8

Error Recovery- Channel Coding and Packetization

Bengt J. Borgström, Alexis Bernard, and Abeer Alwan

Abstract. Distributed Speech Recognition (DSR) systems rely on efficient transmission of speech information from distributed clients to a centralized server. Wireless or network communication channels within DSR systems are typically noisy and bursty. Thus, DSR systems must utilize efficient Error Recovery (ER) schemes during transmission of speech information. Some ER strategies, referred to as forward error control (FEC), aim to create redundancy in the source coded bitstream to overcome the effect of channel errors, while others are designed to create spread or delay in the feature stream in order to overcome the effect of bursty channel errors. Furthermore, ER strategies may be designed as a combination of the previously described techniques. This chapter presents an array of error recovery techniques for remote speech recognition applications.

This chapter is organized as follows. First, channel characterization and modeling are discussed. Next, media-specific FEC is presented for packet erasure applications, followed by a discussion on media-independent FEC techniques for bit error applications, including general linear block codes, cyclic codes, and convolutional codes. The application of unequal error protection (UEP) strategies utilizing combinations of the aforementioned FEC methods is also presented. Finally, frame-based interleaving is discussed as an alternative to overcoming the effect of bursty *channel erasures*. The chapter concludes with examples of modern standards for channel coding strategies for distributed speech recognition (DSR).

8.1. Distributed Speech Recognition Systems

Throughout this chapter various error recovery and detection techniques are discussed. It is therefore necessary to present an overview of a complete experimental DSR system, including feature extraction, a noisy channel model, and an automatic speech recognition engine at the server end (Fig. 8.1).

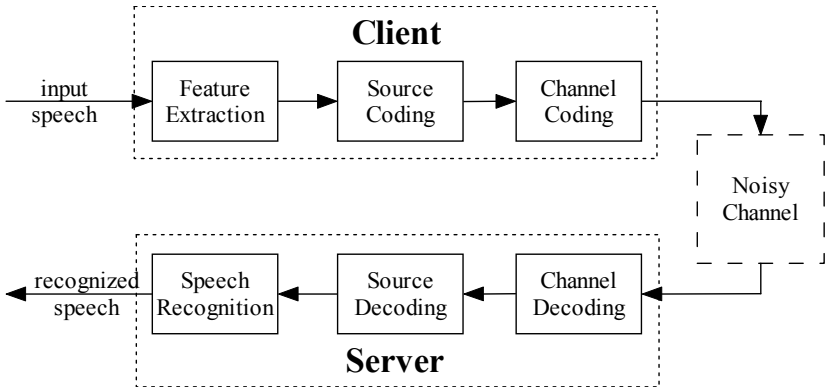

Fig. 8.1. Overview of the complete distributed speech recognition system

The *feature extraction* and source coding algorithms implemented for this chapter are similar to those described by the *ETSI* standards (ETSI 2000), whereby split vector quantization (SVQ) is used to compress the first 13 Mel-Frequency Cepstral Coefficients (MFCCs) as well as the log-energy of the speech frame. The SVQ then allocates 8 bits to the vector-quantization of the log-energy and the 0th cepstral coefficient pair, and 6 bits to each of the following 6 pairs. The vector quantizers were trained using the K-means algorithm, and quantization was carried out via an exhaustive search.

Two types of communication channels are studied, wireless circuit-switched and IP packet-switched. Although these channels are inherently different, they do share the characteristic that errors, whether they are flipped bits or packet erasures, tend to occur in bursts. Thus, similar models are used to simulate the effects of noisy channels in the wireless and IP network scenarios. Channel degradation and modeling are discussed in further detail in Section 8.2.

The server-end speech recognition engine is implemented using the HTK toolkit (Young, Kershaw, Odell, Ollason, Valtchev, and Woodland 2000). The training process uses an implementation of the forward-backward algorithm, and the recognition process uses an implementation of the Viterbi algorithm. The speech database used for experiments is the Aurora-2 database (Hirsch and Pearce 2000), which consists of connected digit strings, spoken by various male and female speakers. During the recognition process, 16-state word models were used, with each state comprised of 3 mixtures. 8440 digit utterances were used for training, and 1001 utterances were used for testing (500 males, 501 females for a total of 3257 digits).

8.2. Characterization and Modeling of Communication Channels

Distributed speech recognition systems face the challenge of processing signal noise induced by communication channels. Such systems transmit extracted speech features from distributed clients to the server, and typically operate at lower bitrates than traditional speech communication systems.

8.2.1 Signal Degradation over Wireless Communication Channels

Signal degradations caused by wireless channels are highly dependent on the specific physical properties of the environment between the transmitter and receiver, and it is a difficult task to accurately generalize performance results of communication over a wireless channel (Sklar 1997; Bai and Atiquzzaman 2003).

There are three general phenomena that affect the propagation of radio waves in wireless communication systems:

1. Reflection: Radio waves are reflected off smooth surfaces which are large in comparison to the wavelength of the wave.
2. Diffraction: Radio waves propagate around relatively large impenetrable objects when there exists no line of sight (LOS) between the transmitter and receiver. This effect is also referred to as "shadowing", since radio waves are able to travel from the transmitter to the receiver even when shadowed by a large object.
3. Scattering: Radio waves are disrupted by objects in the transmission path whose size is smaller than the wavelength of the radio wave.

These propagation phenomena cause fluctuations of the amplitude, phase, and angle of incidence of wireless signals, resulting in multipath propagation from the transmitter to the receiver. Multipath propagation leads to fading characteristics in the received signal. Large-scale fading occurs when the mobile receiver moves to/from the transmitter over large distances. Small-scale fading occurs when the distance between the receiver and transmitter changes in small increments. Thus, the speed of the mobile client has a great effect on the resulting channel behavior. When no dominant path exists, the statistics for the signal envelope can be described by a Rayleigh distribution, and when a dominant LOS path exists, the statistics of the signal envelope can be described by a Ricean distribution.

Since wireless communication systems are generally built upon circuit-switched networks, corrupted data occur as bit errors in the modulated bitstream. Furthermore, due to the fading nature of wireless channels, bit errors tend to occur in bursts. The probability of occurrence and expected duration of bit error bursts are dependent upon the time varying channel signal-to-noise ratio (SNR).

8.2.2 Signal Degradation over IP Networks

IP networks rely on packet-switching, wherein packet loss and delay are caused mainly by congestion at the routers, and individual bit errors rarely occur (Tan, Dalsgaard, and Lindberg 2005). Specifically, packet losses may appear if the input flow of data is higher than the processing capacity of the switching logic, or if the processing capacity of the switching logic is higher than the output flow speed (Kurose and Rose 2003).

Similar to the occurrence of bit-error bursts in wireless networks, packet losses in packet-switched IP networks tend to occur in bursts (Jiao, Schwiebert, and Xu 2002; Bolot 2003). Probability of occurrence and expected duration of the packet loss bursts are dependent on the congestion of the network.

8.2.3 Modeling Bursty Communication Channels

A common method for simulating a bursty channel in a communication system is the two-state *Gilbert-Elliot (GE) model* (Elliot 1963), which has been widely used for DSR studies. The GE model includes a good state, S_g, which incurs no loss, and a bad state, S_b, which assumes some probability of loss, as is illustrated in Fig. 8.2. Here, p represents the probability of transitioning from S_g to S_b, whereas q represents the probability of transitioning from S_b to S_g. To completely characterize the model, the parameter h, which represents the probability of loss while in state S_b, is used.

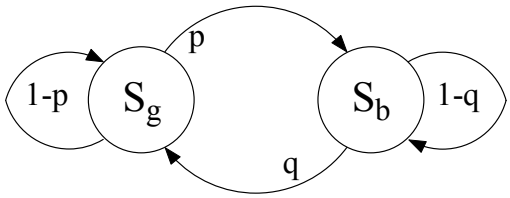

Fig. 8.2. The Gilbert-Elliot Model for simulating bursty channels: S_g and S_b represent the good and bad states, respectively, and p and q represent transitional probabilities.

8.2.3.1 Bit-Level Channel Models

Throughout this chapter, specific bit-level channel conditions are used for simulations and the probability of transition from S_g to S_b is set to p=0.002. The noise incurred in S_b is additive white noise at a level of *2* dB, corresponding to an error probability of $P_{err}|S_b$=0.157, whereas the noise incurred in S_g is at a level of *20* dB, corresponding to an error probability of $P_{err}|S_g$≈0.00. Thus, the remaining variable is the transitional probability q. The conditions to be used, similar to those described in (Han, Srinivasamurthy, and Narayanan 2004), are shown in Table 8.1.

Table 8.1. Channel conditions used for bit-level simulations

Channel Condition	*p*	*q*	*BER*
Clean	0.000	1.000	0.00 %
1	0.002	0.019	1.50 %
2	0.002	0.009	2.87 %
3	0.002	0.0057	4.10 %
4	0.002	0.003	6.32 %
5	0.002	0.002	7.90 %
6	0.002	0.0013	9.58 %
7	0.002	0.0009	10.90 %

8.2.3.2 Packet-Level Channel Models

An efficient method to simulate errors introduced by IP networks is to apply the GE model on the packet level, and thus the GE model iterates at each packet. Furthermore, the loss incurred in S_b represents a packet erasure.

Throughout this chapter, specific packet-level channel conditions will be used for simulations. These conditions, as described in (Han et al. 2004), are shown in Table 8.2. Note that for simulating packet-level channels, the probability of packet loss while in S_b is assumed to be 1.0. In Fig. 8.2, the average burst duration can be determined as $1/q$, and the percentage of packets lost can be calculated as $100[p/(p+q)]$%.

Table 8.2. Channel conditions used for packet-level simulations

Channel Condition	*p*	*q*	*packets lost*	*ave. burst duration*
Clean	0.000	1.000	0.0 %	0.00 frames
1	0.005	0.853	0.6 %	1.17 frames
2	0.066	0.670	9.0 %	1.49 frames
3	0.200	0.500	28.6 %	2.00 frames
4	0.250	0.400	38.5 %	2.50 frames
5	0.300	0.300	50.0 %	3.33 frames
6	0.244	0.200	55.0 %	5.00 frames

8.3. Media-Specific FEC

Media-specific FEC involves insertion of additional copies of speech features into the DSR datastream prior to transmission. The additional copies, referred to as replicas, are coarsely quantized in order to reduce the additional required bandwidth. Furthermore, due to the bursty nature of both wireless and IP-network transmission, speech feature replicas are inserted into the source-coded bitstream at certain frame intervals away from the original features.

In (Peinado, Gomez, Sanchez, Perez-Cordoba, and Rubio 2005a) the authors introduce a *media-specific FEC* method compatible with the *ETSI* DSR source coding standards (ETSI 2000). The *media-specific FEC* algorithm creates speech feature replicas by using B_{vq}-bit vector quantization applied to the entire 14-element feature vector of each speech frame. Additionally, the VQ replicas are inserted into the bitstream at intervals of T_{fec} frames away from the original speech feature, with the aim of overcoming the effects of clustered packet erasures characteristic of bursty channels. In general, optimal performance of media-specific FEC can be expected if T_{fec} is chosen to be at least as long as the expected burst duration. Figure 8.3 illustrates an example of media-specific FEC packetization for T_{fec}=4. Here, the source-coded frames are shown along the top, and the FEC replicas are shown along the bottom. Note the separation between coded frames and corresponding vector quantized replicas. Additionally, Fig. 8.4 shows word-accuracy results obtained for various values of B_{vq}, for T_{fec} =6 frames, and for various channel conditions described in Table 8.2.

	packet #4		packet #5		packet #6		packet #7	
Features	7	8	9	10	11	12	13	14
VQ Replicas	3	12	5	14	7	16	9	18

Fig. 8.3. Example of Media-Specific FEC packetization using vector-quantized replicas with T_{fec}=4 (based on Peinado et al. 2005a).

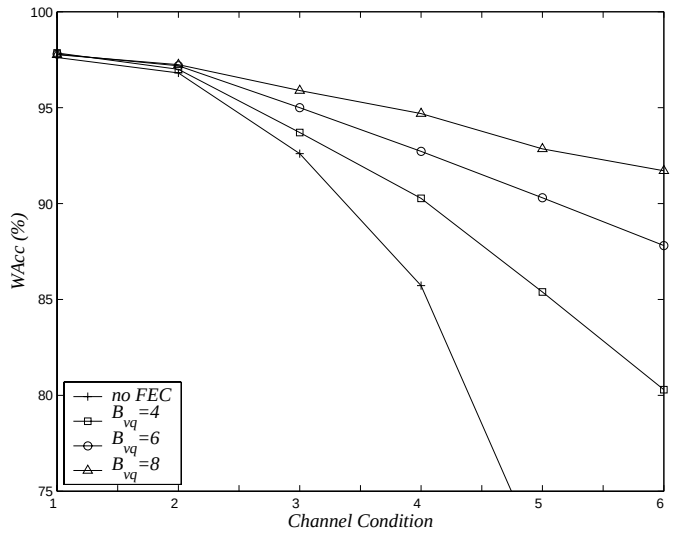

Fig. 8.4. Word-accuracy results obtained using the Aurora-2 database with Media-Specific FEC using B_{vq}-bit VQ replicas and packetization with T_{fec}=6.

As can be concluded from Fig. 8.4, *media-specific FEC* provides improved performance for DSR systems in the case of bursty channels, as compared with transmission without VQ replicas. However, the recognized speech at the server can be delayed by up to T_{fec} frames, which may introduce problems for delay-sensitive applications.

8.4. Media-Independent FEC

Media-independent FEC techniques are applied within DSR systems with the aim of correcting transmission errors or predicting reliability of transmitted speech features, especially with wireless transmission. These techniques include the use of linear block codes, cyclic codes, or convolutional codes. It has been shown in (Bernard and Alwan 2002a) that packet losses or erasures degrade the word-accuracy performance of DSR much less than incorrectly decoded packets. Specifically, it is shown that while channel errors have a disastrous effect on recognition accuracy, the recognizer is able withstand up to 15% of randomly inserted *channel erasures* with negligible loss of accuracy. Therefore, it may at many times be enough for the overall DSR system to simply detect packet errors, as opposed to attempting to decode unreliable data. Thereafter, lost speech features can be estimated at the server through various *Error Concealment* (EC) methods, which are discussed in Chapter 9.

8.4.1 Combining FEC with Error Concealment Methods

There exist numerous error concealment methods to deal with packet erasures in DSR systems which estimate lost speech features before speech recognition. The simplest of such methods include frame dropping and frame repetition. These methods involve low complexity, although they may produce poor recognition accuracy as error burst durations increase (Bernard and Alwan 2002a). Other algorithms to estimate lost speech features include various interpolation techniques, such as linear interpolation or polynomial interpolation, which provide better performance for longer error bursts (Peinado and Segura 2006).

More successful EC methods, however, are based on minimum-mean-square-error (MMSE) algorithms. In (Peindao, Sanchez, Perez-Cordoba, and Rubio 2005b), the authors successfully apply the Forward-Backward MMSE (FB-MMSE) algorithm to determine the maximum likelihood lost or erroneous observations given the most recent correct observation and the nearest correct future observation. However, due to the high complexity of the FB-MMSE algorithm, (Peinado et al. 2005b) introduce simplified versions which greatly decrease the computational load without significantly reducing the performance.

There also exist recognizer-based EC methods which involve soft-feature Viterbi decoding at the server (Bernard and Alwan 2001), known as *weighted viterbi decoding*. In such techniques, channel decoding is performed to determine a measure of reliability of the current received packet. The reliability measure is then passed to the recognition engine, which applies corresponding weights to speech features during the Viterbi algorithm within the recognition process.

8.4.2 Linear Block Codes

For wireless communication channels, the transmitted bitstream generally becomes degraded due to reasons discussed in Section 8.2. An option for providing the detection or correction of transmission errors is through the use of *linear block codes*. *Linear block codes* are especially attractive for the problem of error-robust wireless communication due to their low delay, complexity, and overhead.

The aim of block codes is to provide $m=n-k$ redundancy bits to a block of k dataword bits, resulting in a block of n codeword bits prior to transmission. Such a code is referred to as a (n,k) block code.

Let $D=\{\boldsymbol{d}_1,\boldsymbol{d}_2,\ldots,\boldsymbol{d}_k\}$ represent the set of all possible k-dimensional *datawords*, where $\boldsymbol{d}_i \in \Re^{1\times k}$. Also, let $C=\{\boldsymbol{c}_1,\boldsymbol{c}_2,\ldots,\boldsymbol{c}_n\}$ represent the set of all possible n-dimensional *codewords*, where $\boldsymbol{c}_i \in \Re^{1\times n}$. Thus block codes can be represented in matrix form as $\boldsymbol{c}_i = \boldsymbol{d}_i\boldsymbol{G}$ where the matrix $\boldsymbol{G} \in \Re^{k\times n}$ is referred to as the *generator matrix*.

Furthermore, a code is defined as systematic if each dataword is contained within its corresponding codeword (Blahut 2004). For systematic codes, the *generator matrix* must be in the form

$$\boldsymbol{G} = \left[\boldsymbol{I}_{\mathrm{k}} \;\; \boldsymbol{P}\right], \tag{1}$$

where $\boldsymbol{P} \in \Re^{k\times(n-k)}$ is referred to as the *parity matrix*, and $\boldsymbol{I}_{\mathrm{k}}$ is the $k \times k$ identity matrix. The *parity-check matrix*, $\boldsymbol{H} \in \Re^{n-(k\times n)}$, which is used in the decoding process, is then defined as:

$$\boldsymbol{H} = \left[-\boldsymbol{P}^{\mathrm{T}} \;\; \boldsymbol{I}_{\mathrm{n-k}}\right]. \tag{2}$$

An important parameter to measure the effectiveness of a specific linear block code is the *minimum distance*, d_{min}. The *minimum distance* of a code with codeword set C represents the minimum Hamming distance between any two codewords $\boldsymbol{c}_i, \boldsymbol{c}_j \in C$, for $i \neq j$. It can be shown that the $d_{\min}$ of a systematic code can be determined by finding the minimum Hamming weight of any nonzero codeword in C:

$$d_{\min} = \min_{\boldsymbol{c}\in C, \boldsymbol{c}\neq \boldsymbol{0}} w(\boldsymbol{c}), \tag{3}$$

where the function $w(\cdot)$ represents the Hamming weight, i.e. the number of 1's in a given binary vector.

A linear code with a *minimum distance* of d_{min} is able to detect no more than $d_{\min} - 1$ errors, or is able to correct no more than $\left\lfloor \frac{1}{2}(d_{\min} - 1) \right\rfloor$ errors.

In (Bernard and Alwan 2002a), the authors provide analysis of the performance of systematic *linear block codes* for the application of DSR over wireless channels. In this work, good codes are determined through exhaustive searches by maximizing $d_{\min}$ for various values of n and k, and minimizing the corresponding weights. Table 8.3 shows the resulting optimal codes. Note that the parity matrix $\boldsymbol{P}$ is given in hexadecimal form.

Table 8.3. Systematic linear block codes for channel coding of speech features (from Bernard and Alwan 2002a, © IEEE 2002)

(n,k)	*m*	$\boldsymbol{P}^T$	d_{min}
(12,10)	2	[1,1,1,2,2,2,3,3,3,3]	2
(12,9)	3	[1,2,3,3,4,5,5,6,7]	2
(12,8)	4	[3,5,6,9,A,D,E,F]	3
(10,8)	2	[1,1,1,2,2,3,3,3]	2
(12,7)	5	[07,0B,0D,0E,13,15,19]	4
(10,7)	3	[1,2,3,4,5,6,7]	2

8.4.2.1 Hard vs. Soft Decoding

Linear block code theory discussed in the previous section builds on arithmetic over Galois Fields (Blahut 2004), most commonly over *GF*(2), i.e. binary data with modulo-2 operations. However, the input data stream at the receiver is in the form of demodulated bits with superimposed channel noise. That is, the received data vector is in the form $\boldsymbol{y}=\boldsymbol{x}+\boldsymbol{n}$, where $\boldsymbol{x}$ is the transmitted codeword and $\boldsymbol{n}$ is the channel noise vector. Binary decisions must be made for each received noisy bit, *y(i)*,before channel decoding of the block can be performed, resulting in the approximated bitstream $\tilde{\boldsymbol{y}}$, where $\tilde{y}(i) \in \{0,1\}$. There exists two classical methods of decoding noisy data into a binary bitstream: hard-decision decoding and soft-decision decoding, which will be denoted as $\tilde{\boldsymbol{y}}_h$ and $\tilde{\boldsymbol{y}}_s$, respectively.

Hard-decision decoding maps each received noisy data vector to an approximated transmitted bitstream by the following relationship:

$$\tilde{y}_h(i) = \arg\min_{b \in \{0,1\}} d_E(b, y(i)), \tag{4}$$

where $d_E(\cdot,\cdot)$ represents the Euclidean distance, and *b* corresponds to possible bit values.

As can be interpreted from Equation 4, *hard-decision decoding* simply entails rounding to the nearest modulated bit. The resulting approximated bitstream, $\tilde{\boldsymbol{y}}_h$, is then used to find the estimated transmitted codeword by minimizing the Hamming distance between itself and all possible true codewords. Thus, the detected codeword chosen for channel decoding, denoted as $\tilde{\boldsymbol{y}}_h^{opt}$,is determined as:

$$\tilde{\boldsymbol{y}}_h^{opt} = \arg\min_{\boldsymbol{d}_i \in D} d_H(\tilde{\boldsymbol{y}}_h, \boldsymbol{d}_i), \tag{5}$$

where $d_H(\cdot,\cdot)$ represents the Hamming distance (Bernard and Alwan 2002a).

Since the mapping function described by Equation 5 is not one-to-one, hard-decision decoding can lead to scenarios in which distinct transmitted codewords may be approximated as the same received codeword. That is, there may exist $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, for $i \neq j$, such that $d_H(\tilde{\boldsymbol{y}}_h^{opt}, \boldsymbol{x}_i) = d_H(\tilde{\boldsymbol{y}}_h^{opt}, \boldsymbol{x}_j)$. In such situations, the decoder can detect an error, but cannot correct the error, since multiple distinct codewords could have been transmitted.

Figure 8.5(a) shows an example of hard-decision decoding. In this example, a (2,1) linear code with a generator matrix ***G***= [1,1] was used to transmit the dataword d=[1]. The next most likely dataword in this example is [-1]. In Fig. 8.5(a), the region labeled **CD** corresponds to the region in which the received noisy data vector **y** would have been correctly decoded. Conversely, **UE** (undetected error) corresponds to the region in which **y** would have been incorrectly decoded. Furthermore, the regions labeled **ED** (error detected) refer to the regions in which **y** could not have been decoded, and an error would have been declared.

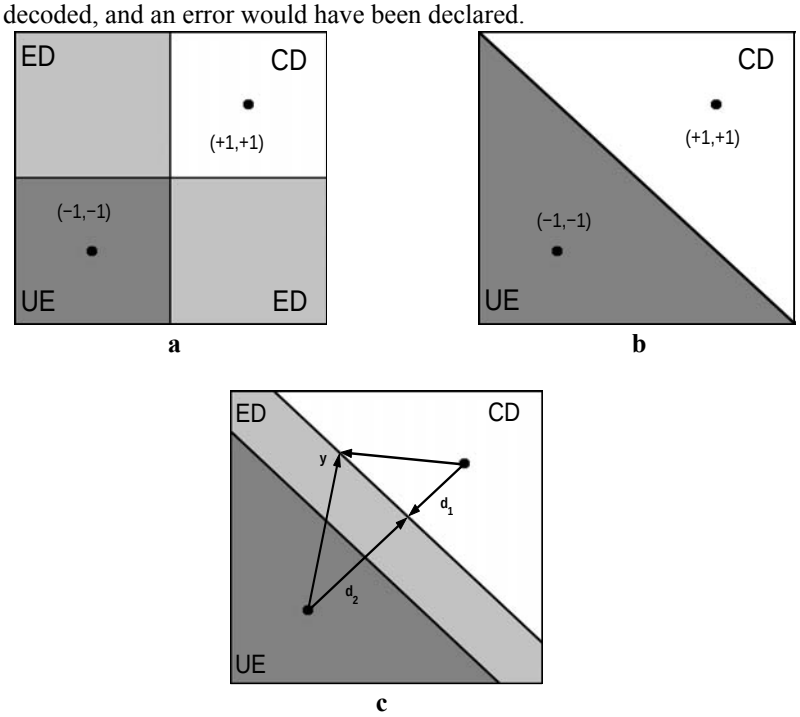

Fig. 8.5. Examples of (a) hard-decision decoding, (b) soft-decision decoding, and (c) λ-soft-decision decoding: This scenario represents a transmitted dataword of *d*=[1], with *G*=[1,1]. Also, CE represents the region corresponding to a corrected error, UE corresponds to the region of an uncorrected error, and DE represents the region corresponding to a detected error (from Bernard and Alwan 2002a, © IEEE 2002).

An alternative to the previously discussed hard-decision decoding is *soft-decision decoding*, which minimizes the Euclidean distance between the received noisy data vector and all possible codewords. The approximated bitstream obtained through soft-decision decoding, denoted by $\widetilde{\boldsymbol{y}}_s$, can be determined as:

$$\widetilde{\boldsymbol{y}}_s^{opt} = \arg\min_{\boldsymbol{d}_i \in D} d_E^2(\boldsymbol{d}_i, \boldsymbol{y}). \tag{6}$$

Figure 8.5(b) illustrates an example of soft-decision decoding for the same channel coding scheme explained for Fig. 8.5(a). Once again, **CD** denotes the region corresponding to a correctly decoded transmitted dataword, and **UE** denotes the region corresponding to an incorrectly decoded transmitted dataword. Note that for soft-decision decoding, all received data vectors must be decoded, and thereis no region for error detection (ED).

For AWGN channels, soft-decision decoding typically shows a 2 dB gain relative to hard-decision decoding in terms of the bit error rate (BER). However, for the specific application of DSR, the authors of (Bernard and Alwan 2002a) show the negative effect of incorrectly decoded transmission errors on word recognition results.

8.4.2.2 λ-Soft Decoding

The adaptive *λ-soft decoding* algorithm presented in (Bernard and Alwan 2002a) provides the error-correcting advantage of soft-decision decoding with the error detecting capabilities of hard-decision decoding by recreating an "error detection" region based on the confidence in the decoding operation.

In order to accept a soft-decision decoded codeword as correct, the λ-soft decoding algorithm compares the likelihood of that transmitted codeword relative to the next most likely codeword. Let $\boldsymbol{y}$ represent the received codeword, and let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ represent the first and second most probable transmitted codewords. The ratio of likelihoods of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ can be expressed as:

$$\frac{P(\boldsymbol{y} \mid \boldsymbol{x} = \boldsymbol{x}_1)}{P(\boldsymbol{y} \mid \boldsymbol{x} = \boldsymbol{x}_2)} = \exp\left(\frac{d_E(\boldsymbol{y}, \boldsymbol{x}_2) - d_E(\boldsymbol{y}, \boldsymbol{x}_1)}{N_o}\right) = \exp\left(\frac{D^2}{N_o} \cdot \frac{d_2 - d_1}{D}\right), \tag{7}$$

where d_1 and d_2 represent the distances from the orthogonal projection of the received codevector $\boldsymbol{y}$ to the line joining $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, and D represents the distance between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Note that $d_2 + d_1 = D$. Also, N_o is a constant related to the noise level of the channel.

The important factor in Equation 7 is $\lambda = \frac{d_2 - d_1}{D}$, since it relates the relative proximity of the received codeword to the two possible transmitted codewords. Note that if $\lambda \approx 0$, the codevectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are almost equiprobable, the soft-decision decoded codeword should be rejected, and an error should be declared. Conversely, if $\lambda \approx 1$, the soft-decision decoded codeword is highly likely. Thus, the region corresponding to error detection (**DE**) grows as λ increases.

For a system implementing the λ-soft decoding algorithm, parameter λ_o is predetermined to perform soft-decision decoding with error detection capability. For received codewords resulting in $\lambda \geq \lambda_o$, soft-decision decoding is utilized, and for $\lambda < \lambda_o$, an error is declared. Recognition accuracy results for hard, soft, and λ-soft decoding are shown in Table 8.4. Here, the λ-soft parameter is set as λ_o=0.16, and the linear block codes used are described in Table 8.3. Note that λ-soft decoding significantly outperforms both hard and soft decoding by reducing the word error rate.

Table 8.4. Word accuracy results using hard-, soft-, and λ-soft decoding over Rayleigh fading channels (from Bernard and Alwan 2002a, © IEEE 2002).

Code (n,k)	*SNR (dB)*	*Hard*	*Soft*	*λ-Soft*
(10,10)	19.96	94.71%	94.71%	98.32%
(10,9)	13.87	97.31%	96.35%	98.12%
(10,8)	10.69	94.47%	95.03%	97.82%
(11,8)	8.80	87.24%	95.62%	97.43%
(12,8)	6.29	67.25%	93.17%	97.04%
(12,7)	4.53	40.48%	91.31%	95.88%

8.4.3 Cyclic Codes

Cyclic codes are a subclass of linear block codes, for which a shift of any codeword is also a codeword (Blahut 2004). That is, if C is a *cyclic code*, then whenever $\boldsymbol{c}=[c_0,c_1,\ldots,c_{n-1}]^T \in C$, this guarantees that $\hat{\boldsymbol{c}}=[c_{n-1},c_0,\ldots,c_{n-2}]^T \in C$. A convenient way to represent cyclic codes is in polynomial form. Thus, the *codeword* $\boldsymbol{c}$ can be described as:

$$\boldsymbol{c} = c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1} = \sum_{j=0}^{n-1} c_j x^j . \tag{8}$$

The *generator matrix* can also be written in polynomial form as:

$$g(x) = \sum_{j=0}^{n-k} g_j x^j . \tag{9}$$

Since a cyclic code is specific to its generator polynomial, a cyclic code can therefore be defined as all multiples of the generator polynomial $g(x)$ by a polynomial of degree $k-1$ or less. Thus, for any polynomial $a_i(x)$ of degree $d \leq k-1$, the corresponding codeword polynomial is given by:

$$c_i(x) = g(x) a_i(x). \tag{10}$$

Cyclic codes are often used for DSR applications. Specifically, three types of cyclic codes are very useful: *cyclic redundancy codes* (CRCs), *Reed-Solomon* (RS) codes, and *Bose-Chaudhuri-Hocquenghem* (BCH) codes.

8.4.4 Convolutional Codes

A *convolutional code* is a channel coding technique which encodes the input bit stream as a linear combination of the current bit and past data in mod-2 arithmetic, and the encoder can be interpreted as a system of binary shift registers and mod-2 addition components. Similar to linear block codes, convolutional codes are characterized by their protection rate: a rate-k/n encoder processes n output bits for every k input bits. A major difference between linear block codes and convolutional codes is the fact that convolutional codes contain memory.

Although there exist various ways to decode convolutional coded data, the most common is the Viterbi algorithm (Viterbi 1971), due to its relatively low computational load. The goal of the Viterbi algorithm is to find the most likely transmitted bitstream by minimizing the total error between the entire received noisy datastream and potential transmitted bitstreams. The Viterbi algorithm accomplishes this minimization by iterating through a trellis, for which the number of states and possible paths within the trellis are determined by the structure of the given encoder (Wesel 2003).

Viterbi decoding of *convolutional codes* can be performed on hard or soft data. That is, the minimization discussed can be carried out on the hard-decision decoded bitstream, which can be determined from the received datastream through Equation 5, or it can be carried out directly on the received soft data. Typically, soft-decision decoding outperforms hard-decision decoding by approximately 2 dB.

Convolutional codes also provide added flexibility to adjust the final protection rate through puncturing. Puncturing entails deleting certain outputs bits according to a specific pattern to obtain a lower final bitrate. For example, a rate-$1/2$ encoder can be punctured by deleting every 4th bit, and the resulting code will be rate-$2/3$. By

puncturing bits in convolutional codes, the channel coding rate can be varied instantaneously, allowing for *rate compatible punctured codes* (RCPC) (Bernard, Liu, Wesel, and Alwan 2002b). Word recognition results for error protection schemes based on convolution codes are shown in Fig. 8.6.

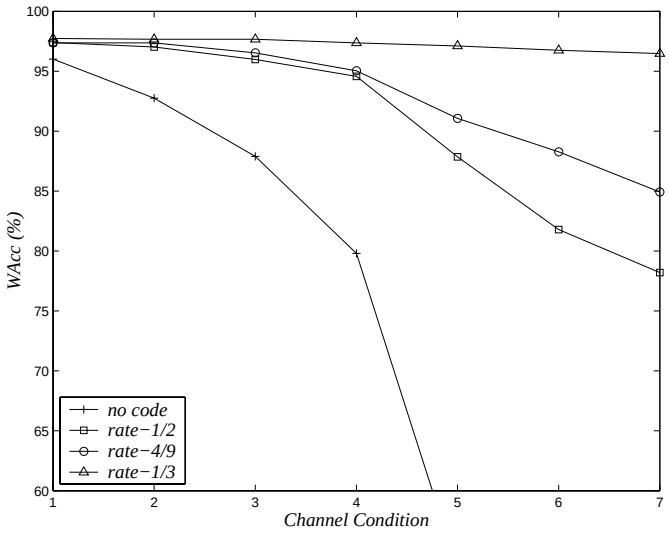

Fig. 8.6. Word recognition results tested on the Aurora-2 database for error protection using convolutional codes (Simulation details are given in Section 8.1.)

8.5 Unequal Error Protection

Performance of media-independent FEC codes, with respect to the level of redundancy inserted, can be enhanced by providing for each bit in the bitstream the adequate level of protection against channel errors. The bits comprising the transmitted datastream of DSR systems do not have equal effect on the word recognition performance of the overall system (Boulis, Ostendorf, Riskin, and Otterson, 2002; Weerackody, Reichl, and Potamianos 2002). For example, in the case of scalar quantization of extracted speech features, the most significant bits (MSBs) generally provide more information for the recognizer than the least significant bits (LSBs). Furthermore, in some source coding algorithms, various speech features affect the system performance differently. For example, in (ETSI 2000), the log-energy and 1st MFCC are considered to play a greater role in speech recognition than higher order MFCCs.

In bandwidth-restrictive systems, it may therefore be beneficial to use *Unequal Error Protection* algorithms to provide more protection for certain bits, while pro-

viding less protection for others. Cyclic codes provide an efficient tool for constructing UEP schemes, due to their flexibility regarding protection rate.

For example, in (Boulis et al. 2002), the authors present an UEP scheme for DSR systems utilizing shortened RS codes. The proposed source coding scheme extracts the first 5 MFCCs, and quantizes the 1st, 2nd, and 4th coefficients using 6 bits, while allocating 4 bits each to the 3rd and 5th coefficient. Also, 20 adjacent windowed frames are packetized, and corresponding bits are grouped across time to form 20 bit symbols. Furthermore, these symbols are allocated to different streams, each of which is protected by a separate RS code. Table 8.5 illustrates the UEP scheme proposed in (Boulis et al. 2002), where integers represent the MFCC corresponding to the given symbol, and *F* denotes an error correction symbol. In this example, streams 1 and 2 are protected with (12,4) RS codes, stream 3 is protected with a (12,6) code, and stream 4 is transmitted without protection.

Table 8.5. Unequal error protection scheme utilizing cyclic codes: integers represent the MFCC corresponding to the given symbol, and F denotes an error correction symbol (from Boulis et al. 2002, © IEEE 2002).

Stream	*Symbols (1-12)*											
1	1	2	3	4	F	F	F	F	F	F	F	F
2	5	1	1	3	F	F	F	F	F	F	F	F
3	5	1	2	2	3	4	F	F	F	F	F	F
4	4	5	4	1	1	4	3	4	2	2	2	2

Convolutional codes can be used for *UEP* schemes to provide various amounts of error protection for bits within a data block. In (Weerackody, Reichl, and Potamianos 2002), the authors propose two different UEP schemes for DSR applications based on convolutional codes. The source coding used in the study involves the extraction of the frame energy and the first 11 cepstral coefficients, either obtained through Mel Filterbank analysis or through Linear Prediction analysis. The source coder allocates 6 bits each to the frame energy and the first 5 cepstral coefficients, while allocating 4 bits each to the remaining coefficients, resulting in a payload bitrate of 4.8 kbps.

The *UEP* algorithms presented in (Weerackody et al. 2002) groups the total 60 bits of each block into 4 different levels of importance. Each level is then encoded into a separate bitstream using a distinct channel coding scheme, and the total bitrate for each case is 9.6 kbps. Let $e^i(n)$ represent the ith bit of the frame energy value at block n, and let $c^i_j(n)$ represent the ith bit of the jth cepstral coefficient for block n. Tables 8.6 illustrates the bit allocation and channel coding scheme involved in the UEP scheme proposed in (Weerackody et al. 2002), referred to as UEP_1. The corresponding results reported for Rayleigh fading channels are provided in Table 8.7.

Table 8.6. Unequal error protection scheme UEP_1 utilizing convolutional codes (from Weerackody et al. 2002, © IEEE 2002)

Level	Speech Feature Bits	Error Protection
1	$e^0(n),e^1(n),c^0_1(n),c^0_2(n),c^0_3(n),c^0_4(n),c^0_5(n)$	rate-1/2 conv. Code
2	$e^2(n),c^1_1(n),c^1_2(n),c^1_3(n),c^1_4(n),c^1_5(n)$	rate-1/2 conv. Code
3	$e^3(n),e^4(n),c^2_1(n),c^3_1(n),c^2_2(n),c^3_2(n),\ldots$ $c^0_6(n),c^1_6(n),c^0_7(n),c^1_7(n),\ldots,c^0_{11}(n),c^1_{11}(n)$	rate-1/2 conv. code + puncturing
4	$e^5(n),c^4_1(n),c^5_1(n),c^4_5(n),c^5_5(n),$ $c^2_6(n),c^3_6(n),c^2_7(n),c^3_7(n),\ldots,c^2_{11}(n),c^3_{11}(n)$	no code

Table 8.7. Word accuracy results for unequal error protection scheme utilizing convolutional codes: "dec. type" refers to the type of data used, i.e. hard-decision decoded data (*Hard*) or soft-decision decoded data (*Soft*) (from Weerackody et al. 2002,© IEEE 2002).

UEP scheme	*dec. type*	*SNR*			
		15 dB	*10 dB*	*7 dB*	*5 dB*
UEP_1	Hard	92.6%	89.8%	82.7%	71.7%
UEP_1	Soft	92.7%	91.3%	87.1%	80.1%

8.6. Frame Interleaving

It is known that for speech recognition purposes, errors that occur in groups are more degrading to word-accuracy performance than errors which occur randomly. Frame interleaving is a technique aimed at countering the effects of such bursty channels, through the addition of delay but no redundancy.

Frame interleaving aims to reorder speech frame packets within the transmitted bitstream so that frames which are adjacent with respect to the original speech signal are separated within the transmitted signal. Thus, grouped errors in the interleaved signal will result in scattered errors in the de-interleaved signal.

Two parameters commonly used to measure the effectiveness of an interleaver are the *delay*, δ, and the *spread*, s. The *delay* of an interleaver is defined as the maximum delay that any packet experiences before being transmitted. That is, for the interleaving function described by $j=\pi(i)$, where j is the original packet position and i is the resulting packet position for transmission, the *delay* can be determined as:

$$\delta = \max_i \left(\pi^{-1}(i) - i\right) \tag{11}$$

An interleaver is said to have a *spread* of s if for any packet positions, p_1 and p_2, within the original ordering:

$$\left|\pi(p_1)-\pi(p_2)\right| < s \quad \Rightarrow \quad \left|p_1 - p_2\right| \geq s. \tag{12}$$

In (James and Milner 2004), three classes of interleavers are discussed for the use of DSR: optimal spread block interleavers, convolutional interleavers, and decorrelated block interleavers. They are successfully shown to provide improved results for DSR systems.

8.6.1 Optimal Spread Block Interleavers

A defining parameter of a block interleaver is the degree d. A block interleaver of degree d operates on a set of d^2 packets by reordering them prior to transmission. A pair of interleaving functions, $\pi_1(\cdot)$ and $\pi_2(\cdot)$, of degree d are considered optimal with respect their spread if, and only if:

$$\pi_1(i \cdot d + j) = (d - 1 - j)d + i, \tag{13}$$

and

$$\pi_2(i \cdot d + j) = j \cdot d + (d - 1 - i), \tag{14}$$

for $0 \leq i, j \leq d-1$. Note that $\pi_1(\cdot)$ and $\pi_2(\cdot)$ are inverse functions, i.e. $\pi_1(\pi_2(i)) = i$. Furthermore, it is shown in (James et al. 2004) that *optimal spread block interleaver* function pairs can be interpreted as 90° clockwise and counter-clockwise rotations of $d \times d$ packet matrices. Figure 8.7 illustrates the d=4 case.

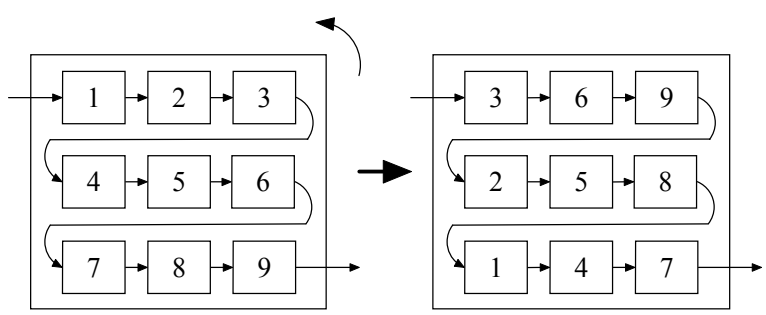

Fig. 8.7. Example of an optimal spread block interleaver with d=3: Note that the interleaving function can be interpreted as a 90° counter-clockwise rotation of the d×d block.

It can also be concluded that for optimal spread block interleavers,

$$\delta_{opt} = d^2 - d, \tag{15}$$

and

$$s_{opt} = d. \tag{16}$$

The performances of optimal spread block interleavers of various degrees are shown in Fig. 8.7.

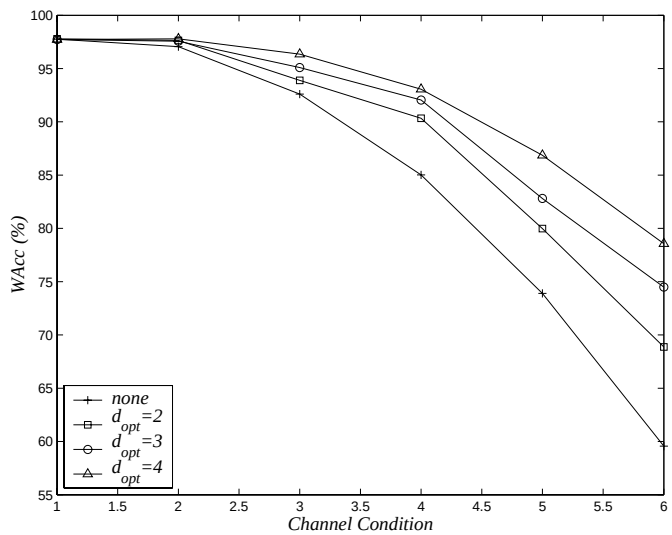

Fig. 8.7. Word recognition results tested on the Aurora-2 database using optimal spread block interleavers and decorrelated block interleavers of various degrees: d_{opt} refers to the degree of the optimal spread interleaver. (Simulation details are given in Section 8.1.)

8.6.2 Convolutional Interleavers

A *convolutional interleaver* can be interpreted as a multirate device operating on a stream of packets. It involves an input multiplexer, followed by a groups of shift registers in parallel, and finally an output multiplexer. A convolutional interleaver of degree d=4 is illustrated in Fig. 8.8.

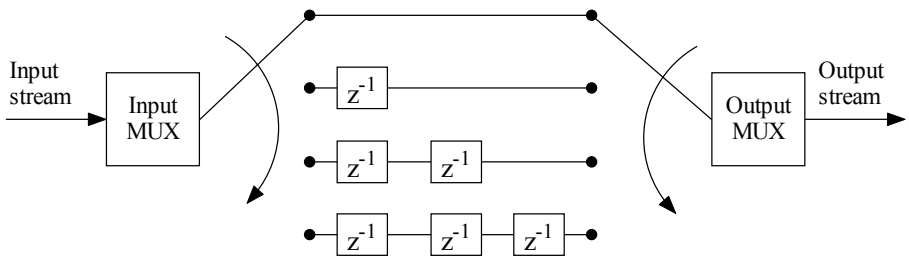

Fig. 8.8. Example of a convolutional interleaver with degree *d*=3.

The interleaving function of the *convolutional interleaver* with degree *d* is given by:

$$\pi_{conv}(i) = i - (i \bmod d). \tag{17}$$

It is also shown in (James et al. 2004) that the maximum delay and spread are given by:

$$\delta_{conv} = d^2 - d, \tag{18}$$

and

$$s_{conv} = d - 1. \tag{19}$$

8.6.3 Decorrelated Block Interleavers

The optimal spread block interleaver and convolutional interleaver previously discussed utilize the maximum delay and spread parameters to measure the effectiveness of specific interleaving functions. However, there exist other parameters by which the performance of an interleaving function can be measured.

For example, in (James et al. 2004), the authors introduce a decorrelation measurement given by:

$$D = \sum_{i=1}^{d^2} \sum_{j=1}^{d^2} \frac{|\pi(i) - \pi(j)|}{|i - j|}. \tag{20}$$

The decorrelation measurement *D* shows the ability of an interleaving function to spread the input feature stream. In order to illustrate this, 2000 interleaving functions with degree *d*=4 were randomly created, and the corresponding decorrelation values were determined. Additionally, the chosen interleavers were tested on a bursty packet-level channel with parameters *p=0.75* and *q=0.75*. Note that the given channel

parameters produce an average burst duration of $\overline{d}_b = 4$ without interleaving. The average burst lengths of the de-interleaved feature streams were then plotted against the corresponding decorrelation values in Fig. 8.10. The final *decorrelated block interleaving function* chosen for each degree is determined through an exhaustive search of randomly created permutations by minimizing *D*.

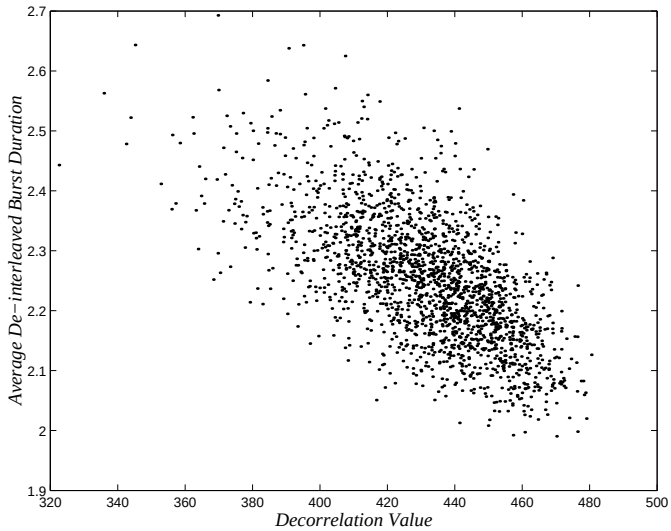

Fig. 8.9. Average burst durations of de-interleaved feature streams as a function of the corresponding decorrelation values.

As can be concluded from Fig. 8.9, there is a strong correlation between the *D* values of interleaver functions and the resulting burst durations.

8.7. Examples of Modern Error Recovery Standards

There exist a number of standards for DSR and speech recognition systems currently in use, for which a main contributer is the *European Telecommunications Standards Institute (ETSI)*. *ETSI* has developed error protection and packetization standards for both low bitrate DSR systems (ETSI 2000), and for full rate NSR systems (ETSI 1998). The channel coding schemes described by these systems are summarized in the following sections.

8.7.1 ETSI DSR Standard (ETSI 2000)

The *ETSI* DSR standard describes a low bitrate speech recognition system intended for DSR applications. The compression algorithm implemented by the given standard extracts the frame energy, as well as the first 13 MFCCs, for every window of speech, at a windowing frequency of 100 Hz. A total of 44 bits is allocated to the extracted speech data through a split vector quantization (SVQ) system, resulting in a data bitrate of 4.4 kbps.

The packetization scheme in (ETSI 2000) includes a multiframe format in order to reduce the overhead required for synchronization and header information. Thus, speech feature data from 24 adjacent speech frames are grouped with one synchronization sequence and one header field.

The information contained in the header field is considered to be critical for decoding purposes, and thus a great deal of error protection is allocated to this data. The header field is protected by an extended (31,16) systematic cyclic code, with the addition of an overall parity check bit. The *cyclic code* has a minimum distance of $d_{\min}$=8, and can therefore support correction of up to 3 channel errors or detection of up to 7 channel errors.

The 24 frames included with the header field and synchronization sequence are grouped into 88 bit pairs. Each pair of frames is then protected with a 4-bit CRC code. The resulting error protection and packetization scheme requires a data rate of 4.8 kbps.

8.7.2 ETSI GSM/EFR Standard (ETSI 1998)

The *ETSI* GSM Enhanced Full Rate (EFR) system, described by (ETSI 1998), is intended for speech transmission, and thus operates at a higher rate than the system described by (ETSI 2000). The GSM/EFR speech coding algorithm compresses 20 ms windows of speech signal into 244-bit blocks of data, which corresponds to a preliminary data rate of 12.2 kbps. The channel coding algorithm presented by (ETSI 1998) is a *UEP* scheme involving *CRC cyclic codes*, *convolutional codes*, and interleaving.

The first step in the channel coding and packetization scheme described for the GSM/EFR system is an expansion of the 244-bit blocks into 260-bit blocks, which results in 16 redundancy bits. Of these 16 redundancy bits, 8 are repetition bits, and 8 correspond to a CRC code.

Each 260-bit expanded data block is then grouped into two classes of bits. The first 50 bits of class 1, referred to as class 1a, are protected with three parity check bits generated with a shortened cyclic code. Both class 1a and class 1b bits are protected with a rate-1/2 convolutional code and interleaved with a bit-level interleaving

function. Class 2 bits are transmitted without protection, and thus the resulting bitrate of the GSM/EFR system after channel coding and packetization becomes 28.4 kbps. The UEP scheme described in (ETSI 1998) is illustrated in Table 8.8.

Table 8.8. UEP Scheme for Channel Coding in ETSI GSM/EFR System (from ETSI 1998)

Class	*original bits*	*channel coded bits*	*error protection*
1a	50	106	cyclic code + rate-1/2 conv. Code
1b	132	264	rate-1/2 conv. Code
2	78	78	no code

8.8. Summary

This chapter focused on error recovery (ER) methods for transmission of speech features over error-prone channels in remote speech recognition applications. Such systems tend to be bandwidth-restrictive and often delay-sensitive, and thus channel coding schemes must be efficient. We discussed both FEC techniques, which add redundancy to the source coded data stream prior to transmission, as well as frame-based interleaving, which creates spread in the data stream prior to transmission to overcome the effect of bursty errors. Table 8.9 summarizes the discussed ER methods.

The individual ER techniques discussed in this chapter, and summarized in Table 8.9, serve as useful tools in providing protection against channel noise or packet erasures within DSR systems. However, channel coding strategies can be enhanced by combining these techniques and offering more error protection to those bits in the bitstream which provide more utility to the overall system, as in UEP schemes. This chapter concludes with example standards of channel coding for modern DSR and speech recognition systems.

Table 8.9. Summary of discussed error recovery techniques

FEC Technique	Error Pattern Application	Advantages	Disadvantages
Media-Specific FEC- coarsely quantized feature replicas are inserted into the datastream with the aim of reconstructing lost packets			
Media-Specific FEC (Section 8.3)	packet erasures	low complexity	introduces delay
Media-Independent FEC- redundancy is added to blocks of data in order to detect or correctly decode corrupted data in the presence of bit-level errors			
Linear Block Codes (Section 8.4.2)	bit errors or additive noise	low delay, low overhead	complexity increases for soft and λ-soft decoding
Cyclic Codes (Section 8.4.3)	bit errors or additive noise	low delay, efficient for decoding bursty channels	restrictive in terms of block length
Convolutional Codes (Section 8.4.4)	bit errors or additive noise	low delay, flexible in terms of block length	complexity increases for soft-decision decoding
Frame Interleaving- frames are interleaved prior to transmission so that bursty packet losses in the transmitted stream may become scattered packet losses in the de-interleaved stream			
Optimal Spread Block Interleavers (Section 8.6.1)	packet erasures	no required additional bandwidth	introduces delay
Convolutional Interleavers (Section 8.6.2)	packet erasures	no required additional bandwidth	introduces delay
Decorrelated Interleavers (Section 8.6.3)	packet erasures	no required additional bandwidth	introduces delay, requires extensive and complex training

Acknowledgements

This work was supported in part by the NSF and by a Radcliffe Institute Fellowship to Abeer Alwan.

References

K. Andrews, C. Heegard, and D. Kozen (1997) A Theory of Interleavers. Technical Report 97-1634, Computer Science Department, Cornell University, June.

H. Bai, and M. Atiquzzaman (2003) Error Modeling Schemes for Fading Channels in Wireless Communications: A Survey. IEEE Communications Surveys and Tutorials, Fourth Quarter, Volume 5, No. 2, pp. 2-9.

A. Bernard, and A. Alwan (2001) Joint Channel Decoding- Viterbi Recognition for Wireless Applications. Europspeech, vol.3, pp. 2703-2706.

A. Bernard and A. Alwan (2002a) Low-Bitrate Distributed Speech Recognition for Packet-Based and Wireless Communication. IEEE Transactions on Speech and Audio Processing, Vol. 10, No. 8, pp. 570-579.

A. Bernard, X. Liu, R. Wesel and A. Alwan (2002b) Speech Transmission Using Rate-Compatible Trellis Codes and Embedded Source Coding, IEEE Transactions on Communications, vol.50, no.2, pp. 309-320.

A. Bernard, (2002) Source and Channel Coding for Speech Transmission and Remote Speech Recognition, PhD Thesis, University of California, Los Angeles.

R. E. Blahut (2004) Algebraic Codes for Data Transmission. Cambridge.

J.-C. Bolot (2003) End-to-End Packet Delay and Loss Behavior in the Internet. ACM Sigcomm, pp. 289-298.

C. Boulis, M. Ostendorf, E. A. Riskin, and S. Otterson (2002) Graceful Degradation of Speech Recognition Performance Over Packet-Erasure Networks. IEEE Trans. Speech and Audio Processing. Vol. 10, No.8, pp. 580-590.

R. O. Duda, P. E. Hart, and D. G. Stork. (2000) *Pattern Classification*. Wiley-Interscience.

E. Elliot (1963) Estimates of Error Rates For Codes on Bursty Noise Channels. Bell Systems Technical Journal 42(9).

ETSI EN 300 909 v7.3.1, 1998, (1998) Digital Cellular Telecommunications System (Phase 2+)(GSM); Channel Coding.

ETSI ES 201 108 v1.1.2, 2000 (2000) Distributed Speech Recognition; Front-end Feature Extraction Algorithm; Compression Algorithms.

R. G. Gallager (2001) *Discrete Stochastic Processes*. Springer.

K. J. Han, N. Srinivasamurthy, and S. Narayanan (2004) Robust Speech Recognition over Packet Networks: An Overview. Proc. of ICSLP, vol. 3, pp. 1791-1794.

S. Haykin (2000) *Communication Systems*. Wiley.

H. G. Hirsch and D. Pearce (2000) The AURORA Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Condition. ISCA ITRW ASR 2000.

W.-H. Hsu and L.-S. Lee (2004) Efficient and Robust Distributed Speech Recognition (DSR) over Wireless Fading Channels: 2D-DCT Compression, Iterative Bit Allocation, Short BCH Code and Interleaving, Proc. of ICASSP, Vol. 1, pp. 69-72.

A. B. James and B. P. Milner (2004) An Analysis of Interleavers for Robust Speech Recognition in Burst-Like Packet Loss. Proc. of ICASSP. Vol. 1, pp. 853-856.

C. Jiao, L. Schwiebert, and B. Xu (2002) On Modeling the Packet Error Statistics in Bursty Channels. IEEE LCN, pp. 534-538.

A.M. Kondoz (2004) Digital Speech: Coding for Low Bitrate Communication Systems. John Wiley and Sons.

J. Kurose, and K. Rose (2003) Computer Network: a Top-Down Approach Featuring the Internet. Addison-Wesley.

A.Leon-Garcia (2007) Probability and Random Processes for Electrical Engineers. Prentice Hall.

G. T. Nguyen (1996) A Trace-Based Approach for Modeling Wireless Channel Behavior. Proc. Winter Simulation Conf. pp. 597-604.

A. M. Peinado, A. M. Gomez, V. Sanchez, J. L. Perez-Cordoba, and A. J. Rubio (2005a) Packet Loss Concealment Based on VQ Replicas and MMSE Estimation Applied to Distributed Speech Recognition. Proc. of ICASSP. Vol. 1, pp. 329-330.

A. M. Peindao, V. Sanchez, J. L. Perez-Cordoba, and A. J. Rubio (2005b) Efficient MMSE-Based Channel Error Mitigation Techniques. Application to Distributed Speech Recognition Over Wireless Channels. IEEE Trans. on Wireless Communications. Vol. 4, No. 1, pp.14-19.

A. M. Peindao, and J. C. Segura (2006) *Speech Recognition Over Digital Channels*. Wiley.

J. Proakis (1995) *Digital Communications*. New York: Mcgraw-Hill.

L. Rabiner, B.-H. Juang (1993) *Fundamentals of Speech Recognition*. Prentice Hall.

B. Sklar (1997) Rayleigh Fading Channels in Mobile Digital Communication Systems Part 1: Characterization. IEEE Communications Magazine, pp. 90-100.

Z.-H. Tan, P. Dalsgaard, and B. Lindberg (2005) Automatic Speech Recognition over Error-Prone Wireless Networks. Speech Communication. 47, pp. 220-242.

V. Weerackody, W. Reichl, and A. Potamianos (2002) An Error-Protected Speech Recognition System for Wireless Communications. IEEE Trans. on Wireless Communications, Vol. 1, No. 2, pp. 282-291.

A. Viterbi (1971) Convolutional Codes and Their Performance in Communication Systems. IEEE Trans. on Communications, vol. 19, issue 5, part 1, pp. 751-772.

R. D.Wesel (2003) Convolutional Codes, from *Encyclopedia of Telecommunications*, Wiley.

S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, P. Woodland. (2000) *The HTK Book*. Microsoft Corporation.

Index Words