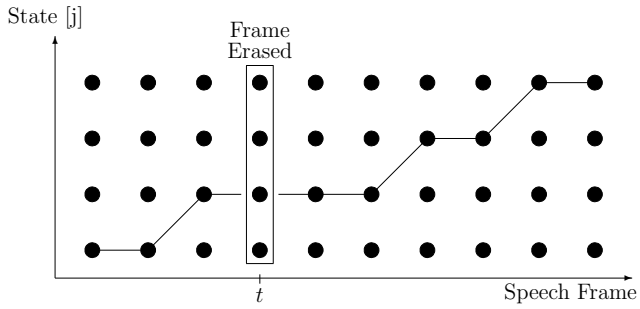
Figure 6.1: Illustration of the Viterbi speech recognition algorithm (after [3]).

for each state j may differ considerably ($b_j(\hat{\boldsymbol{o}}_t) \neq b_j(\boldsymbol{o}_t)$), which will disturb the state metrics $\phi_{j,t}$. A large discrepancy between $b_j(\hat{\boldsymbol{o}}_t)$ and $b_j(\boldsymbol{o}_t)$ can disturb the information accumulated thus far for each state metric. Furthermore, note that the observation probability $\hat{\boldsymbol{o}}_t$ may force the best path in the trellis to branch out (dotted path) from the error-free best path. Consequently, some features may be accounted for in the overall likelihood computation using the state model $\hat{j}$ instead of the correct state model j, which again will modify the probability of observation since $b_{\hat{j}}(\boldsymbol{o}_{t'}) \neq b_j(\boldsymbol{o}_{t'})$.

On the other hand, it can be seen that channel *erasures* may have little effect on recognition performance. Figure 6.2(b) illustrates the same sequence of observations as in Figure 6.2(a), but with a channel erasure. State metrics are not disturbed with a channel erasure since the probability of the missing observation cannot be computed. Not updating the state metrics ($\phi_{j,t} = \phi_{j,t-1}$) is not as likely to create a path split between the best paths (with and without an erasure) as a channel error, whether or not the erasure occurs at a state transition. Hence, channel erasures typically do not propagate through the trellis.



(a) Channel error at time t



(b) Channel erasure at time t

Figure 6.2: Illustration of the effect of (a) a frame error and (b) a frame erasure on Viterbi speech recognition.

6.1.2 Recognition experiment with channel errors and erasures

In this section, we simulate the effects of channel erasures and channel errors in a speech recognition task. Speech recognition experiments consist of continuous digit recognition based on 4 kHz bandwidth speech signals. Training is done using speech from 110 males and females from the Aurora-2 database [104] for a total of 2200 digit strings. The feature vector consists of 5 PLP cepstral coefficients. Word HMM models contain 16 states with 6 mixtures each, and are trained using the Baum-Welch algorithm assuming a diagonal covariance matrix. Recognition tests contain 1000 digit strings spoken by 100 different speakers (male and female) for a total of 3241 digits. Recognition results reported are in word accuracy, which

is computed in percent as:

$$\mathrm{ACC} = 100 \cdot \frac{\mathrm{REC} - \mathrm{INS} - \mathrm{DEL}}{\mathrm{N}}, \tag{6.3}$$

where ACC is the recognition accuracy, REC is the number of tokens correctly recognized, INS is the number of insertions made, DEL is the number of digits deleted, and N is the number of tokens tested.

The first experiment analyzes in detail the effect of a channel erasure or error on three aspects of the Viterbi recognition algorithm: 1) dynamic search for the most likely trellis paths, 2) computation of the average probability of observing the features while in those given states, and 3) computation of the overall accumulated likelihood of observing the token given the trellis paths. These aspects are analyzed in Figure 6.3 for the trellis paths, in Figure 6.4 for the average probability of observing the features in a given state, and in Figure 6.5 for the accumulated likelihood. For each figure, three cases are analyzed: 1) no error or erasure, 2) one channel erasure and 3) one channel error. For the last two cases, the erasure or error takes place at the 17^{th} frame. The task consisted of the likelihood computation of an observation sequence (60 frames) given its word HMM model.

Figure 6.3 shows that the erasure did not create a significant path disturbance, except for being shifted in time by one frame. The consequence of the channel error was to significantly disturb the most likely path, both before and after the error took place.

The resulting effect on the average probabilities of observing the feature sequence in any given frame is considerable in the case of a channel error, as shown in Figure 6.4. The consequence on the *overall* likelihood computation, shown in Figure 6.5, is that while the channel erasure only moderately changes the overall likelihood of observing the token, the channel error occurring at the same time

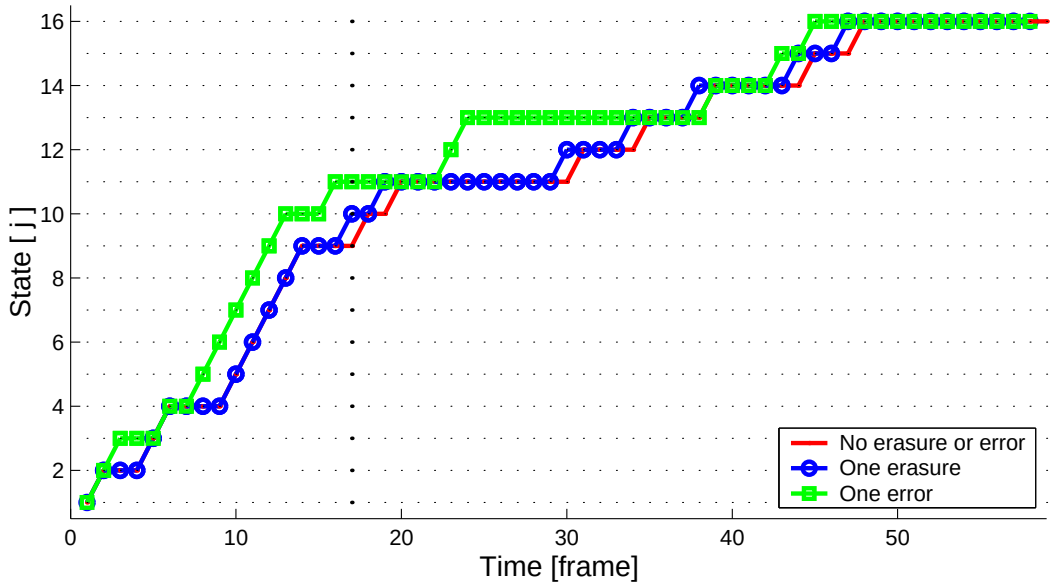


Figure 6.3: Illustration of the consequence of a channel erasure and error on the most likely paths taken in the trellis by the received sequence of observations, given a 16-state word digit model. The erasure and error occur at frame number 17.

frame modifies the likelihood results considerably, hence increasing the chance of a recognition error.

The second experiment performs a complete recognition task when the observations are impaired by channel noise. Figure 6.6 illustrates the effect of randomly inserted channel erasures and errors in the communication link between the client and the server. The feature vector transmitted consists of 5 PLP cepstral coefficients (P-LPCC), and is computed every 10 ms. Recognition is performed with the time-derivative and acceleration of the PLP coefficients. Computation of the temporal features at the receiver accentuates error propagation. Figure 6.6 shows that channel errors, which propagate through the trellis, have a disastrous effect on recognition accuracy, even at less than 1%, while the recognizer is able to

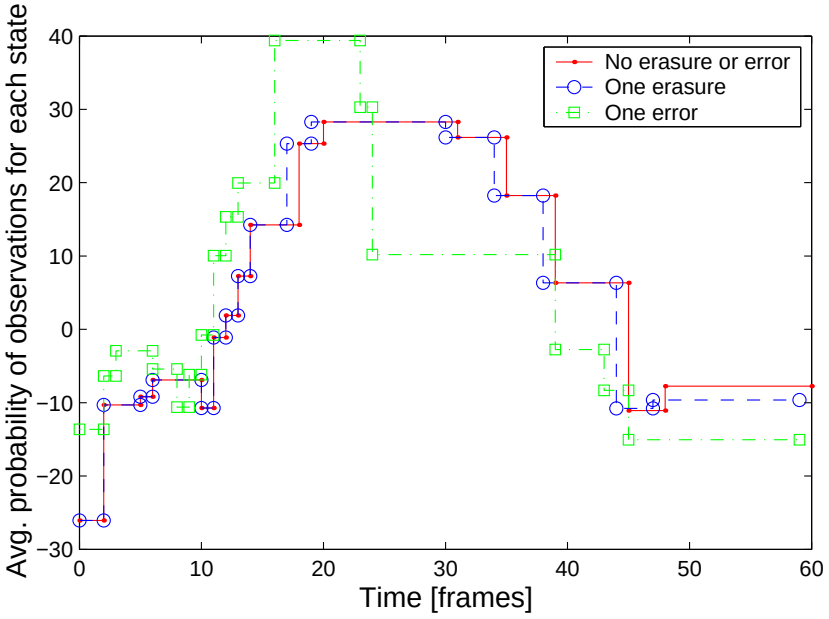


Figure 6.4: Illustration of the consequence of a channel erasure and error on the average probability of observing the features in each state of the trellis, given a 16-state word digit model. The erasure and error occur at frame number 17.

operate with almost no loss in accuracy with up to 10% of channel erasures. This confirms our results presented in [105] for isolated digit recognition based on PLP coefficients and in [58] for MFCCs.

These results indicate that when designing channel coders for remote recognition applications, the emphasis should be on error detection more than on error correction. The remainder of this chapter will investigate innovative techniques that maximize error detection capabilities of linear block codes suitable for remote speech recognition applications.

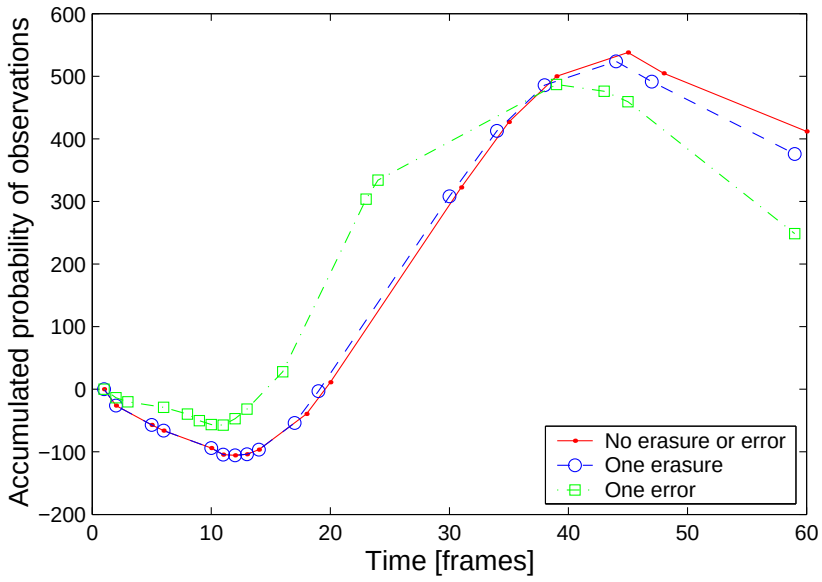


Figure 6.5: Illustration of the consequence of a channel erasure and error on the accumulated probability of observation, given a 16-state word digit model. The final accumulated likelihoods represent the probability of observing the complete sequence of observations given the model.

6.1.3 Channel erasure models

Two types of erasure models are analyzed. In the first type, channel erasures occur independently, with a given probability of erasures. In the second type, channel erasures occur in bursts. This is typically the case for wireless or IP based communication systems, where correlated fading or network congestion may cause the loss of consecutive frames.

A classic model for bursty channels is the Gilbert-Elliot model [123] in which the transmission is modeled to be a Markov system where the channel is assigned to be in either one of two states: "0" for *good* and "1" for *bad*. Figure 6.7 illustrates a Gilbert channel model. With such a model, there is a probability $P_G = \frac{P_{BG}}{P_{BG}+P_{GB}}$

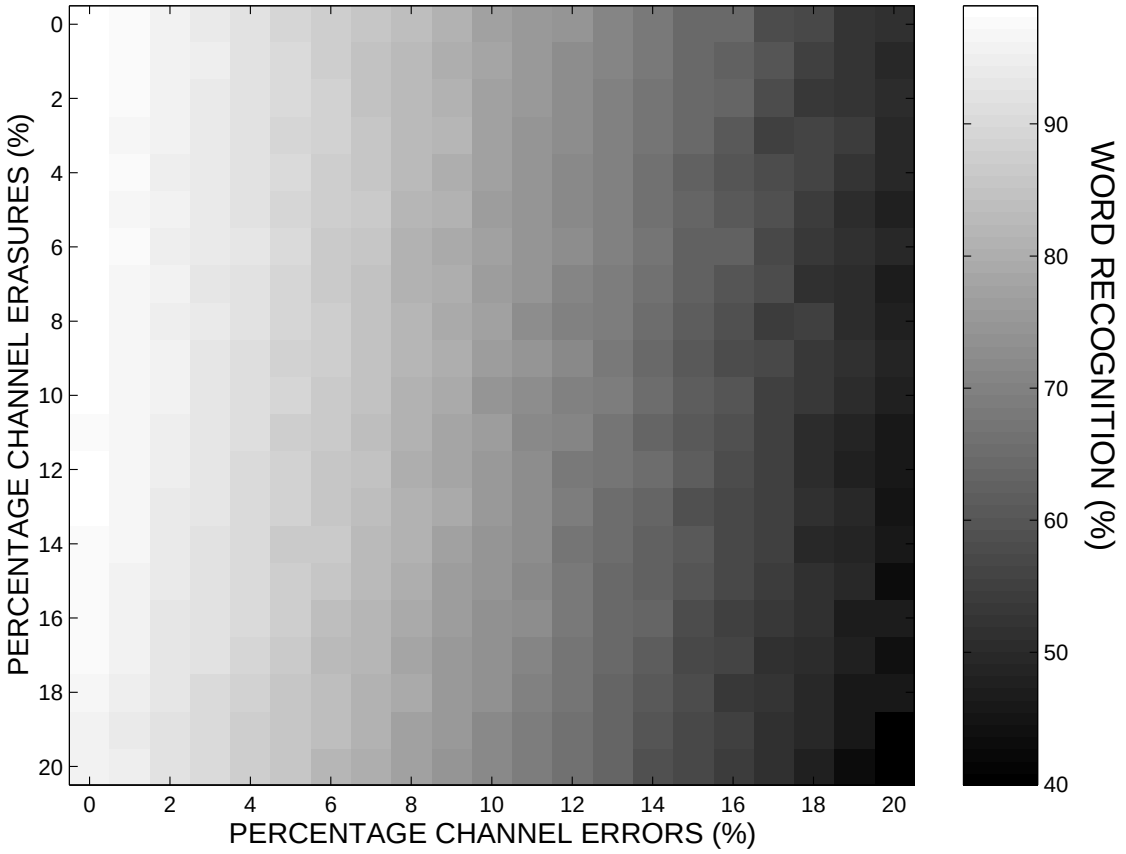


Figure 6.6: Simulation of the effect of channel erasures and errors on continuous digit recognition performance using the Aurora-2 database and PLP features. Recognition accuracies are represented in percent on a gray scale.

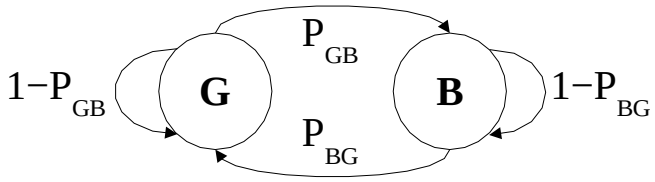


Figure 6.7: State diagram for the Gilbert-Elliot bursty channel.

(P_{GB}, P_{BG})	(2.5,20)	(2.5,15)	(5,20)	(2.5,10)	(1.25,5)	(5,15)	(10,20)	(5,10)
P_B	11.1	14.3	20.0	20.0	20.0	25.0	33.3	33.3
P_G	88.9	85.7	80.0	80.0	80.0	75.0	66.7	66.7
P_E	17.8	20.0	24.3	24.3	24.3	27.5	33.3	33.3
$\overline{L_b}$	5.0	6.6	5.0	10.0	20.0	6.6	5.0	10.0

Table 6.1: Characteristics of the Gilbert-Elliot channels of interest. Probabilities are given in percent.

to be in state "0" and a probability $P_B = \frac{P_{GB}}{P_{GB}+P_{BG}}$ to be in state "1". If the probabilities of channel erasures are assigned to be P_{E_G} for the good state and P_{E_B} for the bad state, the overall average probability of erasure is: $P_E = P_G P_{E_G} + P_B P_{E_B}$.

Throughout the experiments, P_{E_G} is considered to be equal to 10% and P_{E_B} is set to 80%. Different types of bursty channels are analyzed, depending on the state transition probabilities P_{GB} and P_{BG}, which in turn determine how bursty the channel is. Table 6.1 summarizes the properties of the bursty channels studied, including the probability (in percent) of being in the bad state (P_B) or in the good state (P_G), the overall probability of erasure (P_E), and the average length (in frames) of a burst of erasures ($\overline{L}_b$).

6.2 Channel coding for remote speech recognition applications

Section 6.1 indicated that an important requirement for a channel coding scheme for remote ASR is low probability of undetected error ($P_{UE} << 1\%$) and large enough probability of correct decoding ($P_{CD} > 90\%$). The next two sections present channel coding and decoding strategies that meet these requirements under a variety of channel conditions.

For packet based transmission, frames are typically either received or lost but not in error. If a packet order number was assigned before transmission, frame erasures can be detected by analyzing the ordering of the received packet. Therefore, for IP systems, there may not be a need for sophisticated error detection techniques.

With wireless communications systems, however, the transmitted bits $\boldsymbol{x}$ can be altered during transmission. Based on the values of the received bits $\boldsymbol{y}$, the receiver can either correctly decode the message (CD for correct decoding), detect a transmission error (ED for error decoding) or fail to detect such an error (UE for undetected error).

Since the number of source information bits necessary to code each frame can be as low as 6-40 bits/frame for efficient speech recognition feature coding schemes [105], linear block codes are favored over convolutional or trellis codes for delay and complexity considerations, as well as for their ability to provide error detection for each frame independently.

6.2.1 Description of error detecting linear block codes

An (N, K) linear block code maps K information bits into N bits ($N \geq K$) in such a way as to maximize the minimum distance between valid codewords in the N-dimensional space. The larger the number of redundancy bits $(N - K)$, the larger the minimum distance (d_{min}) between any two of the 2^K valid codewords.

In order to guarantee the best possible recognition rate over a wide range of channel conditions, a combination of different block codes is used. More information bits (K) are used for high SNR channels while more redundancy bits $(N - K)$ are used for low SNR channels. With such adaptive schemes, graceful degradation in recognition performance is provided with decreasing channel quality.

In the proposed design, we want to minimize the probability of undetected errors since it significantly reduces recognition accuracy. Therefore, one would like to find codes that maximize the probability of error detection while still guaranteeing a large enough probability of correct decoding.

For good channel conditions, Single Error Decoding (SED) codes, which detect any one bit error event, are sufficient. A minimum Hamming distance of $d_{min} = 2$ is necessary and sufficient to form an SED code. SED codes can be obtained using simple Cyclic Redundancy Check (CRC) codes. For instance, when a single parity bit is added to the information codeword (N-K=1), the minimum Hamming distance between any two valid codewords is always $d_{min} = 2$.

However, when there are 2 errors among the N received bits, SED codes may fail to detect the error. To increase channel protection, Double Error Detection (DED) codes are utilized. Any linear block code with $d_{min} = 3$ can be used to correct single error events (Single Error Correcting (SEC) code), or to detect

all one- and two-bit error events (Double Error Detection (DED) code). For our application, since residual channel errors degrade recognition accuracy more significantly than channel erasures, all codes with $d_{min} = 3$ will be used as DED codes. Finally, codes with $d_{min} = 4$, which can also be used as SEC/DED codes, will be utilized as Triple Error Detecting (TED) codes.

6.2.2 The search for "good" codes

Let C be an (N, K) linear block code. Let A_i be the number of codevectors of Hamming weight i in C. The numbers $A_0, A_1, \ldots, A_N$ are referred to as the *distance spectrum* of the code C. By linearity of the block code, d_{min} corresponds to the smallest non-zero index i such that $A_i \neq 0$.

Exhaustive searches over all possible linear block codes were run for all dimensions of interest, *i.e.* $7 \leq K \leq 10$ and $8 \leq N \leq 12$, in order to find the codes with the best distance spectrum.

For the particular case of N-K=1, *i.e.* a (K+1, K) code, the parity matrix P of dimension 1×K of the code is given by P = [1, 1, $\cdots$, 1, 1] and the distance spectrum by $A_i = \binom{N}{i}$ for i even and $A_i = 0$ for i odd.

The parity matrices P, the minimum Hamming distance d_{min} and the distance spectra A_i of all other codes of interest are given in Table 6.2. Parity matrices (dimensions (N-K)×K) are given in hexadecimal notation. The generator matrix can be obtained from the parity matrix.

Note that a subset of these codes can be obtained from some special codes. For instance, the (10,8) and (10,7) codes can be obtained by expurgating and shortening the extended (15,11) Hamming code.

(N,K)	N-K	P	d_{min}	Type	A_0, A_1, A_2, $\cdots$, A_N
(12,10)	2	1,1,1,2,2,2,3,3,3,3	2	SED	1,0,18,64,111,192,252,192,111,64,18,0,1
(12,9)	3	1,2,3,3,4,5,5,6,7	2	SED	1,0,5,34,66,88,114,108,61,24,9,2,0
(11,9)	2	1,1,1,2,2,2,3,3,3	2	SED	1,0,15,48,74,112,126,80,37,16,3,0
(12,8)	4	3,5,6,9,A,D,E,F	3	DED	1,0,0,16,39,48,48,48,39,16,0,0,1
(11,8)	3	1,2,3,4,5,6,7,7	2	SED	1,0,4,25,46,52,52,46,25,4,0,1
(10,8)	2	1,1,1,2,2,3,3,3	2	SED	1,0,12,36,46,60,60,28,9,4,0
(12,7)	5	07,0B,0D,0E,13,15,19	4	TED	1,0,0,0,38,0,52,0,33,0,4,0,0
(11,7)	4	3,5,6,9,A,D,E	3	DED	1,0,0,12,26,28,24,20,13,4,0,0
(10,7)	3	1,2,3,4,5,6,7	2	SED	1,0,3,19,29,27,25,17,6,1,0
(9,7)	2	1,1,2,2,3,3,3	2	SED	1,0,9,27,27,27,27,9,0,1
(11,6)	5	07,0B,0D,13,15,19	4	TED	1,0,0,0,25,0,27,0,10,0,1,0
(10,6)	4	3,5,6,9,E,F	3	DED	1,0,0,8,18,16,8,8,5,0,0
(9,6)	3	2,3,4,5,6,7	2	SED	1,0,2,14,18,12,10,6,1,0
(8,6)	2	1,1,2,2,3,3	2	SED	1,0,7,18,15,12,9,2,0
(10,5)	5	07,0B,13,1D,1E	4	TED	1,0,0,0,10,16,0,0,5,0,0
(9,5)	4	3,5,9,E,F	3	DED	1,0,0,4,14,8,0,4,1,0
(8,5)	3	1,3,5,6,7	2	SED	1,0,1,10,11,4,3,2,0
(7,5)	2	1,1,2,3,3	2	SED	1,0,5,12,7,4,3,0
(10,4)	6	07,1B,2B,35	4	TED	1,0,0,0,2,8,4,0,1,0,0
(9,4)	5	07,0B,13,1D	4	TED	1,0,0,0,6,8,0,0,1,0
(8,4)	4	7,B,D,E	4	TED	1,0,0,0,14,0,0,0,1
(7,4)	3	3,5,6,7	3	DED	1,0,0,7,7,0,0,1
(6,4)	2	1,2,3,3	2	SED	1,0,3,8,3,0,1

Table 6.2: Characteristics of the linear block codes that can be used for channel coding of ASR features. Acronyms SED, DED and TED stand for Single, Double and Triple Error Detection, respectively.

6.3 Channel decoding for remote speech recognition applications

In the previous section, we defined good linear block codes for error detection. In this section, we will analyze different channel decoding techniques that maximize the error detection capability of such codes.

For wireless communications, the information symbol x_i is transmitted and distorted by the channel, and the received symbol y_i is $y_i = \alpha(t) \cdot x_i + n(t)$, where $\alpha(t)$ is the complex channel gain and $n(t)$ is the additive white Gaussian noise (AWGN) component. For Rayleigh fading channels, α is Rayleigh distributed. For AWGN channels, $\alpha(t) = 1$. Depending on whether the continuous real values of the received bits or only their signs are used for channel decoding, the decoder is said to perform *soft* or *hard* decision decoding, respectively.

For a discrete memoryless channel (DMC), the probability of receiving the vector $\boldsymbol{y}$ (N bits) given that the codeword $\boldsymbol{x}_m$ was transmitted is given by

$$p(\boldsymbol{y}|\boldsymbol{x_m}) = \prod_{j=1}^{N} p(y_j|x_{mj}) \qquad (0 \leq m \leq 2^K - 1). \tag{6.4}$$

A decoder maximizing Eq. 6.4 without regard to the *a priori* probabilities of the messages is called a *maximum likelihood* decoder. This decoding rule is applicable to all discrete memoryless channels, including both hard- and soft-decision channels.

6.3.1 Hard decision decoding

Additive white Gaussian noise and Rayleigh fading channels followed by *hard decision* thresholding act like a binary symmetric channel (BSC). For AWGN

and Rayleigh fading channels, the cross probability of the equivalent BSC is

$$p = Q\left(\sqrt{\alpha^2 \frac{2E_b}{N_0}}\right), \tag{6.5}$$

where E_b denotes the average energy per bit, N_0 is the average noise energy and $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz$ is the tail integral of the standard ($\mu = 0$, $\sigma = 1$) Gaussian density function. For Rayleigh channels, α is the Rayleigh distributed random variable, and for AWGN channels, $\alpha = 1$. If channel noise statistics are stationary over the transmission of a given codeword (N bits), the cross probability is a constant and the likelihood equation becomes

$$p(\boldsymbol{y}|\boldsymbol{x}_m) = p^{d_H}(1-p)^{N-d_H}, \tag{6.6}$$

where d_H is the Hamming distance between $\boldsymbol{y}$ and $\boldsymbol{x}_m$. Maximizing $p(\boldsymbol{y}|\boldsymbol{x}_m)$ is equivalent to minimizing the *Hamming* distance d_H between $\boldsymbol{y}$ and $\boldsymbol{x}_m$.

If an (N, K) linear code is only used for error *detection* over a BSC channel, the probability of correct decoding, undetected errors, and error detection are given by

$$P_{CD} = (1-p)^N, \tag{6.7}$$

$$P_{UE} = \sum_{w=d_{min}}^{N} A_w \, p^w \, (1-p)^{N-w} \quad \text{and} \tag{6.8}$$

$$P_{ED} = 1 - P_{CD} - P_{UE}, \tag{6.9}$$

where p is the transition probability of the BSC.

Example: Hard decision decoding of a (2,1) block code. Figure 6.8 shows a 2-dimensional example for decoding a (2,1) linear block with hard decision decoding. The distance spectrum of this simple CRC code is $A_w = [1, 0, 1]$. The valid codevectors are shown in dark circles. Assume the (+1,+1) codevector was transmitted.

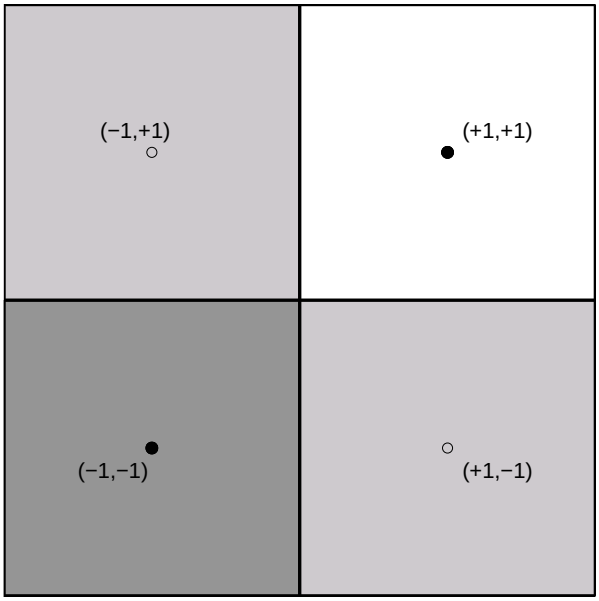


Figure 6.8: Illustration of hard decision decoding for the (2,1) block code. Color code is white for correct decoding (CD), light gray for error detection (ED), and dark gray for incorrect decoding or undetected error (UE).

If the soft received bits end up in the second or fourth quadrant, the resulting received codevector after bit thresholding is equally distant, in terms of Hamming distance, from two different valid codewords. No decision can be made and an erasure is declared (P_{ED}). If the received symbol is in the first or third quadrant, the codeword is correctly (P_{CD}) or incorrectly decoded (P_{UE}), respectively.

Mathematically, Eqs. 6.7 through 6.9 hold for the particular case of the (2,1) linear block code and provide the following relations:

$$P_{CD} = (1-p)^2 \tag{6.10}$$

$$P_{UE} = p^2 \tag{6.11}$$

$$P_{ED} = 2p\,(1-p). \tag{6.12}$$

6.3.2 Soft decision decoding

As mentioned in Section 1.5, soft decision decoding always outperforms hard decision decoding, both for AWGN and multi-path communication channels. This is a consequence of the data signal processing inequality which states that a loss of information is always associated with any processing of data [124]. This is easily verified when the processing operation for hard decision decoding is the non-invertible thresholding operation.

Consider then a *soft decision* memoryless channel where the channel input is ± 1 and the channel output is a real number with Gaussian statistics. Specifically, the stationary DMC is specified by

$$p(\boldsymbol{y}|\boldsymbol{x}_m) = \frac{1}{(\sqrt{\pi N_0})^N}\ e^{-\sum_{j=1}^{N} \frac{(y_j - x_{mj})^2}{N_0}}. \tag{6.13}$$

Maximizing $p(\boldsymbol{y}|\boldsymbol{x}_m)$ is equivalent to minimizing the squared *Euclidean* distance ${d_E}^2 = \sum_{j=1}^{N}(y_j - x_{mj})^2$ between $\boldsymbol{y}$ and $\boldsymbol{x}_m$.

The maximum likelihood (ML) decoder chooses its output to be the codeword for which the Euclidean distance between the received N-tuple $\boldsymbol{y}$ and the N-tuple codeword $\boldsymbol{x}_m$ is minimal. With soft decision decoding, as opposed to hard decision decoding which operates with Hamming distance, it is virtually impossible to be equidistant in Euclidean distances from two valid codewords. Consequently, $P_{ED} = 0$, allowing only for correct or erroneous decoding. There is no closed form formula for the values of P_{CD} and P_{UE}.

Example: Soft decision decoding of a (2,1) block code. Figure 6.9 illustrates soft decision decoding for the (2,1) CRC code. Based on minimum Euclidean distance decoding, the decision boundary for soft decision decoding is the median between two valid codewords. Both P_{CD} and P_{UE} increase since

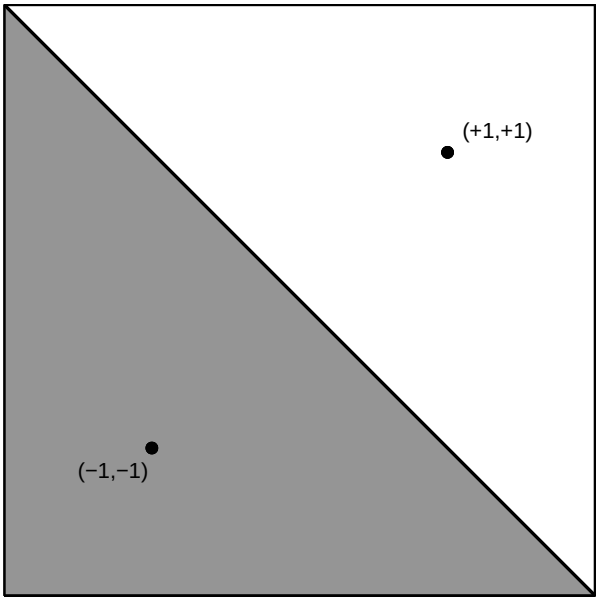


Figure 6.9: Illustration of soft decision decoding for the (2,1) block code. Color code is white for correct decoding (CD) and dark gray for incorrect decoding or undetected error (UE).

error detection is absent and redistributed equally between P_{CD} and P_{UE}. This ultimately decreases recognition performance.

For the particular case of the (2,1) linear block code, one can see that we have the following probabilities:

$$P_{CD} = Q\left(\sqrt{\alpha^2 \frac{4E_b}{N_0}}\right) \tag{6.14}$$

$$P_{UE} = Q\left(\sqrt{\alpha^2 \frac{-4E_b}{N_0}}\right) \tag{6.15}$$

$$P_{ED} = 0. \tag{6.16}$$

Soft decision decoding creates a paradox: while soft decision decoding typically improves transmission reliability in communication systems, it does not

help distributed speech recognition applications that are governed by different channel coding criteria, *i.e.* low P_{UE}, while it can tolerate large P_{ED}. In the next two sections, we investigate how to combine error detection with soft decision decoding.

6.3.3 Soft decision decoding using maximum *a posteriori* probabilities (β-soft)

In order to accept a decision provided by the soft decoder, one would like to evaluate the probability that the decoded codevector was indeed the one transmitted. Such an *a posteriori* probability is given by

$$p(\hat{\boldsymbol{x}} = \boldsymbol{x}_m|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x}_m) \cdot p(\boldsymbol{x}_m)}{P(\boldsymbol{y})} \tag{6.17}$$

$$= \frac{p(\boldsymbol{y}|\boldsymbol{x}_m) \cdot p(\boldsymbol{x}_m)}{\sum_{m'=0}^{2^K-1} p(\boldsymbol{y}|\boldsymbol{x}_{m'}) \cdot p(\boldsymbol{x}_{m'})} \tag{6.18}$$

which, if the assumption of equiprobable symbols is made, can be rewritten as

$$p(\hat{\boldsymbol{x}} = \boldsymbol{x}_m|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x}_m)}{\sum_{m'=0}^{2^K-1} p(\boldsymbol{y}|\boldsymbol{x}_{m'})} \tag{6.19}$$

$$= \frac{\prod_{j=1}^{N} e^{-\frac{(y_j - x_{m_j})^2}{N_0}}}{\sum_{m'=0}^{2^K-1} \prod_{j=1}^{N} e^{-\frac{(y_j - x_{m'_j})^2}{N_0}}} \tag{6.20}$$

$$= \frac{e^{-\frac{\sum_{j=1}^{N}(y_j - x_{m_j})^2}{N_0}}}{\sum_{m'=0}^{2^K-1} e^{-\frac{\sum_{j=1}^{N}(y_j - x_{m'_j})^2}{N_0}}} \tag{6.21}$$

$$= \frac{e^{-d_E^2(\boldsymbol{y},\boldsymbol{x}_m)/N_0}}{\sum_{m'=0}^{2^K-1} e^{-d_E^2(\boldsymbol{y},\boldsymbol{x}'_m)/N_0}}. \tag{6.22}$$

Eq. 6.22 is complex and requires the knowledge of N_0 that is difficult to probe. However, if we limit ourselves to the two closest valid codevectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ from the received codeword $\boldsymbol{y}$ and ignore the other distances, Eq. 6.22 becomes for the

closest vector $\boldsymbol{x}_1$

$$p(\hat{\boldsymbol{x}} = \boldsymbol{x}_1|\boldsymbol{y}) \approx \frac{e^{-d_E^2(\boldsymbol{y},\boldsymbol{x}_1)/N_0}}{e^{-d_E^2(\boldsymbol{y},\boldsymbol{x}_1)/N_0} + e^{-d_E^2(\boldsymbol{y},\boldsymbol{x}_2)/N_0}} \tag{6.23}$$

$$= \frac{1}{1 + e^{-(d_{E_2}^2 - d_{E_1}^2)/N_0}}, \tag{6.24}$$

where $d_{E_i} = d_E(\boldsymbol{y}, \boldsymbol{x_i})$ and $\boldsymbol{x_i}$ is the i^{th} closest valid codevector from the received symbol $\boldsymbol{y}$. For instance, if $d_{E_2} = d_{E_1}$, Eq. 6.24 gives $p(\hat{\boldsymbol{x}} = \boldsymbol{x}_1|\boldsymbol{y}) \approx \frac{1}{2}$.

While Eq. 6.24 still depends on the value of the channel noise N_0, it is clear that the reliability of the decoding operation can be evaluated from the values of d_{E_1} and d_{E_2}. Since the noise level may be unknown, we present a solution for estimating the confidence in the decoding of the feature based on the relative difference between the Euclidean distances (d_{E_1} and d_{E_2}) of the two closest valid codevectors ($\boldsymbol{x}_1$ and $\boldsymbol{x}_2$) from the received bit sequence $\boldsymbol{y}$,

$$\beta = \frac{d_{E_2} - d_{E_1}}{d_{E_1}}, \tag{6.25}$$

which is independent of the channel noise N_0.

If the received vector $\boldsymbol{y}$ lies exactly between two valid codewords ($\beta = 0$), the decoder's best decision is a guess between both codewords. On the other hand, if there is no noise in the channel, $d_{E_1} = 0$ and $\beta = \infty$. This shows that β can be used as a confidence measure of the decoding operation.

For future reference, this soft decision based error detection channel decoding scheme based on the value of β computed in Eq. 6.25 will be referred to as **$\boldsymbol{\beta}$-soft** decoding.

Example: $\boldsymbol{\beta}$-soft decision decoding of a (2,1) block code. Figure 6.10 illustrates an example of decoding the (2,1) linear block code using the *a posteriori* probability criterion. Error detection based on soft decision decoding can be

declared when β is smaller than a given threshold. As one can see, the decision region around the correct and incorrect codewords are circles. These circles represent the set of points in a 2-dimensional system whose ratio of distances with respect to two fixed points are a constant. The ratio $\rho = \frac{d_{E_2}}{d_{E_1}}$ can be immediately found from Eq. 6.25 as being equal to $\rho = 1 + \beta$.

If we assume the transmitted symbol was the point (+1, +1), the area where correct decoding is made, in accordance with Eq. 6.25, is the Appolonius[1] circle whose center is located at $(\frac{\rho^2+1}{\rho^2-1}, \frac{\rho^2+1}{\rho^2-1})$ and whose radius is $\sqrt{2}\frac{2\rho}{\rho^2-1}$. Note that if $\beta = 0$, which corresponds to $d_{E_1} = d_{E_2}$, we have a circle whose center is located at the coordinates (∞, ∞) and whose radius is infinitely large. This is geometrically equivalent to the median between both codewords, as in soft decision decoding. However, if β increases, the center of the circle approaches the transmitted symbol and its radius decreases.

6.3.4 Soft decision decoding using log likelihood ratios (λ-soft)

Since maximum likelihood is the optimal decision rule, it might be desirable to perform error detection based on the ratio of the likelihoods of the two most probable codevectors. Using Bayes rule and assuming that all codewords are equiprobable, the ratio of the likelihoods of the two most probable vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ (which are also the two closest codevectors from the received vector $\boldsymbol{y}$ at Euclidean distances d_{E_1} and d_{E_1} from $\boldsymbol{y}$) is given by

$$\frac{P(\boldsymbol{y}|\boldsymbol{x} = \boldsymbol{x}_1)}{P(\boldsymbol{y}|\boldsymbol{x} = \boldsymbol{x}_2)} = \frac{e^{-\frac{d_{E_1}^2}{2\sigma^2}}}{e^{-\frac{d_{E_2}^2}{2\sigma^2}}} \tag{6.26}$$

$$= e^{\frac{1}{2}\frac{(d_{E_2}^2 - d_{E_1}^2)}{\sigma^2}}. \tag{6.27}$$

[1]In geometry, the Appolonius circle is defined as the set of points whose ratio of Euclidean distances with respect to two fixed points is a constant. If the ratio is 1, the circle becomes the median line between both points.

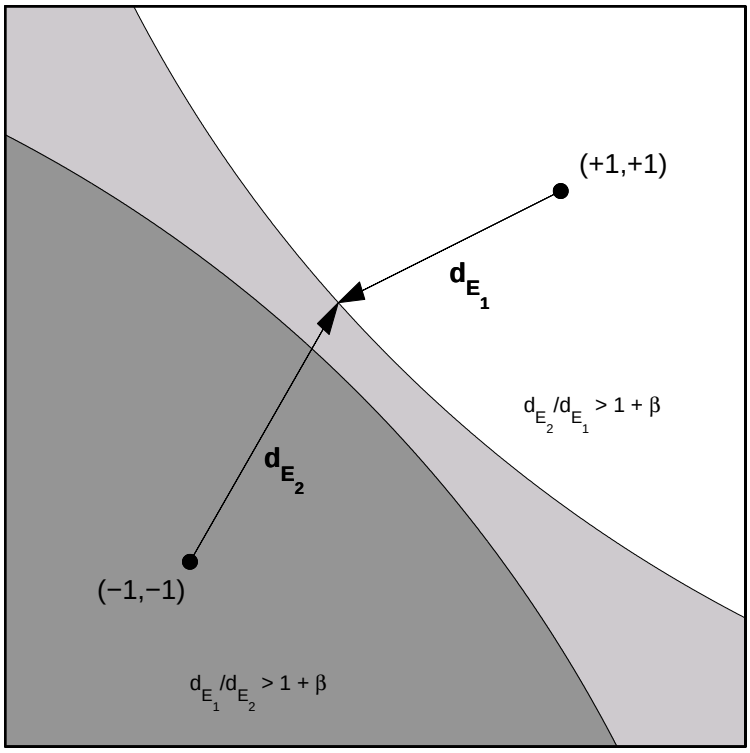


Figure 6.10: Illustration of *a posteriori* β-soft decision decoding for the (2,1) block code. Color code is white for correct decoding (CD), light gray for error detection (ED), and dark gray for incorrect decoding or undetected error (UE).

If one determines the projection of the received codevector $\boldsymbol{y}$ onto the line segment joining $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ as defining the distance d_1 and d_2 on the inter-segment, geometry tells us that $(d_{E_2}^2 - d_{E_1}^2) = (d_2^2 - d_1^2)$ and Eq. 6.30 can be rewritten as

$$\frac{P(\boldsymbol{y}|\boldsymbol{x} = \boldsymbol{x}_1)}{P(\boldsymbol{y}|\boldsymbol{x} = \boldsymbol{x}_2)} = e^{\frac{1}{2}\frac{(d_2^2 - d_1^2)}{\sigma^2}} \tag{6.28}$$

$$= e^{\frac{1}{2}\frac{(d_2 - d_1)(d_2 + d_1)}{\sigma^2}} \tag{6.29}$$

$$= e^{\frac{1}{2}(\frac{D}{\sigma})^2(\frac{d_2 - d_1}{D})} \tag{6.30}$$

where the variance σ^2 is equal to $N_0/2$, D is the Euclidean distance between the two codevectors closest to the received codeword $\boldsymbol{y}$, and d_1 and d_2 are the distances from the projection of the received codevector to the line joining the two closest codevectors. The important factor in Eq. 6.30 is

$$\lambda = \frac{d_2 - d_1}{D} \ , \tag{6.31}$$

which is independent of the channel noise N_0.

If $\lambda = 0$, both codevectors are equally probable and the decision of the maximum likelihood decoder should be rejected. If $\lambda = 1$ ($d_1 = 0$ and $d_2 = D$), correct decision is almost guaranteed since the block codes used are chosen according to channel conditions so that the minimum Euclidean distance between any two codevectors is at least several times as large as the expected noise ($D^2/N_0 \gg 1$). Note that $\lambda > 1$ would occur if the received vector $\boldsymbol{y}$ does not lie somewhere in the space between the two closest codevectors but away from $\boldsymbol{x}_1$ in the opposite direction from $\boldsymbol{x}_2$, in which case $d_1 < 0$.

For general cases, simulations are necessary to evaluate the performance of the new decoding scheme. This soft decision based error detection channel decoding scheme based on the value of λ computed in Eq. 6.31 will be referred to as **$\boldsymbol{\lambda}$-soft** decoding.

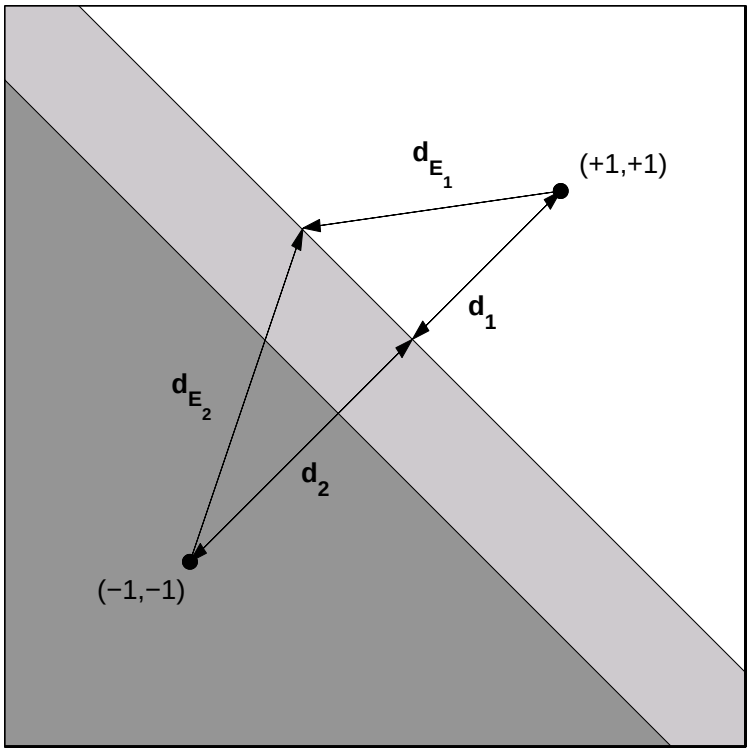


Figure 6.11: Illustration of *a posteriori* λ-soft decision decoding for the (2,1) block code. Color code is white for correct decoding (CD), light gray for error detection (ED), and dark gray for incorrect decoding or undetected error (UE).

Example: λ-soft decision decoding of a (2,1) block code. Figure 6.11 illustrates the decoding operation when the likelihood ratio of the two most likely codewords is used as criterion. Error detection based on soft decision decoding can be declared when λ is smaller than a given threshold. Note that the larger λ is, the larger the area of error detection. The gray area which represents the area of erasure declaration grows linearly with λ. Classic soft decision decoding is a particular case with $\lambda = 0$.

With such a soft decision scheme adapted for error detection based on the ratio of the likelihood, P_{UE} and P_{CD} can be expressed for the specific case of the

(2,1) linear block code as follows:

$$P_{UE} = Q\left(\sqrt{(1+\frac{\lambda}{2})\ \alpha^2\frac{4E_b}{N_0}}\right) \tag{6.32}$$

$$P_{CD} = Q\left(-\sqrt{(1-\frac{\lambda}{2})\ \alpha^2\frac{4E_b}{N_0}}\right) \tag{6.33}$$

$$P_{ED} = 1 - P_{UE} - P_{CD}. \tag{6.34}$$

6.3.5 Comparison between β- and λ-soft decision decoding

In order to compare both soft decision based schemes allowing error detection, one would like to pursue the example of the (2,1) linear block code and analyze the merits of both β- and λ-soft decision decoding procedures presented. Figure 6.12 shows the merits of both channel decoding algorithms using the simple (2,1) linear block code for different values of β (for *a posteriori* based soft decision decoding) and λ (for maximum-likelihood based soft decision decoding) when operating on an AWGN channel at -2 dB SNR. Note that maximum likelihood based soft decision decoding always outperforms maximum *a posteriori* based soft decision decoding at least as far as P_{UE} is concerned, which is the most important probability for remote speech recognition applications.

This observation can intuitively be verified if one observes in Figure 6.11 that the boundaries between error detection and correct or incorrect decoding is a set of points of equal likelihood given the channel conditions. This means that the subspace that is set aside for error detection only contains vectors whose likelihoods are smaller than a certain threshold. In other words, λ-soft decision decoding separates the reliable feature from the unreliable ones in a manner that is consistent with their likelihoods. This is not the case with β-soft decision decoding.

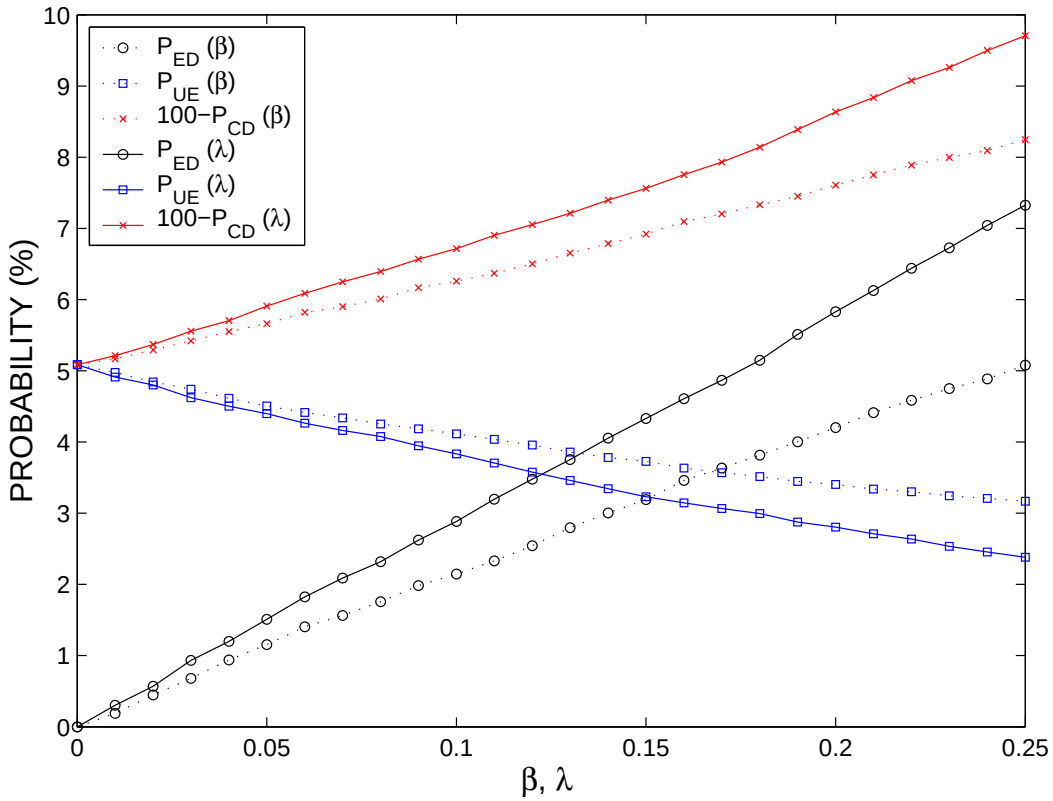


Figure 6.12: Comparison of the probability of correct detection (P_{CD}), error detection (P_{ED}) and undetected error (P_{UE}) depending on the channel decoding system used (β-soft or λ-soft) for the (2,1) linear block codes over an independent Rayleigh fading channel at -2 dB SNR.

This result is an intrinsic property of λ-soft decoding and can be verified with different types of linear block codes or channel conditions. For this reason, λ-soft decision decoding is our choice for soft decision channel decoding in the design of complete remote speech recognition systems.

6.4 Performance of the different channel decoding schemes

As an example, consider the (10,7) SED block code of Table 6.2 over an independent Rayleigh fading channel at 5 dB SNR. Hard decoding yields $P_{UE} = 0.3\%$, $P_{ED} = 30.2\%$ and $P_{CD} = 69.5\%$. These numbers are insufficient to provide good recognition results. Soft decision decoding, on the other hand, does not perform much better since the probability of undetected errors is too large ($P_{UE} = 2.6\%$).

Figure 6.13 illustrates the performance of the λ-soft decision decoding schemes for the same code over the same channel for different values of λ. Note first that λ-soft decision decoding with $\lambda = 0$ corresponds to classic soft decision decoding. With increasing λ, however, one can rapidly reduce P_{UE} to the desired values, while still keeping P_{CD} large enough and usually above that of hard decision decoding. For instance, with $\lambda = 0.16$, we have $P_{UE} = 0.5\%$, $P_{ED} = 7.7\%$ and $P_{CD} = 91.8\%$, which results in good recognition accuracy. Note that when P_{UE} decreases, P_{CD} decreases as well, which indicates that a tradeoff must be found.

Figure 6.14 illustrates the different probabilities (correct detection, erasure detection and undetected errors) for a family of block codes with increasing redundancy over a wide range of Rayleigh fading channel conditions. The figure motivates the design of different channel coding schemes for different channel conditions. Once the signal-to-noise ratio of the independent Rayleigh fading

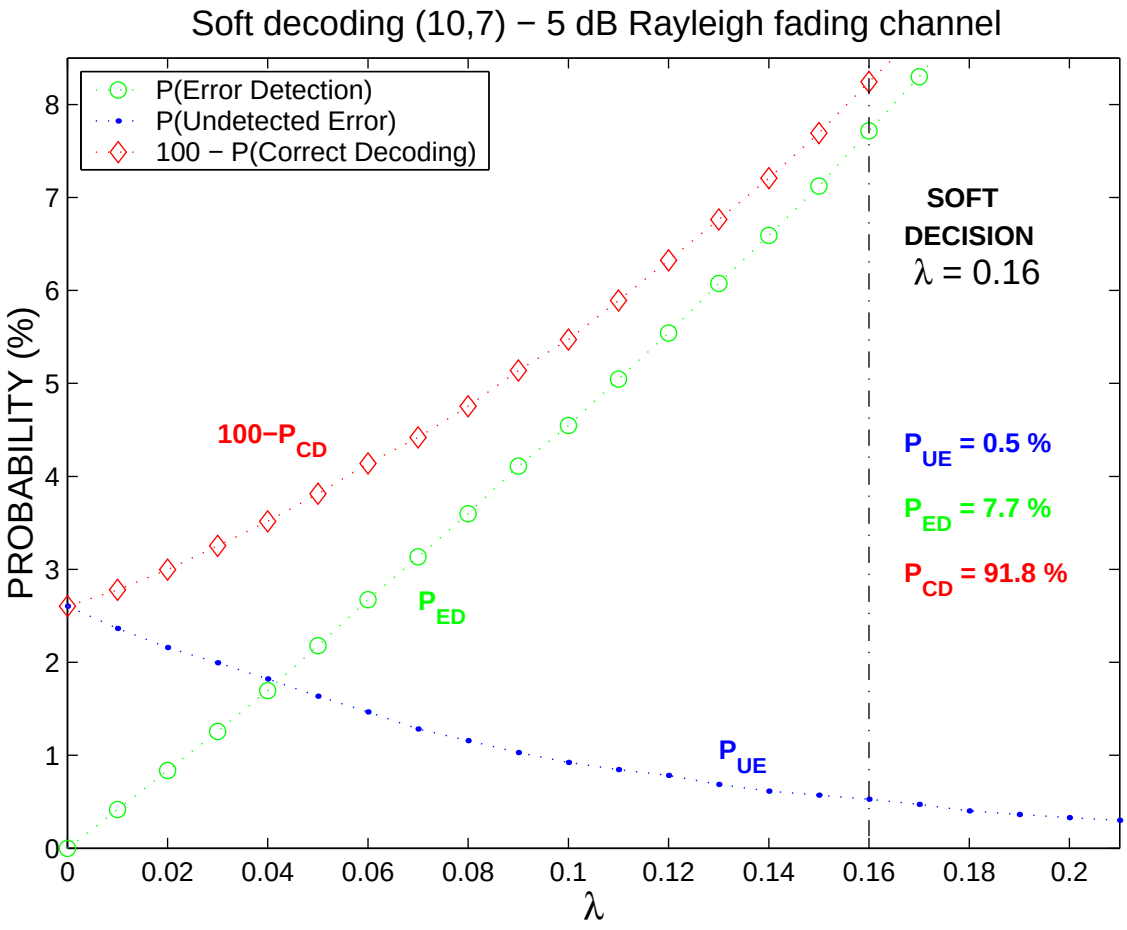


Figure 6.13: Illustration of the probability of correct detection (P_{CD}), error detection (P_{ED}) and undetected error (P_{UE}) as a function of the parameter λ when using λ-soft decision decoding of the (10,7) DED linear block code over an independent Rayleigh fading channel at 5 dB SNR.

channel no longer guarantees a low enough probability of undetected errors, even with soft decision decoding, it is time to switch to a higher rate channel encoder, which is often done at the expense of source coding resolution.

The probabilities (correct decoding, undetected error and error detection) for the block codes designed for different independent Rayleigh fading channel SNRs are listed in Table 6.3. The codes presented can be used for transmission of PLP cepstral coefficients quantized with 5 to 7 bits per split (see Section 5.3.3). The value $\lambda = 0.16$ is experimentally found appropriate to keep the number of undetected errors small while the probability of correct decoding remains high.

Note that soft decoding is made at the cost of additional complexity of computing Euclidean distances for all 2^K codewords. However, channel decoding is done at the server, where the complexity of the recognizer prevails.

Similar results are obtained for the additive white Gaussian noise channel and are presented in Table 6.4. The codes presented can be used for transmission of MFCC coefficients quantized with 7 to 10 bits per split (see Section 5.4.1).

In both cases, hard decoding typically keeps P_{UE} small enough, but at the cost of too many frames being erased, and there are not enough frames correctly decoded to attain high recognition accuracy. Classic soft decision decoding, on the other hand, suffers from the fact that it cannot detect errors, which results in a large proportion of erroneously decoded frames. λ-soft decision decoding, however, always meets the channel coding requirements for remote speech recognition, $P_{UE} << 1\%$ and $P_{CD} > 90\%$.

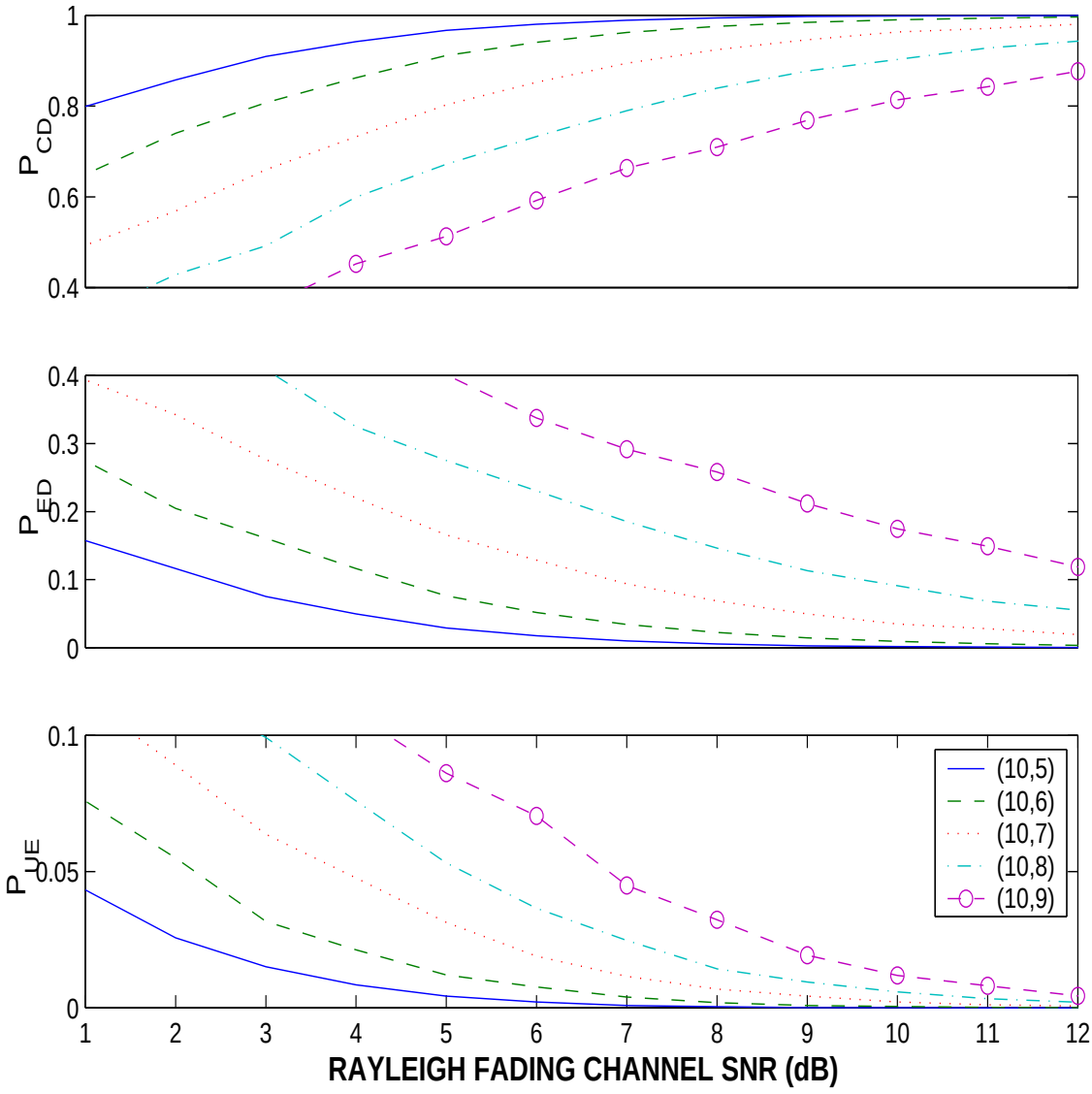


Figure 6.14: Illustration of the probability of correct detection (P_{CD}), error detection (P_{ED}) and undetected error (P_{UE}) as a function of the independent Rayleigh fading channel SNR for a variety of linear block codes.

Code	**SNR**	P_{CD} (%)			P_{ED} (%)		P_{UE} (%)		
(N,K)	(dB)	Hard	Soft	λ-soft	Hard	λ-soft	Hard	Soft	λ-soft
(8,7)	10	90.5	98.6	93.0	9.1	6.9	0.4	1.4	0.1
(8,7)	9	88.7	97.9	91.3	10.7	8.5	0.6	2.1	0.2
(8,6)	8	86.2	99.0	95.3	13.6	4.6	0.2	1.0	0.1
(8,6)	7	82.9	98.4	94.0	16.7	5.7	0.3	1.6	0.3
(10,7)	6	74.7	98.5	94.0	25.0	5.7	0.3	1.5	0.2
(10,7)	5	69.5	97.4	91.8	30.2	7.7	0.3	2.6	0.5
(10,6)	4	64.3	98.7	95.3	35.7	4.4	0.1	1.3	0.2
(10,6)	3	58.1	97.9	93.3	41.8	6.3	0.1	2.1	0.4
(10,5)	2	51.6	98.9	96.1	48.4	3.7	0.0	1.1	0.2
(10,5)	1	45.0	97.8	93.6	55.0	6.0	0.0	2.2	0.5

Table 6.3: Probability of correct detection (P_{CD}), error detection (P_{ED}) and undetected error (P_{UE}) using hard, soft and λ-soft ($\lambda = 0.16$) decision decoding for the proposed linear block codes over different independent Rayleigh fading channel SNRs. $P_{ED} = 0$ for soft decision decoding.

6.5 Summary

In this chapter, we analyzed and simulated the effect of channel errors and channel erasures on recognition accuracy. As opposed to speech coding, it was shown that remote speech recognition systems are significantly more sensitive to channel errors than channel erasures. In the first case, the ML path might diverge for the error free case, which may considerably modify the resulting likelihood computation. In the second case, redundancy in the speech signal indicates that a recognizer can cope with a large percentage of channel erasures. Note that this analysis does not take into account the impact that channel errors may also have on the predictive source coding scheme, in which case channel errors can

Code	**SNR**	**BER**	P_{CD} (%)			P_{ED} (%)		P_{UE} (%)		
(N,K)	(dB)	(%)	Hard	Soft	λ-soft	Hard	λ-soft	Hard	Soft	λ-soft
(10,10)	5.97	0.25	97.4	97.4	91.2	0.0	8.2	2.6	2.6	0.6
(10,9)	4.33	1.00	90.4	98.3	92.1	9.2	7.8	0.4	1.7	0.1
(10,8)	3.24	2.00	81.4	97.8	91.2	18.2	8.5	0.4	2.2	0.3
(11,8)	2.48	3.00	71.8	97.3	90.8	27.8	8.7	0.4	2.7	0.5
(12,8)	1.31	5.00	54.5	96.1	86.8	45.3	12.7	0.2	3.9	0.5
(12,7)	0.37	7.00	42.2	95.8	86.7	57.7	12.6	0.1	4.2	0.7

Table 6.4: Probability of correct detection (P_{CD}), error detection (P_{ED}) and undetected error (P_{UE}) using hard, soft and λ-soft ($\lambda = 0.16$) decision decoding for the proposed linear block codes over different AWGN channel SNRs. $P_{ED} = 0$ for soft decision decoding.

propagate through signal prediction.

Based on this observation, we presented channel coding techniques utilizing linear block codes which would maximize the error detection capability of the code and be adapted to the low source coding rate for ASR feature quantization.

Finally, we presented several channel decoding techniques, including two techniques which allowed for soft-decision based error detection. In particular, error detection using the likelihood ratios outperformed error detection using the *a posteriori* probabilities. In both cases, the channel decoder, which can probe the channel using the soft outputs of the channel, returned a soft channel decoding reliability metric which can be used in a weighted Viterbi recognizer, as will be seen in the next chapter.

CHAPTER 7

Remote recognition system design and performance

This chapter presents different techniques that can be applied at the server (receiver) in order to improve recognition accuracy over a wide range of channel conditions. Section 7.1 presents several techniques aimed at alleviating the effect of channel transmission. The results of the different solutions proposed in Section 7.1 are compared in Section 7.2. Section 7.3 presents a method for incorporating the effect of channel transmission in the training of the HMM models. Finally, Section 7.4 presents recognition results for complete DSR systems, including source coding, channel coding and decoding, as well as frame erasure concealment.

7.1 Alleviating the effect of channel transmission and erasures

This section presents different techniques specifically designed for coping with channel transmission and erasures, regardless of whether the erasures are the result of a detected channel error or an actual channel erasure (*i.e.* packet loss). The different techniques proposed include: 1) dropping the frames that are de-

clared missing or in error; 2) keeping the doubtful frames, but weighting down their importance in the Viterbi likelihood computation; 3) applying frame erasure concealment; and 4) combining weighted Viterbi recognition with frame erasure concealment. Recognition results using these different techniques are compared in Section 7.2.

7.1.1 Frame dropping

The first technique reduces the effect of channel transmission on recognition accuracy by detecting channel errors, and consequently removes the "suspicious" feature vectors from the sequence of observations. The motivation behind this technique is that, as suggested in Section 6.1, channel errors rapidly degrade recognition accuracy, while recognizers can cope with missing segments in the sequence of observations given the redundancy of the speech signal.

The question regarding detection of errors occurring during transmission was studied in Chapter 6. For now, we assume that such a detection scheme exists, and study ways of alleviating the effect of missing frames.

The drawback of removing altogether the unreliable frames from the stream of feature vectors is that the timing information associated with it is lost. This can significantly impact recognition accuracy.

7.1.2 Weighted Viterbi recognition (WVR)

With remote recognition, reliability of the decoded features is a function of channel characteristics. When channel characteristics degrade, one can no longer guarantee the confidence in the decoded feature. If the Viterbi algorithm (VA) operates without taking into account the decreased feature reliability, this can

have a dramatic effect on recognition accuracy since errors in feature decoding can propagate through the trellis (see Section 6.1.1).

In this section, we present a solution for modifying the recursive step (Eq. 6.1) of the VA to take into account the effect of channel transmission. Ideally, one would like to weigh the probability of observing the decoded feature given the HMM state model $b_j(\boldsymbol{o}_t)$ with the probability of decoding the feature vector $\boldsymbol{o}_t$. The time-varying weighting coefficient γ_t can be inserted into Eq. 6.1 by raising the probability $b_j(\boldsymbol{o}_t)$ to the power γ_t to obtain

$$\phi_{j,t} = \max_i \left[\phi_{i,t-1}\ a_{ij}\right]\ \left[b_j(\boldsymbol{o}_t)\right]^{\gamma_t}. \tag{7.1}$$

If one is certain about the received feature (no channel noise), $\gamma_t = 1$ and Eq. 7.1 is equivalent to Eq. 6.1. If, on the other hand, the decoded feature is unreliable, $\gamma_t = 0$ and the probability of observing the feature given the HMM state model $b_j(\boldsymbol{o}_t)$ is discarded in the VA recursive step.

Note that, under the hypothesis of a diagonal covariance matrix $\boldsymbol{\Sigma}$, the overall probability $b_j(\boldsymbol{o}_t)$ can be computed as the product of the probabilities of observing each individual feature. If the features are quantized and transmitted separately, the channel-matched recursive formula (Eq. 7.1) is improved to include individual weighting factors $\gamma_{k,t}$ for each of the N_F features:

$$\phi_{j,t} = \max_i \left[\phi_{i,t-1}\ a_{ij}\right]\ \prod_{k=1}^{N_F} \left[b_j(o_{k,t})\right]^{\gamma_{k,t}}. \tag{7.2}$$

7.1.2.1 Binary weighting

With binary weighting, the weighting coefficients γ_t can either be zero (if the frame is lost or declared in erasure) or one (if the frame is received). The advantage of this technique over frame dropping, where state metrics are not updated ($\phi_{j,t} = \phi_{j,t-1}$), is that the timing information of the observation sequence is con-

served. State metrics are continuously updated, even when $\gamma_t = 0$, by virtue of the state transition probability matrix using

$$\phi_{j,t} = \max_i \ [\phi_{i,t-1} \ a_{ij}]. \tag{7.3}$$

7.1.2.2 Continuous weighting

Note that the system can be refined if a time-varying continuous estimate $\gamma_t \in [0, 1]$ of the feature vector reliability is made available to the recognition engine.

We introduced the WVR technique in [58] to match the recognizer with the confidence in the decoded feature after channel transmission. Such weighting is a function of the channel decoder. The weighting coefficients would be binary if hard decision decoding is used, and continuous between 0 and 1 if soft decision decoding is used. We will return to channel decoding based continuous weighting for WVR in Section 7.4, which includes channel coding and decoding in a complete DSR system. For the moment, we use WVR to take into account frame erasures (binary weighting) only.

There exists another way to characterize a continuous weighting coefficient for WVR through the use of frame erasure concealment, where missing frames are concealed with an estimate. Quality of the substitutions can be evaluated and used as continuous weighting coefficients for the WVR scheme.

7.1.3 Frame erasure concealment

The problem when a large number of frames is not received at the decoder is that the synchronization of the Viterbi recognizer may be disturbed, even if state metrics are continuously updated using only the transition matrix. Hence, subsequent received features might be analyzed using an inappropriate state.

This problem becomes more significant when erasures occur in bursts, almost forcing the best path in the trellis to remain in the same state for a long period of time.

Feature concealment methods not only preserve the timing information, but also attempt to recreate the missing feature vector by replacing it with an estimate. *Repetition*-based schemes replace missing frames with copies of previously-received frames, while *interpolation*-based schemes use some form of pattern matching and interpolation from the neighboring frames to derive a replacement frame [60, 61, 62]. Both techniques are justified by the high correlation between consecutive frames of speech signals. Interpolation techniques require reception of the next valid feature vector, which adds significant delay when errors occur in bursts. For this reason, only repetition-based techniques will be analyzed.

7.1.4 Erasure concealment combined with WVR

Performance of repetition or interpolation techniques degrades rapidly as the number of consecutive lost frames increases, since the quality of the replacement features decreases with regards to its similarity with the missing features. For instance, when packet losses approach or exceed the length of a phoneme (10-100 ms or 1-10 frames), the speech signal may already have evolved to another sound, which no longer justifies repetition of the last correctly received feature vector.

This section presents an extension to the repetition-based concealment technique, whereby the confidence in the frame erasure concealment is fed into the Viterbi recognizer for improved recognition performance. Indeed, it is beneficial to decrease the weighting factor $\gamma_{k,t}$ when the number of consecutively repeated

CHANNEL STATE	GOOD	BAD	GOOD
Frames status	√ √ √	− − − − √ − − − −	√ √ √ √ √ − √ √
Temporal Features	√ √ √	⇝ ⇝ ⇝ ⇝ √ ⇝ ⇝ ⇝ ⇝	√ √ √ √ √ ⇝ √ √
Derivatives	√ √ ↶	× × × × × × × × ×	↷ √ √ √ ↶ √ ↷ √
Accelerations	√ √ ↶	× × × × × × × × ×	↷ √ √ √ √ √ √ √

Table 7.1: Example of computation of temporal and dynamic features in the presence of frame erasures.

frames increases. For the weighting coefficients, we propose

$$\gamma_{k,t} = \sqrt{\rho_k(t - t_c)}, \tag{7.4}$$

where ρ_k is the time auto-correlation of the k^{th} feature and t_c is the time instant of the last correctly received frame. Note that if there is no erasure, $t = t_c$ and $\gamma_{k,t} = 1$.

7.1.4.1 Computation and weighting of dynamic features

Erasures also propagate through the computation of feature derivatives and accelerations. For this reason, only immediate left and right neighboring frames are used for the computation of dynamic features. While this might result in a slight loss of performance in erasure-free conditions, it provides robustness against erasures.

The case of PLP_D_A features is analyzed where recognition is performed including the temporal features (PLP) and the dynamic features (derivative 'D' and acceleration 'A'). The dynamic features are computed at the receiver as follows. First, the receiver determines what the status of the channel is. If two consecutive frames are lost/received, then the receiver determines that the channel is bad/good.

Gilbert Channel State	*Good*	*Bad*
Temporal features	$\gamma_{k,t} = \sqrt{\rho_k(t-tc)}$	
Dynamic features	$\gamma_{k,t} = 1$	$\gamma_{k,t} = 0$

Table 7.2: Determination of the frame erasure concealment based weighting coefficients for WVR.

Table 7.1 explains in detail how to compute and weigh the features depending on whether the frame was received (shown as $\surd$) or lost (shown as $-$). Table 7.1 includes transitions from a good state to a bad state in the Gilbert channel model and vice versa, as well as erasures within a good state and correct transmission within a bad state. For the *temporal* features, missing frames are replaced with a copy of the temporal features in the previous frame (shown as $\rightsquigarrow$). Weighting coefficients follow the square root of the time-correlation of each feature (Eq. 7.4). For the *dynamic* features, if only one of the two features necessary for the computation of a derivative is missing, a one-sided derivative is computed ($\curvearrowleft$ or $\curvearrowright$). If both are missing, the dynamic feature is not computed and is discarded in the Viterbi search (shown as $\times$) by assigning zero weighting.

In the bad channel state, temporal features are repeated and the weighting coefficients of the dynamic features are set to zero. If the channel state is good, the dynamic features are computed and the weighting coefficients of the dynamic features are set to one. A one-sided derivative is used if a neighboring frame is lost on either side while still in a good channel state. Table 7.2 summarizes the WVR weighting coefficients as a function of the channel status.

This option is chosen over repeating the entire previous frame (temporal *and* dynamic features) for the following reason. While time-correlation between successive temporal features allows for repetition, time-correlation of the dynamic

features is significantly smaller than for the temporal features. Hence, repetition of dynamic features does not necessarily lead to a good estimate of the missing features and should be avoided.

7.2 Recognition results for the different techniques alleviating the effect of channel erasures

This section compares recognition results for the different techniques aimed at alleviating the effect of channel erasures. The recognition experiment is based on continuous digit recognition using unquantized PLP_6 features and the Aurora-2 database [104] and configuration (word models, 16 states and 6 mixtures/states).

Table 7.3(a) illustrates recognition accuracy for the different frame erasure concealment techniques applied to the independent erasure channel. Baseline recognition accuracy for erasure-free channels is 98.52%. Several observations can be made.

1. After about 10-20% of independent frame erasures, recognition accuracy degrades rapidly with an increased percentage of erasures.

2. Transmission of the binary frame erasure reliability measurement to the weighted Viterbi recognizer preserves synchronization of the VA and significantly reduces the word error rate.

3. Repetition-based frame erasure concealment, which in addition to preserving the timing also provides an approximation for the missing feature, typically outperforms binary WVR for the channel with independent erasures.

4. Addition of the continuous weighting coefficients $\gamma_{k,t}$ representing the quality of the feature concealment technique (Eq. 7.4) in the Viterbi search

Independent Erasures	0%	10%	20%	30%	40%	50%	60%
Frame dropping	98.52	97.19	93.51	85.49	71.23	56.33	38.76
Binary WVR	98.52	98.31	98.11	97.19	96.87	94.31	93.19
Repetition	98.52	98.47	98.31	98.19	97.67	96.35	94.31
Repetition + Cont. WVR	98.52	98.52	98.47	98.39	98.11	97.61	96.01

(a) Independent erasure channels.

Gilbert Channels	(2.5,20)	(2.5,15)	(5,20)	(2.5,10)	(1.25,5)	(5,15)	(10,20)	(5,10)
Frames dropping	90.68	87.04	85.79	81.67	80.07	79.33	74.70	69.85
Binary WVR	97.35	96.27	96.20	94.53	93.69	95.06	94.85	92.82
Repetition	97.41	96.41	96.77	94.42	93.27	94.35	93.83	92.11
Repetition + Cont. WVR	98.07	97.55	97.84	97.37	97.03	97.15	96.87	96.09

(b) Bursty (Gilbert) erasure channels.

Table 7.3: Recognition accuracy with the Aurora-2 database and PLP_D_A features using two types of channel erasures: (a) independent and (b) bursty. Different techniques for the effect of channel erasures are compared: frame dropping; frame dropping with binary WVR ($\gamma_t = 0$ if frame is dropped); frame erasure concealment (repetition); and repetition with continuous WVR ($\gamma_{k,t} = \sqrt{\rho_k(t - t_c)}$).

further improves recognition performance.

These results are confirmed in Table 7.3(b) for the bursty Gilbert channels of Table 6.1 for which we can make additional observations:

1. Binary WVR may outperform repetition-based erasure concealment when the average burst lengths are large.

2. Again, frame erasure concealment combined with WVR provides the best recognition results. For instance, for the Gilbert channel with $(P_{GB}, P_{BG}) = (1.25, 5)$, recognition accuracy improves from 93.27% to 97.03%, a 71% relative word error rate (WER) reduction compared to the baseline recognition performance of 98.52%.

3. Despite average overall probability of frame erasures between 18% and 33% and average length of erasure bursts between 5 and 20 frames (see Table 6.1), recognition accuracy is kept within 2% of the baseline erasure-free performance.

7.3 Note on channel multi-conditional training

A typical solution for speech recognition applications to gain robustness against acoustic noise is to train the acoustic models on a training data set that includes both clean and noisy signals. If the training is done with a variety of background acoustic noises, the resulting HMM models are hybrid enough to provide high recognition accuracy both for clean and noisy speech. This technique is called *multi-conditional* training.

One would like to use the same technique to combat channel noise. Unfortunately, multi-conditional training can be used for acoustic noise because the

feature representing the speech signal impaired by noise will be correlated to the original feature, to a certain degree that may vary with noise levels. For instance, evaluations on the Aurora-2 database with MFCCs showed that recognition accuracy for clean speech signals based on clean and multi-conditionally trained HMM models was 98.4% and 98.2%, respectively. This indicates that the multi-conditionally trained HMM models should be comparable to the clean HMM models.

This would not be the case for channel noise, where after channel and source decoding, the decoded features may look very different from the original feature and the resulting HMM models may fail to represent scattered distributions of features. This will notably reduce recognition accuracies in clean channel situations and fall short from representing all possible channel noises.

The solution to the problem of channel multi-conditional training lies in finding a way to cope at the *statistical level* (not at the feature level) with the randomness of feature distributions in the presence of channel noise. The answer to the problem is to prevent erroneous features from destroying the acoustic models by filtering them out after channel decoding. This is also the goal successfully pursued by the weighted Viterbi recognition algorithm. One could imagine that a similar technique could be used for channel multi-conditional training.

The novelty of the WVR algorithm is to modify the probability of observing a feature vector from $P(\boldsymbol{o}_t)$ to $P(\boldsymbol{o}_t)^{\gamma_t}$ to incorporate the effect of the channel transmission. The effect of channel noise is mapped into a single parameter, γ_t, which represents the channel decoding reliability.

For the purpose of channel multi-conditional training, imagine further that the new value $P(\boldsymbol{o}_t)^{\gamma_t}$ can represent, with proper normalization, a new probability taking into account the effect of channel transmission. Acoustic models estimated

using these probabilities should be more robust to channel transmission if the same weighting is applied during both training and testing.

The appropriate normalization of the probability must be equal to the integral from $-\infty$ to $+\infty$ of the new probability distribution. This guarantees that the new probability distribution sums up to one. The normalization factor can then be computed as follows:

$$\begin{aligned} K_t &= \int_{-\infty}^{\infty} P(\boldsymbol{o}|\gamma_t)\ \mathrm{dx} && (7.5)\\ &= \int_{-\infty}^{\infty} P(\boldsymbol{o_t})^{\gamma_t}\ \mathrm{dx} && (7.6)\\ &= \int_{-\infty}^{\infty} \left(\frac{1}{\sqrt{(2\pi)^{N_F}|\boldsymbol{\Sigma}|}}\ e^{-\frac{1}{2}(\boldsymbol{o_t}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{o_t}-\boldsymbol{\mu})} \right)^{\gamma_t} \mathrm{dx} && (7.7)\\ &= \int_{-\infty}^{\infty} \left(\frac{1}{\sqrt{(2\pi)^{N_F}|\boldsymbol{\Sigma}|}} \right)^{\gamma_t} e^{-\frac{1}{2}(\boldsymbol{o_t}-\boldsymbol{\mu})'\left(\frac{\boldsymbol{\Sigma}}{\gamma_t}\right)^{-1}(\boldsymbol{o_t}-\boldsymbol{\mu})}\ \mathrm{dx}. && (7.8) \end{aligned}$$

This value can be simplified if one can re-write Eq. 7.8 using another multi-variate Gaussian distribution with covariance matrix $\frac{\boldsymbol{\Sigma}}{\gamma_t}$ for which the determinant is $|\frac{\boldsymbol{\Sigma}}{\gamma_t}| = \frac{1}{\gamma_t^{N_F}}|\boldsymbol{\Sigma}|$ to obtain

$$\begin{aligned} K_t &= \frac{1}{\sqrt{(2\pi)^{N_F}|\boldsymbol{\Sigma}|}^{\gamma_t - 1}} \frac{1}{\sqrt{\gamma_t}^{N_F}} \int_{-\infty}^{\infty} \frac{1}{\sqrt{(2\pi)^{N_F}|\frac{\boldsymbol{\Sigma}}{\gamma_t}|}}\ e^{-\frac{1}{2}(\boldsymbol{o_t}-\boldsymbol{\mu})'\left(\frac{\boldsymbol{\Sigma}}{\gamma_t}\right)^{-1}(\boldsymbol{o_t}-\boldsymbol{\mu})}\ \mathrm{dx} \\ &= \frac{1}{\sqrt{\gamma_t^{N_F}((2\pi)^{N_F}|\boldsymbol{\Sigma}|)^{\gamma_t - 1}}}. && (7.9) \end{aligned}$$

Once the normalization constant, which is time-varying and a function of γ_t, is computed, we obtain the following expression of the weighted probability:

$$\begin{aligned} P(\boldsymbol{o}_t|\gamma_t) &= \frac{P(\boldsymbol{o}_t)^{\gamma_t}}{K_t} && (7.10)\\ &= P(\boldsymbol{o}_t)^{\gamma_t} \cdot \sqrt{\gamma_t^{N_F}((2\pi)^{N_F}|\boldsymbol{\Sigma}|)^{\gamma_t - 1}}. && (7.11) \end{aligned}$$

Note that K_t is time-varying, independent of $\boldsymbol{\mu}$, but is dependent of $\boldsymbol{\Sigma}$. Note

also that if $\gamma_t = 1$, which corresponds to the original unweighted case, one has $K_t = 1$, as expected.

The EM algorithm could be applied to this new set of probabilities, which would take into account the channel conditions to derive HMM models that are more channel robust by matching HMM training with channel conditions.

For instance, if *single* mixture Gaussian models with diagonal covariance were used (or if the two Gaussians were sufficiently separated), one can analytically derive the resulting Gaussian mean ($\hat{\boldsymbol{\mu}}$) and variance ($\hat{\boldsymbol{\Sigma}}$) parameters by maximizing the log probability of observing the training set of features given the parameters. If we refer to the likelihood by L, one obtains

$$\begin{aligned} L &= \prod_{t=1}^{T} P(\boldsymbol{o}_t|\gamma_t) && (7.12) \\ &= \prod_{t=1}^{T} \frac{P(\boldsymbol{o}_t)^{\gamma_t}}{K_t} && (7.13) \\ &= \prod_{t=1}^{T} \frac{\left[\frac{1}{(2\pi)^{N_F/2}} \frac{1}{|\Sigma|^{1/2}}\ e^{-\frac{1}{2}(\boldsymbol{o}_t-\boldsymbol{\mu})\Sigma^{-1}(\boldsymbol{o}_t-\boldsymbol{\mu})'}\right]^{\gamma_t}}{K_t} && (7.14) \end{aligned}$$

and the following for the logarithm of the likelihood

$$\log(L) = -\frac{1}{2}\sum_{t=1}^{T}\left[N_F \log(2\pi) + \log(|\Sigma|) + (\boldsymbol{o}_t - \boldsymbol{\mu})\Sigma^{-1}(\boldsymbol{o}_t - \boldsymbol{\mu})'\right] \cdot \gamma_t - \log(K_t). \tag{7.15}$$

The parameter $\hat{\boldsymbol{\mu}}$ that maximizes $\log(L)$ is found by taking the derivative of Eq. 7.15 with respect to $\boldsymbol{\mu}$ and equaling it to zero. In other words, since $\frac{\partial K_t}{\partial \mu} = 0$, we have

$$\frac{\partial \log(L)}{\partial \boldsymbol{\mu}} = -\frac{1}{2}\sum_{t=1}^{T}(\boldsymbol{o}_t - \boldsymbol{\mu})\Sigma^{-1}\gamma_t \tag{7.16}$$

which is equal to zero if and only if

$$\sum_{t=1}^{T}\gamma_t \boldsymbol{o}_t = \sum_{t=1}^{T}\gamma_t \boldsymbol{\mu} \tag{7.17}$$

or equivalently if

$$\hat{\boldsymbol{\mu}} = \frac{\sum_{t=1}^{T} \gamma_t \boldsymbol{o}_t}{\sum_{t=1}^{T} \gamma_t}. \tag{7.18}$$

Note that if $\forall t\ \gamma_t = 1$, computation of the mean becomes $\hat{\boldsymbol{\mu}} = \frac{\sum_{t=1}^{T} \boldsymbol{o}_t}{T}$, which is equivalent to the formula for the unweighted case.

For the estimation of the diagonal elements ($\hat{\sigma_j}$) of the covariance matrix $\boldsymbol{\Sigma}$, assuming diagonal covariance matrix, Eq. 7.15 can be rewritten as

$$\begin{aligned} \log(L) &= \sum_{t=1}^{T} \left[-\frac{N_F}{2} \log(2\pi) - \sum_{j=1}^{N_F} \left(\log(\sigma_j) + \frac{(o_{t,j} - \mu_j)^2}{2\sigma_j^2} \right) \right] \cdot \gamma_t \\ &\quad + \frac{N_F}{2} \log \gamma_t + \frac{N_F}{2} (\gamma_t - 1) \log(2\pi) + (\gamma_t - 1) \sum_{j=1}^{N_F} \log(\sigma_j). \end{aligned} \tag{7.19}$$

Taking the derivative of Eq. 7.19 with respect to each variance (σ_j), one can see that the maximum log likelihood is obtained when the derivative

$$\frac{\partial \log(L)}{\partial \sigma_j} = \sum_{t=1}^{T} \left[-\frac{\gamma_t}{\sigma_j} + \frac{(o_{t,j} - \mu_j)^2}{\sigma_j^3} \gamma_t + \frac{(\gamma_t - 1)}{\sigma_j} \right] \tag{7.20}$$

equals zero. This is the case when

$$\hat{\sigma_j}^2 = \frac{\sum_{t=1}^{T} \gamma_t (o_{t,j} - \mu_j)^2}{T}. \tag{7.21}$$

Note again that if $\forall t\ \gamma_t = 1$, the variance becomes

$$\hat{\sigma_j}^2 = \frac{\sum_{t=1}^{T} (o_{t,j} - \mu_j)^2}{T}, \tag{7.22}$$

which is equivalent to the regular expression for the computation of the variance. In the other extreme case where $\forall t\ \gamma_t = 0$ (all the observations are unreliable and the probabilities are equal), then $\hat{\sigma_j}^2 = 0$.

Intuitively, as expected, Eqs. 7.18 and 7.21 indicate that the elements received with a high degree of reliability should be given a larger weight in the estimation

of the means and variances. However, note that no mathematical expression exists for the case of Gaussian mixtures, and an iterative training algorithm, such as the EM algorithm, must be used.

Note that the distribution for $P(\boldsymbol{o}_t|\gamma_t)$ can be computed by directly dividing $P(\boldsymbol{o}_t)^{\gamma_t}$ by K_t to yield

$$\begin{aligned}
P(\boldsymbol{o}_t|\gamma_t) &= \left(\frac{1}{\sqrt{(2\pi)^{N_F}|\boldsymbol{\Sigma}|}} \; e^{-\frac{1}{2}(\boldsymbol{o_t}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{o_t}-\boldsymbol{\mu})} \right)^{\gamma_t} \cdot \sqrt{\gamma_t^{N_F}((2\pi)^{N_F}|\boldsymbol{\Sigma}|)^{\gamma_t - 1}} \\
&= \sqrt{\frac{\gamma_t^{N_F}(2\pi)^{N_F(\gamma_t-1)}|\boldsymbol{\Sigma}|^{\gamma_t-1}}{(2\pi)^{N_F\gamma_t}|\boldsymbol{\Sigma}|^{\gamma_t}}} \; e^{-\frac{1}{2}(\boldsymbol{o_t}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{o_t}-\boldsymbol{\mu})\cdot\gamma_t} \qquad (7.23) \\
&= \sqrt{\frac{\gamma_t^{N_F}}{(2\pi)^{N_F}|\boldsymbol{\Sigma}|}} \; e^{-\frac{1}{2}(\boldsymbol{o_t}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{o_t}-\boldsymbol{\mu})\cdot\gamma_t} \qquad (7.24) \\
&= \frac{1}{\sqrt{(2\pi)^{N_F}|\frac{\boldsymbol{\Sigma}}{\boldsymbol{\gamma_t}}|}} \; e^{-\frac{1}{2}(\boldsymbol{o_t}-\boldsymbol{\mu})'\frac{\boldsymbol{\Sigma}}{\boldsymbol{\gamma_t}}^{-1}(\boldsymbol{o_t}-\boldsymbol{\mu})}, \qquad (7.25)
\end{aligned}$$

which is another Gaussian probability distribution whose covariance matrix is the original covariance matrix divided by γ_t.

The motivation for doing this is that HMM models trained on the features corrupted by channel transmission could provide robustness against channel noise in the same way that HMM models trained on features computed from noisy speech signals provide robustness against acoustic noise. However, this method is not pursued here since it would slightly degrade the results of the recognition engine operating on uncorrupted data.

7.4 Performance of complete remote recognition systems

In the remainder of this chapter, the concepts presented previously (source coding, channel decoding and frame erasure concealment with weighted Viterbi recognition) are applied to quantized ASR features in order to evaluate a complete DSR

system including source and channel coding and decoding.

7.4.1 Comparison between hard and soft decision decoding

In this section, we compare recognition results when using hard and soft decision decoding, both with and without WVR.

For the purpose of performing channel decoding based WVR, we first recall that soft-decision based error detection was achieved using two types of channel decoding reliability measurement: one based on the ratio of the *a posteriori* probabilities (β_t) and one based on the ratio of the log likelihoods (λ_t).

The estimate for the γ_t coefficient can be obtained by defining a mapping function between the decoding measure β_t introduced in Section 6.3.3 (Eq. 6.25) ($0 \leq \beta_t \leq \infty$) or the measure λ_t introduced in Section 6.3.4 (Eq. 6.31) ($0 \leq \lambda_t \leq 1$) and the Viterbi weighting coefficient γ_t ($0 \leq \gamma_t \leq 1$).

For the β_t value, we propose the following sigmoid function,

$$\gamma_t = 1/(1 + e^{-21.8(\beta_t - 0.3)}) \tag{7.26}$$

to map the relative difference in Euclidean distances β_t into confidence estimate γ_t. This function, shown in Figure 7.1, gives a confidence measure $\gamma_t < 0.1$ when $\beta_t < 0.2$ and $\gamma_t > 0.9$ when $\beta_t > 0.4$.

Another solution for defining a weighting coefficient γ_t for the weighted Viterbi algorithm could be more naturally derived from the λ_t value where $\lambda_t \in [0, 1]$. While the mapping function could be $\gamma_t = \lambda_t$, such linear function would not decrease the weight of the unreliable feature enough (like in the sigmoid function Eq. 7.26). We propose the following square function to map the interval $[0, 1]$ for λ_t to the interval $[0, 1]$ for γ_t: $\gamma_t = \lambda_t^2$. The quadratic exponent is empirically chosen after it was proven to provide the necessary statistical rejection of the

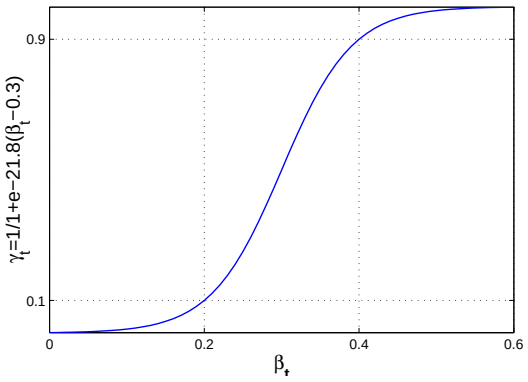


Figure 7.1: Sigmoid function mapping relative Euclidean distance difference (β_t) to confidence measure (γ_t).

uncertain frames. Given the superior behavior of soft decision decoding based on the λ_t measurement, only the second channel decoding (λ-soft) and quadratic mapping will be analyzed.

Note that if hard decision decoding was used, only binary weighting coefficients could be used for the WVR ($\gamma_t = 0$ for erasure and $\gamma_t = 1$ for reception). For soft decision decoding, one can choose to apply the same binary weighting with $\gamma_t = 0$ if $\lambda_t < \tau$ and $\gamma_t = 1$ if $\lambda_t \geq \tau$, or to apply the continuous weighting $\gamma_t = \lambda^2$.

Figure 7.2 compares recognition accuracy using λ-soft decision decoding with continuous WVR, λ-soft decision decoding with binary WVR, and the widely used hard decision decoding with binary WVR when transmitting PLP_6 features using the (10,6) linear block code over a wide range of independent Rayleigh fading channels. Note that performing joint channel decoding and recognition with continuous weighting always outperforms the other two strategies. For a given recognition accuracy level, the gain of the λ-soft decision decoding with continuous WVR is roughly 1 and 3 dB over λ-soft and hard decoding with

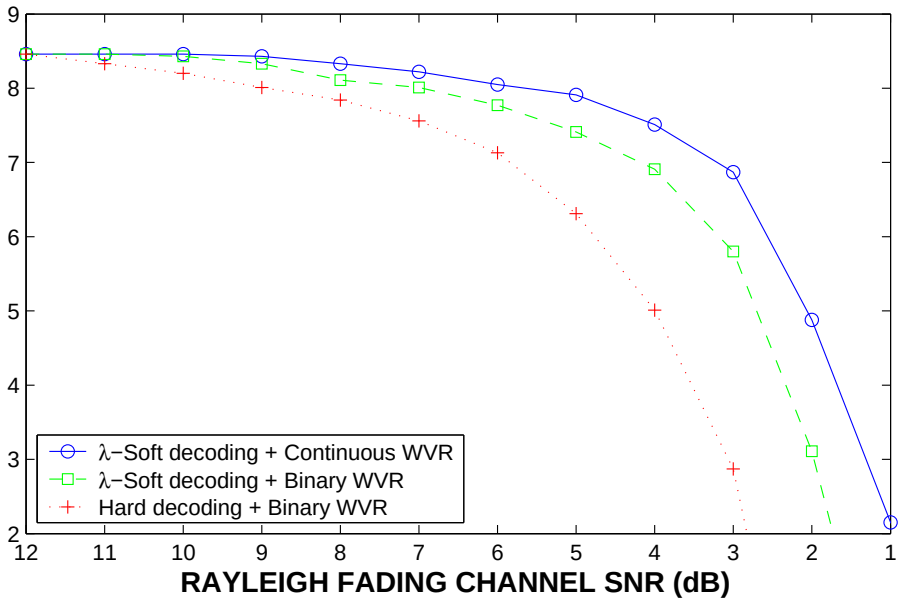


Figure 7.2: Recognition accuracy using the P-LSFs of PLP_6 quantized with 6 bits per frame and the (10,6) linear block code over an independent Rayleigh fading channel.

binary WVR, respectively.

Figure 7.3 illustrates recognition accuracy after choosing for each SNR the block code that yields the best results. Observe again the notably superior performance of the joint soft decision decoding-Viterbi recognition scheme. Even for bit rates as low as 500 bps, λ-soft decision decoding combined with continuous WVR allow for respectable recognition accuracies over a wide range of independent Rayleigh fading channel SNRs.

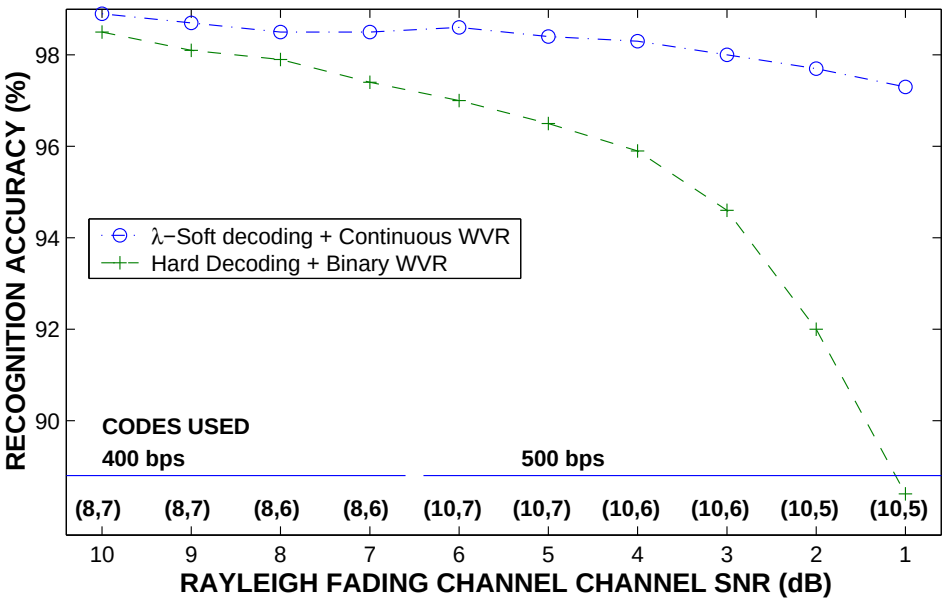


Figure 7.3: Recognition accuracy after transmission of the P-LSFs of PLP_6 over an independent Rayleigh fading channel.

7.4.2 Comparison between WVR with and without frame erasure concealment

Simulations carried out in the previous section indicated that λ-soft decision decoding always outperformed hard decision decoding. It was also shown that performing continuous WVR with $\gamma_t = \lambda_t^2$ as a weighting coefficient instead of frame dropping with binary weighting further improved recognition accuracy throughout the spectrum of channel conditions.

In this section, we compare the merits of such configuration (λ-soft decoding with continuous WVR) with another configuration which conceals the erased or unreliable frames, both with and without continuous WVR, based this time on the concealment quality weighting coefficient.

Table 7.4 presents recognition accuracy after transmission over a wide range

of independent Rayleigh fading channels whose equivalent bit error rate ranges from 2% to 12%. Source coding is applied on the LSFs of the PLP_6 system, using anywhere between 5 and 8 bits per frame. Depending on the channel conditions, different channel encoders are used. Overall bit rate, including source and channel coding, is limited to less than 500 bps. Note that the equivalent BER of the Rayleigh channel, computed as

$$P_e = \frac{1}{2}\left(1 - \sqrt{\frac{z}{1+z}}\right) \tag{7.27}$$

where z is the channel SNR, is also reported.

Two scenarios are considered. In the first one, all the features are transmitted to the recognizer, even the unreliable ones, and the weighting coefficients ($\gamma_t = \lambda_t^2$) will lower the importance of the inaccurate ones. In the second, the unreliable features (those for which $\lambda_t < 0.16$) are dropped and concealed with a substitution feature vector. The WVR weighting coefficient is based on the quality of the concealment operation ($\gamma_{k,t} = \sqrt{\rho_k(t - t_c)}$).

Table 7.4 indicates that no one strategy always outperforms the other in a statistically significant manner. However, it is expected that for heavily correlated fading channels which can cause bursts of errors, frame erasure concealment could provide improved recognition.

7.5 Performance of remote recognition systems using quantized MFCCs

Parts of the experiments presented above for PLP are repeated in this section for MFCC features, illustrating the generality of the source coding, channel coding and channel decoding scheme presented in the previous chapters.

Block Code (N,K)	SNR (dB)	BER (%)	RECOGNITION (%)	
			λ-soft $\gamma_t = \lambda_t^2$ Cont. WVR	λ-soft $\gamma_{k,t} = \sqrt{\rho_k(t - t_c)}$ Cont. WVR
(8,7)	10	2.33	98.7	98.7
(8,7)	9	2.88	98.5	98.6
(8,6)	8	3.55	98.3	98.2
(8,6)	7	4.35	98.3	98.4
(10,7)	6	5.30	98.4	98.5
(10,7)	5	6.42	98.2	98.3
(10,6)	4	7.71	98.1	98.1
(10,6)	3	9.19	97.8	97.8
(10,5)	2	10.85	97.5	97.6
(10,5)	1	12.67	97.1	97.4

Table 7.4: Comparison between performance of channel based continuous WVR ($\gamma_t = \lambda_t^2$) and erasure concealment based continuous WVR ($\gamma_{k,t} = \sqrt{\rho_k(t - t_c)}$).

MFCCs are quantized using the techniques presented in Section 5.4 with 7 to 9 bits per splits. After channel protection, the number of bits after forward error correction is 10 or 12 bits per split, for a total of 1.0 or 1.2 kbps, depending on channel conditions.

The source and channel coding pairs used over an independent Rayleigh fading channel at different SNRs are presented in Table 7.5. Note again that λ-soft decision decoding almost always meets the channel coding requirements for remote speech recognition, $P_{UE} < 0.5\%$ and $P_{CD} > 90\%$. Hard decision would not be able to meet the requirement for P_{CD}, and soft decision would satisfy the requisite for P_{UE}.

Figure 7.4 illustrates recognition accuracy after choosing for each SNR the

Code	**SNR**	P_{CD} (%)			P_{ED} (%)		P_{UE} (%)		
(N,K)	(dB)	Hard	Soft	λ-soft	Hard	λ-soft	Hard	Soft	λ-soft
(10,9)	11	90.6	98.7	91.3	9.0	8.7	0.4	1.3	0.0
(10,9)	10	88.4	97.9	89.3	11.1	10.6	0.6	2.1	0.1
(10,8)	9	86.0	98.8	94.0	13.8	5.8	0.3	1.2	0.1
(10,8)	8	82.6	98.3	92.4	17.0	7.4	0.4	1.7	0.2
(12,9)	7	75.4	98.3	93.0	24.2	6.7	0.3	1.7	0.3
(12,8)	6	70.2	99.3	96.2	29.8	3.8	0.0	0.7	0.0
(12,8)	5	65.0	98.7	94.3	35.0	5.6	0.1	1.3	0.2
(12,8)	4	58.8	97.6	91.6	41.1	8.1	0.1	2.4	0.3
(12,7)	3	52.3	98.6	94.3	47.7	5.5	0.0	1.4	0.2
(12,7)	2	45.1	97.3	91.3	54.9	8.3	0.1	2.7	0.4
(12,7)	1	38.2	95.3	87.3	61.8	11.7	0.1	4.7	1.0

Table 7.5: Probability of correct detection (P_{CD}), error detection (P_{ED}) and undetected error (P_{UE}) using hard, soft and λ-soft ($\lambda = 0.16$) decision decoding for the proposed linear block codes over different independent Rayleigh fading channel SNRs. $P_{ED} = 0$ for soft decision decoding.

block code that yields the best results. The superior performance of the joint soft decision decoding-Viterbi recognition scheme is confirmed for MFCC features. Recognition accuracies remain acceptable over a wide range of independent Rayleigh fading channel SNRs and using overall bit rates less than 1.2 kbps. This improves by a factor of 4 the scheme proposed in the Aurora-2 standard [104].

7.6 Summary

In this chapter, we first developed frame erasure concealment techniques which could significantly improve recognition accuracy over a wide range of indepen-

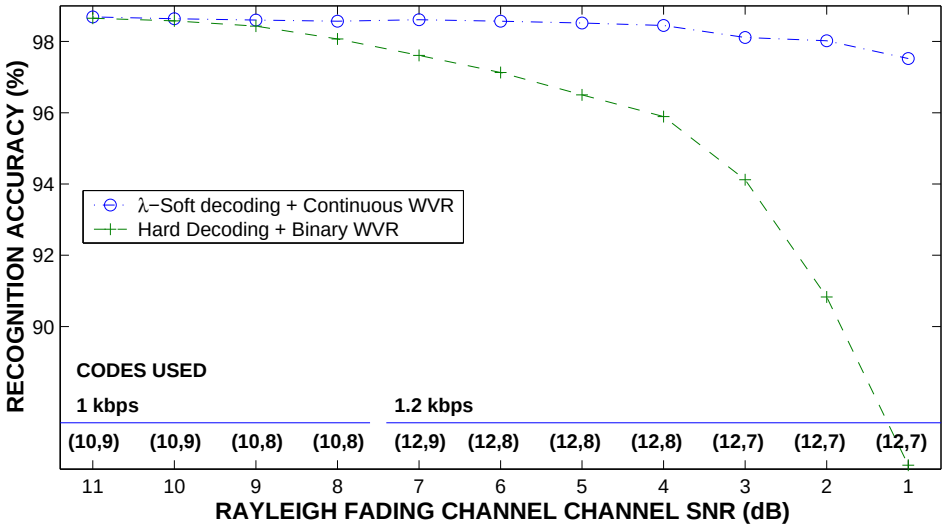


Figure 7.4: Recognition accuracy after transmission of the 13 MFCCs over an independent Rayleigh fading channel.

dent and bursty erasure channels. The techniques presented included frame dropping, weighted Viterbi recognition, frame erasure concealment and channel multi-conditional training.

Second, we analyzed the performance of complete DSR systems, which include ASR feature quantization as well as channel coding and decoding. Over independent Rayleigh fading channels, the λ-soft channel decoding scheme combined with continuous WVR was shown to outperform both hard and λ-soft decoding with frame dropping and binary WVR. By using a variety of source and channel coding rate combinations, it was shown that remote speech recognition could be obtained over a wide range of channel conditions and with a high recognition rate with less than 500 bps for PLP features and 1.2 kbps for MFCC features. Finally, the respective merits of channel based WVR and erasure concealment based WVR are compared.

CHAPTER 8

Summary, discussion and future work

We have presented source and channel coding solutions for two types of speech applications: speech transmission and remote speech recognition. The first application, speech transmission, in which the speech signal is played back at the receiver, focuses on speech quality as judged by a human listener. The second application, remote speech recognition, focuses on the accuracy of the automatic speech recognizer at the server.

While both applications are fundamentally different, the source and channel coding techniques used in both applications involve similar concepts: 1) variable bit rate source coding, 2) analysis of the sensitivity of the information bits to channel errors, 3) determination of channel coding requirements, 4) variable redundancy rate channel coding, and 5) dynamic bit allocation between source and channel coding so that with a fixed overall bit rate, graceful performance degradation over a wide range of channel conditions is achieved.

The realization of these concepts for both applications are reviewed in the next two sections. The last two sections recapitulate the main contributions made and present future research directions.

8.1 Adaptive multi-rate speech transmission

In the first application, considered in Part I, we have shown how to combine embedded and variable bit rate speech source coding with rate-compatible punctured channel coding to build adaptive multi-rate (AMR) transmission systems.

Chapter 2 presented two types of speech coders whose properties (variable bit rate and embedibility) are useful for implementing embedded adaptive multi-rate speech transmission systems. The first coder is a perceptually-based subband coder which incorporates knowledge of the human auditory system to produce a bandwidth efficient bitstream with a very wide range of perceptual sensitivities to channel errors. The second coder is the ITU embedded G.727 ADPCM standard. The bit error sensitivities of both coders are evaluated to help determine for each bit position the maximum tolerable BER that keeps channel distortions inaudible and the amount of channel protection required.

Chapter 3 introduced the new rate-compatible punctured trellis codes (RCPT) and compared their performance with rate-compatible punctured convolutional codes (RCPC) and rate-compatible punctured convolutional codes with bit-interleaved coded modulation (RCPC-BICM). RCPT codes are designed in order to maximize the Euclidean distance between trellis error events, whereas RCPC codes maximize Hamming distances. For 4-PSK constellations, RCPC codes also maximize Euclidean distance, but for larger constellations, they are sub-optimal. Larger constellation sizes are important for larger throughput. For instance, it would be interesting to propose RCPT codes for the upcoming EDGE (enhanced data rate GSM evolution) channels, which will be using 8-PSK constellations.

The advantage of RCPT over RCPC-BICM comes from the combination of trellis coding and modulation for an improved Euclidean distance profile. Also,

RCPT decoding typically requires a smaller punctured traceback depth than RCPC and RCPC-BICM; hence it requires smaller frame sizes and buffering delays. By dropping symbols instead of bits, RCPT codes provide variable *baud* rates. This can be useful in situations where it is advantageous to lower the symbol throughput or the transmitter power consumption. The robustness and flexibility of the progressive symbol puncturing scheme make the proposed architecture promising for communication channels where deep fade or strong interference can be modeled as symbol puncturing.

In Chapter 4, we present a technique for finding the optimal puncturing schemes for different channel conditions and source bit rates, and we design AMR systems for the perceptually-based embedded subband encoder and the embedded ADPCM standard. Based on bit error sensitivity analysis, unequal error channel protection requirements are determined and provided by RCPT, RCPC and RCPC-BICM codes. Performance of AMR systems operating at 10 kbaud/s using an 8-PSK constellation for RCPT and RCPC-BICM or on a 4-PSK constellation for RCPC are compared.

The main advantage of our AMR scheme is that different operating modes can be selected at each time instant according to channel quality. One can switch from a source/channel coding rate combination to another depending on channel conditions. This results in bandwidth efficient speech communication with consistent quality over a wide range of channel conditions.

The intrinsic *embedded* structure of both source and channel encoders offers multiple advantages. First, the entire AMR system can be implemented using a single codec. Only the number of allocated bits and the puncturing table need to be updated at the transmitter when switching between operating modes. At the receiver, adjustment to rate changes is simple. Branch metrics corresponding

to the punctured symbols in the Viterbi decoder are set to zero according to the puncturing table.

Second, one can drop bits or symbols anywhere in the transmission link, without having to re-encode the signal. This offers flexibility for traffic management.

Third, embedded source coders usually produce bits with a wide range of predictable sensitivities against channel errors and are therefore well suited for unequal error protection. We have shown, for instance, that perceptually-based dynamic bit allocation can isolate several bits in the bitstream that are almost insensitive to channel errors and can be left unprotected. In addition, embedded coding structures allow for multi-resolution coding, which is highly desirable for delay sensitive communication systems, as there is no need to wait for the reception of the entire bitstream before recovering speech of reasonable quality.

Systems using AMR source and channel coding are likely to be integrated in future communication systems. We have provided some examples that demonstrate the potential for RCPT AMR systems to provide graceful speech degradation over a wide range of channel SNRs.

8.2 Remote speech recognition

For the second application, considered in Part II, we presented a framework for developing source coding, channel coding and decoding as well as erasure concealment techniques for DSR applications.

As a case study, source coding, channel coding, and speech recognition techniques are combined to provide high recognition accuracy over a large range of channel conditions for two types of features, PLP and MFCC.

In Chapter 5, the perceptual line spectral frequencies representing the PLP

spectrum are quantized using weighted vector quantization operating at low bit rates (300 bps). The weighting coefficients are obtained after analytical and experimental study of the sensitivity of recognition accuracies to PLP quantization. Similar techniques are used for MFCC quantization at less than 1.2 kbps.

It is shown in Chapter 6 that speech recognition, as opposed to speech coding, is more sensitive to channel errors than channel erasures, and appropriate channel coding design criteria are determined.

Efficient channel coding techniques for error detection based on linear block codes are presented in Chapter 6, and a new technique that performs error detection with soft decision decoding is described. The new channel decoder, which introduces additional complexity only at the server, is proven to outperform the widely-used hard decision decoding for error detection.

Once an error is detected, the corresponding frame is erased, and frame erasure concealment techniques which alleviate the effects of channel transmission are discussed. We introduced in Chapter 7 the weighted Viterbi recognizer (WVR), whereby the recognizer is modified to include a time-varying weighting factor depending on the assessed quality of each feature after transmission over time-varying channels.

We demonstrate in Chapter 7 that high recognition accuracy over a wide range of channel conditions is possible with low overall bit rate when using the appropriate source and channel coder, as well as the adapted frame erasure concealment and weighted Viterbi recognition techniques.

Note that the source and channel coding techniques presented are not restricted to the transmission of PLP or MFCC and can be extended to other recognition features.

Together, channel estimation, erasure concealment and weighted Viterbi speech

recognition are shown to improve robustness of the DSR system against channel noise, extending the range of channel conditions over which wireless or internet-based speech recognition can be sustained.

To our knowledge, the effect of channel transmission on remote recognition systems based on quantized ASR features is a topic not yet extensively covered in the literature. Hence, our analysis and the proposed techniques present a significant improvement toward gaining robustness against channel noise.

8.3 Contributions

The main contributions of this dissertation are in the design of source and channel coding strategies for speech transmission and remote speech recognition.

Part I of the dissertation, dedicated to improving speech quality after transmission over a wide range of channel conditions, makes contributions in the following areas.

In the area of *source coding* (Chapter 2), we present the bit error sensitivity of two embedded coders and determine adequate levels of forward error coding to keep channel distortions inaudible.

In the area of *rate-compatible channel coding* (Chapter 3), we introduce the Rate-Compatible Punctured Trellis codes (**RCPT**) whereby unequal error protection is obtained by puncturing symbols in a trellis. RCPT codes are designed to maximize residual Euclidean distance after puncturing, and are well suited for constellations where Euclidean and Hamming distances are not equivalent.

In the area of *AMR system design* (Chapter 4), we introduce a technique for determining puncturing architecture in accordance with channel conditions.

Part II of the dissertation, which is dedicated to improving remote speech

recognition accuracy, makes contributions in several areas.

In the area of *source coding* (Chapter 5), we propose efficient quantization techniques for PLP or MFCC features for remote speech recognition based on quantized features.

In the area of *channel coding* (Chapter 6), it is first predicted and experimentally verified that speech recognition, as opposed to speech coding, is more sensitive to channel errors than channel erasures. Two types of channels are analyzed: independent and bursty channels. Efficient channel coding techniques for error detection based on linear block codes and the above requirements are determined.

In the area of *channel decoding* (Chapter 6), the merits of soft and hard decision decoding are discussed, and new techniques for performing error detection with soft decision decoding are presented. The soft decision channel decoder, which introduces additional complexity only at the server, is proven to outperform the widely-used hard decision decoding scheme.

In the area of *speech recognition* (Chapter 7), the recognition engine is modified to include a time-varying weighting factor depending on the quality of each decoded feature after transmission over time-varying channels. Two different techniques are proposed to assess the quality of the decoder feature: 1) based on the soft values of the received bits, the probability of correctly decoding the feature given the channel condition is computed, and constitutes a basis for estimating the quality of the decoder feature; 2) frame erasure concealment is used to approximate the missing features, and the quality of the concealment operation is estimated. Subsequently, the quality of the features are taken into account by a weighted Viterbi recognizer (**WVR**).

8.4 Looking forward

The first part of this research has demonstrated the usefulness of adaptive transmission with embedded source and channel coding, and provided new methodologies for improved implementation of embedded coding techniques in speech communication. The work provides opportunities for several areas of research.

The current method for designing RCPT codes for Rayleigh fading channels is based on the periodic effective code length (PECL) metric, which is essentially a measure of the diversity provided by the code after periodic puncturing. It would be interesting to extend the code design methodology of RCPT codes using metrics reflecting the true diversity of the code.

Another main area of future work would consist of designing source and channel coders that are truly joint. Indeed, while the existing literature deals extensively with source optimized channel coding or decoding (*e.g.* [125]) and channel optimized source coding (*e.g.* [126]), the approach of joint source and channel coding remains an open area of research. Applied to speech, the very existence of channel coding can be rethought. Since the original redundancy of the speech signal can serve as an efficient method to deal with noisy transmission, perhaps one can design a speech source codec that is in essence robust against reasonable transmission error.

The second part of this research made numerous contributions in the design of remote speech recognition applications operating at low bit rates over error-prone channels. Possible extensions of this work can be found in the following areas of research.

One can foresee that in the future, mobile devices will free themselves from interaction with a server for performing speech recognition. When the increas-

ing memory size and processor speed become sufficient to store the acoustic and language model as well as to perform the tasks of front-end processing and recognition, it is reasonable to believe that a mobile device could perform speech recognition locally. This opens a new area for research, which aims at reducing the memory and computational requirements for achieving local speech recognition. For instance, it would be interesting to see how recognition performance would degrade with constraints on the available memory or computational power.

A second possible extension is to apply the proposed methodologies to different recognition tasks, such as large vocabulary continuous speech recognition (LVCSR), as well as to analyze the effect of model size (word, phone, tri-phone) on source and channel coding design for remote speech recognition.

Finally, after robustness against quantization and channel noise has been achieved through the use of appropriate source and channel coding methods, respectively, another area of research is to analyze the robustness of the proposed features and quantization methods to acoustic noise. Given the mobility of the users, they will be exposed to a large variety of background noise conditions which may degrade recognition performance. This topic has been extensively studied, notably by modifying the front-end to derive robust ASR features or the back-end by adapting the acoustic models to the present acoustic conditions. A possible extension to our research could be to perform joint background noise conditions evaluation and weighted Viterbi recognition to obtain improved robustness against acoustic noise.

APPENDIX A

List of Abbreviations

ABS	Analysis by Synthesis
ACELP	Algebraic CELP coding
ADPCM	Adaptive Differential Pulse Coded Modulation
AMR	Adaptive Multi-Rate
AR	Auto-Regressive
ASR	Automatic Speech Recognition
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BES	Bit Error Sensitivity
BICM	Bit-Interleaved Coded Modulation
BPS	Bit Per Second
BSC	Binary Symmetric Channel
CD	Correct Detection
CELP	Coded Excitation Linear Prediction
CPD	Code Product Distance
CPPD	Code Periodic Product Distance
CRC	Cyclic Redundancy Check
DCT	Discrete Cosine Transform
DED	Double Error Detection
DMR	Digital Multi-Rate
DPCM	Differential Pulse Code Modulation
DRT	Diagnostic Rhyme Test
DSR	Distributed Speech Recognition
ECL	Effective Code Length
ED	Error Detection
EFR	Enhanced Full Rate
EMBSD	Enhanced Modified Bark Spectral Distortion
ETSI	European Telecommunications Standardization Institute

FB	Feed-Back
FEC	Forward Error Correction
FF	Feed-Forward
FS	Federal Standard
GSM	Group Special Mobile
HMM	Hidden Markov Model
IDCT	Inverse Discrete Cosine Transform
IIR	Infinite Impulse Response
ITU	International Telecommunication Union
JND	Just Noticeable Distortion
KBPS	Kilo Bit Per Second
LAR	Log-Area Ratio
LP	Linear Prediction
LPC	Linear Predictive Coding
LPCC	Linear Prediction Cepstral Coefficients
LSF	Line Spectral Frequencies
LTP	Long-Term Prediction
LVCSR	Large Vocabulary Continuous Speech Recognition
MELP	Mixed Excitation Linear Prediction
MFCC	Mel Frequency Cepstrum Coefficients
ML	Maximum Likelihood
MOS	Mean Opinion Score
MPEG	Motion Picture Expert Group
MSVQ	Multi-Stage Vector Quantization
NMC	Noise-Masking Curve
PCM	Pulse Code Modulation
PDA	Personal Digital Assistant
PECL	Periodic Effective Code Length
PESQ	Perceptual Evaluation of Speech Quality
PLP	Perceptual Linear Prediction
PSK	Phase Shift Keying
PVQ	Predictive Vector Quantization

QAM	Quadrature Amplitude Modulation
QMF	Quadrature Mirror Filter
RC	Reflection Coefficient
RCPC	Rate-Compatible Punctured Convolutional code
RCPT	Rate-Compatible Punctured Trellis code
RED	Residual Euclidean Distance
RHD	Residual Hamming Distance
SBC	Subband Coding
SEC	Single Error Correction
SED	Single Error Detection
SMR	Signal-to-Mask Ratio
SNR	Signal-to-Noise Ratio
STP	Short-Term Prediction
TCM	Trellis Code Modulation
TED	Triple Error Detection
UE	Undetected Error
UEP	Unequal Error Protection
VA	Viterbi Algorithm
VBR	Variable Bit Rate
VQ	Vector Quantization
WER	Word Error Rate
WVR	Weighted Viterbi Recognition

References

[1] B. Strope, *Modeling auditory perception for robust speech recognition*, Ph.D. thesis, University of California, Los Angeles, 1999.

[2] A.M. Kondoz, *Digital speech coding for low bit rate communication systems*, Wiley, England, 1995.

[3] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.0)*, July 2000.

[4] G. Fant, *Acoustic theory of speech production*, Mounton and Co., Gravenhage, The Netherlands, 1960.

[5] J. Flanagan, *Speech analysis, synthesis and perception*, Springer-Verlag, New York, 1972.

[6] L. Rabiner and B.H. Juang, *Fundamentals of speech recognition*, Prentice Hall, Englewood, New Jersey, 1993.

[7] B.W. Kleijn and K.K. Paliwal, *Speech coding and synthesis*, Elsevier, Amsterdam, Netherlands, 1995.

[8] T. Painter and A. Spanias, "A review of algorithms for perceptual coding of digital audio signals," in *International Conference on Digital Signal Processing Proceedings*, July 1997, vol. 1, pp. 179–208.

[9] A. Alwan, S. Narayanan, B. Strope, and A. Shen, *Speech production and perception models and their applications to synthesis, recognition, and coding*, chapter in book "Speech processing, recognition, and artificial neural networks", pp. 138–161, Springer-Verlag, UK, 1999.

[10] E. Zwicker and H. Fastl, *Psychoacoustics*, Springer-Verlag, Berlin, Germany, 1990.

[11] B. Moore, *An introduction to the psychology of hearing*, Academic Press, London, UK, 1989.

[12] C.D. Geisler, *From sound to synapse: Physiology of the mammalian ear*, Oxford University Press, New York, 1998.

[13] H. Fletcher, "Loudness, masking and their relation to the hearing process and the problem of noise measurement," *Journal of the Acoustical Society of America*, vol. 42, pp. 275–293, 1938.

[14] J.B. Allen, "Fletcher's role in the creation of communication acoustics," *Journal of the Acoustical Society of America*, vol. 99, pp. 1825–1839, 1996.

[15] S.S. Stevens and H.W. Davis, *Hearing*, John Wiley & Sons, New York, 1938.

[16] H. Fletcher, "Auditory patterns," *Review of Modern Physics*, pp. 47–65, 1940.

[17] K. Brandenburg and G. Stoll, "ISO MPEG-1 audio: A generic standard for coding of high quality digital audio," *Journal of Audio Engineering Society*, vol. 42, no. 10, pp. 780–792, Oct. 1994.

[18] S. Shlieng, "Guide to MPEG-1 audio standard," *IEEE Transactions on Broadcasting*, vol. 40, no. 4, pp. 206–218, Dec. 1994.

[19] M.R. Schroeder, "A brief history of speech coding," *Proceedings International Conference on Communications*, pp. 26.01.1–4, Sept. 1992.

[20] A. Gersho, "Advances in speech and audio compression," *IEEE Transactions on Speech and Audio Processing*, vol. 82, no. 6, pp. 900–918, June 1994.

[21] N. Jayant, "Signal compression: Technology targets and research directions," *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 5, pp. 796–818, June 1992.

[22] T. Kailath, *Lectures on Wiener and Kalman filtering*, Springer Verlag, New York, 1981.

[23] L.R. Rabiner and R.W. Schafer, *Digital signal processing of speech signals*, Prentice Hall, New Jersey, 1978.

[24] F. Itakura, "Line spectrum representation of linear predictive coefficients," *Journal of the Acoustical Society of America*, vol. 57, pp. S35, 1975.

[25] N. Sugamara and F. Itakura, "Line spectrum representation of linear predictor coefficients of speech signal and its statistical properties," *Transactions Inst. Electron., Commun. Eng. Japan*, vol. J64-A, pp. 3230–40, July 1981.

[26] S. Quackenbush, T. Barnwell, and M. Clements, *Objective measures for speech quality*, Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[27] International Telecommunication Union, "Objective quality measurement of telephone-band (300–4000 Hz) speech codecs," *Recommendation P.861, Appendix II*, Feb. 1998.

[28] A.W. Rix, J.G. Beerends, M.P. Hollier, and A.P. Hekstra, "Perceptual evaluation of speech quality (PESQ)," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 2001, vol. 2, pp. 749–52.

[29] S. Furui, *Digital speech processing, synthesis and recognition*, Dekker, New York, 1985.

[30] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–52, April 1990.

[31] Mark Hasegawa-Johnson, *EE 214B lecture notes*, University of California, Los Angeles, Los Angeles, California, 1998.

[32] G.D. Forney, "The Viterbi algorithm," *IEEE Transactions on Communications*, vol. 61, no. 3, pp. 268–278, Apr. 1973.

[33] C.-E. Shannon, "A mathematical theory of communication," *Bell Syst. Technical Journal*, vol. 27, pp. 379–423, 1948.

[34] W.W. Peterson, *Error-correction codes*, MIT Press and Wiley & Sons, 1961.

[35] C.-E. Shannon, "A mathematical theory of communication," *Bell Syst. Technical Journal*, vol. 27, pp. 623–656, 1948.

[36] J. Proakis, *Digital communications*, McGraw-Hill, 1995.

[37] S. Wilson, *Digital modulation and coding*, Prentice Hall, 1996.

[38] S. Lin and D. Costello, *Error control coding: Fundamentals and applications*, Prentice Hall, New Jersey, 1983.

[39] G. Ungerboeck, "Channel encoding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. 28, no. 1, pp. 55–67, Jan. 1982.

[40] D. Divsalar, M.K. Simon, P. McLane, and E. Biglieri, *Trellis code modulation*, Macmillan, New York, 1991.

[41] A. Gersho and E. Paksoy, "An overview of variable rate speech coding for cellular networks," in *Proceedings IEEE International Conference on Selected Topics in Wireless Communications*, June 1999, pp. 172–175.

[42] J. Vainio, H. Mikkola, K. Jarvinen, and P. Haavisto, "GSM EFR based multi-rate codec family," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 1998, vol. 1, pp. 141–144.

[43] A. Uvliden, S. Bruhn, and R. Hagen, "Adaptive multi-rate. A speech service adapted to cellular radio network quality," in *Proceedings of the 32nd Asilomar Conference on Signals, Systems and Computers*, Nov. 1998, vol. 1, pp. 343–347.

[44] E. Paksoy, J. Carlos de Martin, A. McCree, C. Gerlach, A. Anandakumar, M. Lai, and V. Viswanathan, "An adaptive multi-rate speech coder for digital cellular telephony," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, March 1999, vol. 1, pp. 193–196.

[45] H. Ito, M. Serizawa, K. Ozawa, and T. Nomura, "An adaptive multi-rate speech codec based on MP-CELP coding algorithm for ETSI AMR standard," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, Apr. 1998, vol. 1, pp. 137–140.

[46] D. Sinha and C.-E. Sundberg, "Unequal error protection methods for perceptual audio coders," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, March 1999, vol. 5, pp. 2423–2426.

[47] J. Hagenauer, "Rate-compatible punctured convolutional codes and their applications," *IEEE Transactions on Communications*, vol. 36, no. 4, pp. 389–400, Apr. 1998.

[48] B. Masnick and J. Wolf, "On linear unequal error protection codes," *IEEE Transactions on Information Theory*, vol. 3, no. 5, pp. 600–607, Oct. 1967.

[49] R. Cox, J. Hagenauer, N. Seshadri, and C.-E. Sundberg, "Subband speech coding and matched convolutional channel coding for mobile radio channels," *IEEE Transactions on Signal Processing*, vol. 39, no. 8, pp. 1717–31, Aug. 1991.

[50] D.J. Goodman and C.-E. Sundberg, "Combined source and channel coding for variable bit-rate speech transmission," *Bell System Technical Journal*, vol. 7, pp. 2017–36, Sept. 1983.

[51] D.J. Goodman and C.-E. Sundberg, "Transmission errors and forward error correction in embedded differential PCM," *Bell System Technical Journal*, vol. 9, pp. 2735–64, Nov. 1983.

[52] A. Bernard, X. Liu, R. Wesel, and A. Alwan, "Speech transmission using rate-compatible trellis codes and embedded source coding," *IEEE Transactions on Communications*, vol. 50, no. 2, pp. 309–320, Feb. 2002.

[53] T. Salonidis and V. Digalakis, "Robust speech recognition for multiple topological scenarios of the GSM mobile phone system," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 1998, pp. 101–4.

[54] S. Dufour, C. Glorion, and P. Lockwood, "Evaluation of the root-normalised front-end (RN LFCC) for speech recognition in wireless GSM network environments," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 1996, vol. 1, pp. 77–80.

[55] L. Karray, A. Jelloun, and C. Mokbel, "Solutions for robust recognition over the GSM cellular network," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, 1998, vol. 1, pp. 166–170.

[56] A. Gallardo, F. Diaz, and F. Vavlerde, "Avoiding distortions due to speech coding and transmission errors in GSM ASR tasks," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 1999, pp. 277–80.

[57] C. Mokbel, L. Mauray, L. Karray, D. Jouvet, J. Monne, C. Sorin, J. Simonin, and K. Bartkova, "Towards improving ASR robustness for PSN

and GSM telephone applications," *Speech Communication*, vol. 23, pp. 141–59, Oct. 1998.

[58] A. Bernard and A. Alwan, "Joint channel decoding - Viterbi recognition for wireless applications," in *Proceedings of Eurospeech*, Sept. 2001, vol. 4, pp. 2703–6.

[59] A. Potamianos and V. Weerackody, "Soft-feature decoding for speech recognition over wireless channels," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 2001, vol. 1, pp. 269–72.

[60] B. Milner and S. Semnani, "Robust speech recognition over IP networks," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, June 2000, vol. 3, pp. 1791–4.

[61] B. Milner, "Robust speech recognition in burst-like packet loss," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 2001, vol. 1, pp. 261–4.

[62] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, vol. 12, no. 5, pp. 40–48, Sept./Oct. 1998.

[63] A. Gersho and S. Gray, *Vector quantization and signal compression*, Kluwer, Doordrecht, 1992.

[64] N. Jayant, J. Johnston, and R. Safranek, "Signal compression based on models of human perception," *Proceedings of the IEEE*, vol. 81, no. 10, pp. 1385–1422, Oct. 1993.

[65] T. Ramstad, "Sub-band coder with a simple adaptive bit allocation algorithm," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 1982, vol. 1, pp. 203–207.

[66] B. Tang, A. Shen, A. Alwan, and G. Pottie, "A perceptually-based embedded sub-band speech coder," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 2, pp. 131–140, March 1997.

[67] A. Shen, "Perceptually based subband coding of speech signals," M.S. thesis, University of California, Los Angeles, 1994.

[68] Z. Jiang, A. Alwan, and A. Wilson, "High-performance IIR QMF banks for speech subband coding," in *Proceedings of IEEE ISCAS*, June 1994, vol. 2, pp. 493–6.

[69] R. Cox, S. Gay, N. Seshadri, Y. Shoham, S. Quackenbush, and N. Jayant, "New directions in subband coding," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 2, pp. 391–409, Feb. 1988.

[70] N. Rydbeck and C.-E. Sundberg, "Analysis of digital errors in non-linear PCM systems," *IEEE Transactions on Communications*, vol. 24, pp. 59–65, Jan. 1976.

[71] C.-E. Sundberg, "The effect of single bit errors in standard non-linear PCM systems," *IEEE Transactions on Communications*, vol. 24, pp. 1062–64, June 1976.

[72] V. Zue, S. Seneff, and J. Glass, "Speech database development at MIT: TIMIT and beyond," *Speech Communication*, vol. 9, no. 6, pp. 351–6, Aug. 1990.

[73] International Telecommunication Union, "5, 4, 3 and 2 bits samples embedded adaptive differential pulse code modulation (ADPCM) Annex A: Extensions for use with uniform-quantized input and output," *Recommendation G.727*, 1990.

[74] International Telecommunication Union, "5, 4, 3 and 2 bits samples embedded adaptive differential pulse code modulation (ADPCM)," *Recommendation G.727*, 1990.

[75] A. McCree, K. Truong, E. Bryan George, T.P. Barnwell, and V. Viswanathan, "A 2.4 kbit/s MELP coder candidate for the new U.S. federal standard," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, Apr. 1996, vol. 1, pp. 200–3.

[76] J.B. Cain, G.C. Clark, and J.M. Geist, "Punctured convolutional code of rate (N-1)/N and simplified maximum likelihood decoding," *IEEE Transactions on Information Theory*, vol. 20, pp. 388–9, May 1974.

[77] Y. Yasuda, K. Kashiki, and Y. Hirata, "High rate punctured convolutional codes for soft decision Viterbi decoding," *IEEE Transactions on Communications*, vol. 32, no. 3, pp. 315–319, Mar. 1984.

[78] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft decision outputs and its applications," in *Proceedings of IEEE Globecom*, Nov. 1989, vol. 3, pp. 1680–3.

[79] J. Hagenauer, N. Seshadri, and C.-E. Sundberg, "The performance of rate-compatible punctured convolutional codes for digital mobile radio," *IEEE Transactions on Communications*, vol. 38, no. 7, pp. 966–980, July 1990.

[80] R.D. Wesel, X. Liu, and W. Shi, "Periodic symbol puncturing of trellis codes," in *Proceedings of the 31st Asilomar Conference on Signals, Systems and Computers*, Nov. 1997, vol. 1, pp. 172–6.

[81] R.D. Wesel, X. Liu, and W. Shi, "Trellis codes for periodic erasures," *IEEE Transactions on Communications*, vol. 58, no. 5, pp. 938–47, June 2000.

[82] L.H.C. Lee, "New rate-compatible punctured convolutional codes for Viterbi decoding," *IEEE Transactions on Communications*, vol. 41, no.

12, pp. 3073–79, Dec. 1994.

[83] J.B. Anderson and K. Balachandran, "Decision depths of convolutional codes," *IEEE Transactions on Information Theory*, vol. 35, no. 2, pp. 455–9, March 1989.

[84] R.D. Wesel and X. Liu, "Analytic techniques for periodic trellis codes," in *Proceedings of the 36th Allerton Conference on Communications, Control and Computing*, Sept. 1998, pp. 39–48.

[85] C. Fragouli, C. Kominakis, and R.D. Wesel, "Minimality for punctured convolutional codes," in *Proceedings of ICC 2001*, June 2001, vol. 1, pp. 300–4.

[86] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 927–946, May 1998.

[87] E. Zehavi, "8-PSK trellis codes for a Rayleigh channel," *IEEE Transactions on Communications*, vol. 40, no. 5, pp. 873–884, May 1992.

[88] D. Divsalar and M.K. Simon, "The design of trellis coded MPSK for fading channels: Performance criteria," *IEEE Transactions on Communications*, vol. 36, no. 9, pp. 1004–12, Sept. 1988.

[89] Y.S. Leung, S.G. Wilson, and J.W. Ketchum, "Multifrequency trellis coding with low delay for fading channels," *IEEE Transactions on Communications*, vol. 41, no. 10, pp. 1450–9, Oct. 1993.

[90] R.D. Wesel, *Trellis code design for correlated fading and achievable rates Tomlinson-Harashima precoding*, Ph.D. thesis, Stanford University, Aug. 1996.

[91] C.-E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," in *IRE Nat. Conv. Rec.*, 1959, pp. 142–163.

[92] A. Bernard, X. Liu, R. Wesel, and A. Alwan, "Channel adaptive joint-source channel coding of speech," in *Proceedings of the 32nd Asilomar Conference on Signals, Systems and Computers*, Nov. 1998, vol. 1, pp. 357–61.

[93] A. Bernard, X. Liu, R. Wesel, and A. Alwan, "Embedded joint-source channel coding of speech using symbol puncturing of trellis codes," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, March 1999, vol. 5, pp. 2427–30.

[94] S. Euler and J. Zinke, "The influence of speech coding algorithms on automatic speech recognition," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, 1994, pp. 621–4.

[95] B.T. Lilly and K.K. Paliwal, "Effect of speech coders on speech recognition performance," in *Proceedings of ICSLP*, Oct. 1996, vol. 4, pp. 2344–7.

[96] L. Yapp and G. Zick, "Speech recognition on MPEG/audio encoded files," in *IEEE International Conference on Multimedia Computing and Systems*, June 1997, pp. 624–5.

[97] J. Huerta and R. Stern, "Speech recognition from GSM parameters," in *Proceedings of ICSLP*, 1998, vol. 4, pp. 1463–6.

[98] S.H. Choi, H.K. Kim, H.S. Lee, and R.M. Gray, "Speech recognition method using quantised LSP parameters in CELP-type coders," *Electronics Letters*, vol. 34, no. 2, pp. 156–7, Jan. 1998.

[99] H.K. Kim and R.V. Cox, "Feature enhancement for a bitstream-based front-end in wireless speech recognition," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 2001, vol. 1, pp. 241–3.

[100] H.K. Kim and R. Cox, "Bitstream-based feature extraction for wireless speech recognition," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, 2000, vol. 1, pp. 21–24.

[101] S.H. Choi, H.K. Kim, and H.S. Lee, "Speech recognition using quantized LSP parameters and their transformations in digital communications," *Speech Communication*, vol. 4, no. 30, pp. 223–33, April 2000.

[102] H.Y. Hur and H.S. Kim, "Formant weighted cepstral feature for LSP-based speech recognition," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 2001, vol. 1, pp. 141–4.

[103] V. Digalakis, L. Neumeyer, and M. Perakakis, "Quantization of cepstral parameters for speech recognition over the World Wide Web," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 1, pp. 82–90, Jan. 1999.

[104] D. Pearce, "Enabling new speech driven services for mobile devices: An overview of the ETSI standards activities for distributed speech recognition front-ends," in *Applied Voice Input/Output Society Conference (AVIOS2000)*, May 2000.

[105] A. Bernard and A. Alwan, "Source and channel coding for remote speech recognition over error-prone channels," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 2001, vol. 4, pp. 2613–6.

[106] G. Ramaswamy and P. Gopalakrishnan, "Compression of acoustic features for speech recognition in network environments," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 1998, vol. 2, pp. 977–80.

[107] N. Srinivasamurthy, A. Ortega, Q. Zhu, and A. Alwan, "Towards efficient

and scalable speech compression schemes for robust speech recognition applications," in *Proceedings of ICME*, 2000, vol. 1, pp. 249–52.

[108] J.P. Campbell, V.C. Welch, and T.E. Tremain, "The new 4800 bps voice coding standard," in *Proceedings of Military Speech Technology*, 1989, pp. 64–70.

[109] A. McCree and T.P. Barnwell III, "A mixed excitation LPC vocoder model for low bit rate speech coding," *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 242–50, July 1995.

[110] W. LeBlanc, C. Liu, and V. Viswanathan, "An enhanced full rate speech coder for digital cellular applications," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 1996, vol. 1, pp. 569–72.

[111] D.L. Thomson and R. Chengalvarayan, "Use of periodicity and jitter as speech recognition features," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 1998, vol. 1, pp. 21–4.

[112] W. Gunawan and M. Hasegawa-Johnson, "PLP coefficients can be quantized at 400 bps," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 2001, vol. 1, pp. 77–80.

[113] 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, 1993.

[114] R. Viswanathan and J. Makhoul, "Quantization properties of transmission parameters in linear predictive systems," *IEEE Transactions on Speech and Audio Processing*, vol. 23, pp. 309–21, March 1975.

[115] K.K. Paliwal, "Interpolation properties of linear prediction parametric representations," in *Proceedings of Eurospeech*, 1995, pp. 1029–32.

[116] B. Atal, "Efficient coding of LPC parameters by temporal decomposition," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, April 1983, vol. 1, pp. 81–84.

[117] M. Yong *et al.*, "Encoding of LPC spectral parameters using switched adaptive inter frame vector prediction," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, April 1988, vol. 1, pp. 402–405.

[118] F. Soong and B. Juang, "Line spectrum pairs and speech data compression," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, 1984, vol. 1, pp. 1–4.

[119] B. Atal, R.B. Cox, and P. Kroon, "Spectral quantization and interpolation for CELP coders," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 1989, vol. 1, pp. 69–72.

[120] W. Gardner and B.D. Rao, "Theoretical analysis of the high-rate vector quantization of LPC parameters," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 5, pp. 367–381, Sept. 1995.

[121] V. Digalakis, S. Tsakalidis, C. Harizakis, and L. Neumeyer, "Efficient speech recognition using subvector quantization and discrete-mixture HMMs," *Computer Speech and Language*, vol. 14, no. 1, pp. 33–46, Jan. 2000.

[122] Q. Zhu and A. Alwan, "An efficient and scalable 2D-DCT based feature coding scheme for remote speech recognition," in *Proceedings of the International Conference on Audio, Speech and Signal Processing*, May 2001, vol. 1, pp. 113–6.

[123] E.N. Gilbert, "Capacity of burst noise channel," *Bell System Technical Journal*, vol. 39, pp. 1253–65, Sept. 1960.

[124] T. Cover and J. Thomas, *Elements of information theory*, Wiley, New York, 1991.

[125] J. Hagenauer, "Source controlled channel decoding," *IEEE Transactions on Communications*, vol. 43, no. 9, pp. 2449–57, Sept. 1995.

[126] I. Kozintsev and K. Ramchandran, "Robust image transmission over energy-constrained time-varying channels using multiresolution joint source-channel coding," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 1012–26, Apr. 1998.