Channel Coding for Enhanced Full Rate GSM

by

Christine N. Liu

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the Massachusetts Institute of Technology

September 1996

Author_____
Department of Electrical Engineering and Computer Science
August 26, 1996

Certified by_____
Anantha P. Chandrakasan
Thesis Supervisor

Accepted by_____
F.R. Morgenthaler
Chairman, Department Committee on Graduate Theses

Channel Coding for Enhanced Full Rate GSM
by
Christine N. Liu

Submitted to the
Department of Electrical Engineering and Computer Science

August 26, 1996

in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

Maintaining good speech quality in cellular phones is difficult, even with effective error protection (channel coding) schemes, because of the noisy, fading transmission channel. As the popularity of cellular phones increases, however, so does the demand for better-sounding speech. Recent advances in speech compression have made coders more robust to the cellular environment. This paper explains the design of a channel coding scheme for a new speech coder intended for use in European digital cellular systems. The scheme is similar to that used in the existing system, with a modification in the decoder. With channel coding, the new speech coder out-performs the existing system in standard listening tests.

Thesis supervisor: Anantha P. Chandrakasan
Title: Assistant Professor

Acknowledgments

Many thanks to everyone in WCS and Speech Research Groups at Texas Instruments for a memorable six-month stay. Special thanks to Wilf LeBlanc, Vishu Viswanathan, and John Crockett. This project would not have been possible without your advice and support. Vishu and Wilf, thanks for having confidence in me, and for your patient technical explanations. John, you're the best supervisor ever.

Thanks to Professor Chandrakasan for valuable comments, and to Edward Slottow and Jason Sachs for word processing tutorials. I am indebted to my friends Jason, Teddy, Andrew Steele, Alex Chen, and Mike Richters. Thanks for stress relief, motivation, and overall caring. It was especially needed this year. Finally, to my family, who had to put up with me for another six months, much love and happiness.

Contents

List of Figures

List of Tables

Chapter 1

Digital Cellular Phone Systems

1.1 Introduction

Cellular phones have become popular in recent years. To handle the increasing demand, many providers are switching to digital technology. However, the speech quality of digital cellular systems does not come close to traditional wireline phones. The problems are due to the low bit rate, combined with a noisy transmission environment. In current systems, channel coding helps to correct transmission errors and improve the speech quality.

Recent advances in speech compression technology have also improved speech quality. This paper presents the design of a channel coding scheme for a new speech coder intended for use in Europe. The first chapters give a general background. Chapter 1 introduces cellular phone systems. Chapter 2 explains the current European standard, Groupe Speciale Mobile (GSM). Chapter 3 describes the cellular transmission environment. Chapter 4 reviews the fundamentals of channel coding relevant to this study, including convolutional codes and parity checking. The later chapters present the specifics of this design. Chapter 5 discusses the reasons behind the design, and Chapter 6 presents the final test results.

1.2 Physical structure

Cellular systems are composed of mobile stations, base stations, and a central switching office (Figure 1).[13][18][20] The mobile station is the cellular phone, which can move as quickly as a car or as slowly as a pedestrian. The mobile station communicates with an immovable base station at frequencies around 800MHz-900MHz. Each base

station handles all mobile stations within a limited geographical area. The base station in turn communicates with the central switching office, which controls the central switching. The central switching office is connected to the ordinary telephone lines.

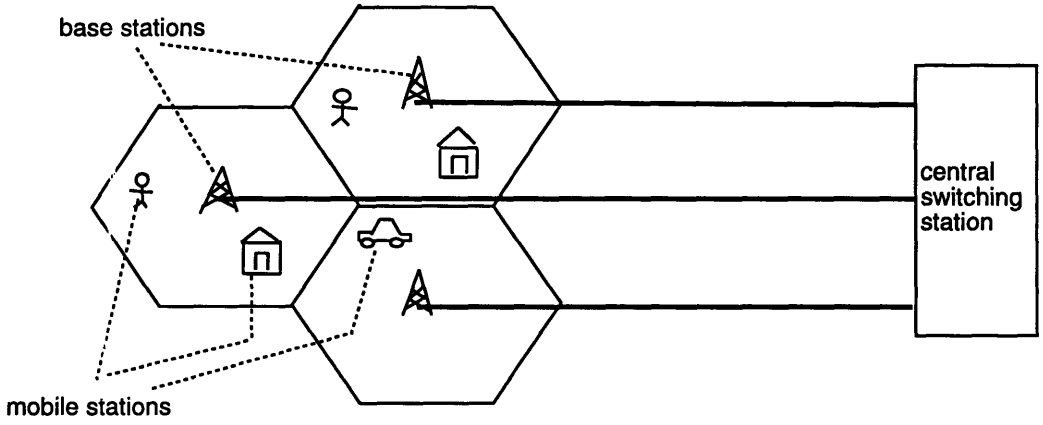

Figure 1: Physical structure of a cellular phone system.

The area that a single base station services is called a cell. In many theoretical drawings, the cells are hexagonal; in principle they should cover the most area without overlapping. In reality, cells are amorphous and may overlap one another (Figure 2).[13] A cell radius can range anywhere from 30km to 1km. Recent work has also involved microcells with radii of merely 100m.[20]

1.3 Making a call

When a user makes a call, the mobile station requests a calling channel from the nearest base station. The base station links to the central switching office and sets up a traffic channel and control center, allowing the mobile station to send and receive voice data. When the call is finished, the mobile station relinquishes control of the traffic channel. If a user in the middle of a call moves out of the range of the base station, control is given to the base station of the new cell. This procedure is called handoff.[18][20]

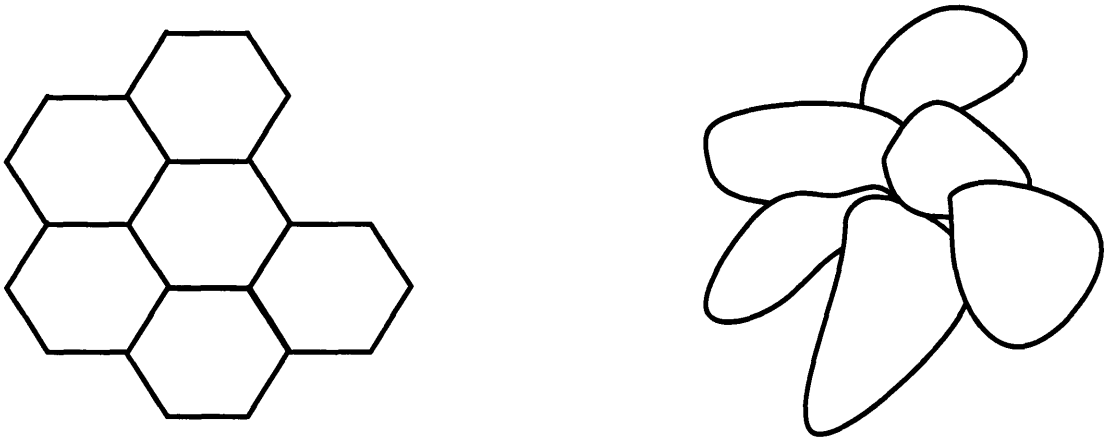

Figure 2: Theoretical vs. actual cell shapes.

On the other hand, for a call to reach the mobile station, the base station sends a paging signal containing the mobile station's identifying code. The mobile station must be in stand-by mode to receive the paging signal. When the mobile station replies, a traffic channel is set up.

1.4 Ways to increase capacity

As the popularity of cellular phones increases, so does the need to allow more users on the system. Cellular systems increase their user capacity over that of traditional radio through two methods: frequency re-use and multiple access.

1.4.1 Frequency re-use

In frequency re-use, two cells operate on the same frequency band.[13][18] The bandwidth available to the system is divided into sections. A base station is assigned a certain set of frequencies. The base stations surrounding it are assigned different frequency sets, while a base station several cells away is assigned the same frequency section (Figure

3). This causes interference between the two cells on the same frequency, but the noise is manageable if the cells are far enough apart. The requisite distance depends on such factors as terrain and antenna design.

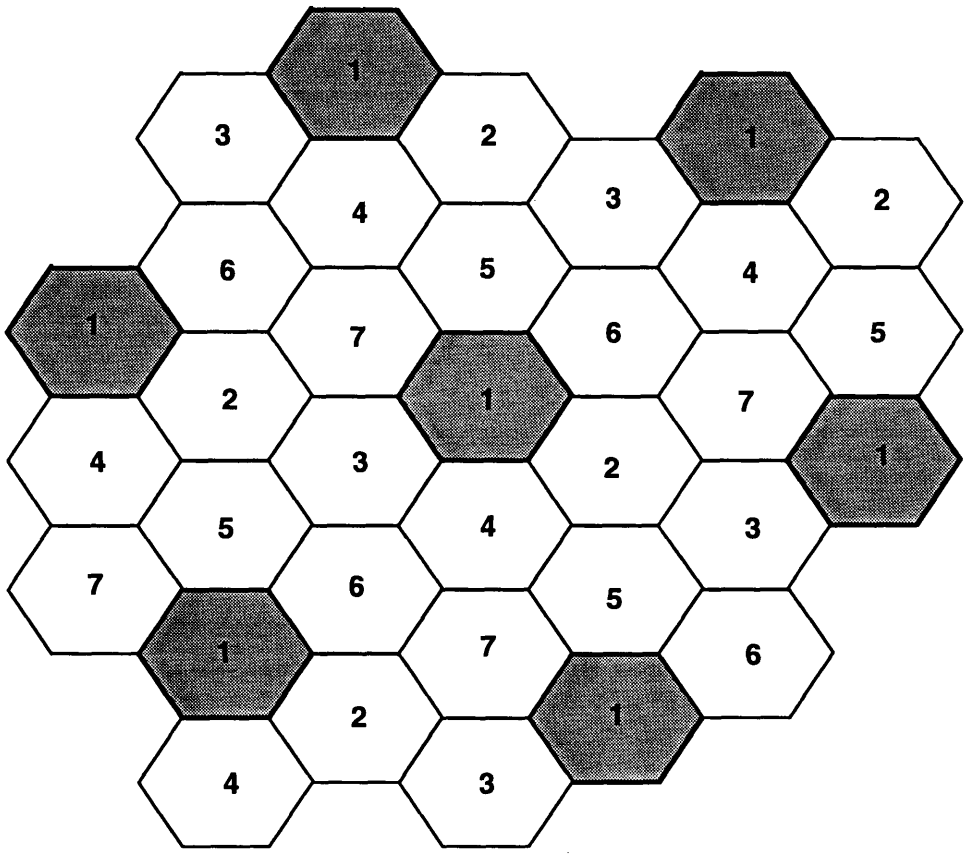

Figure 3: Frequency re-use plan for seven sets of frequencies; shaded cells all use set 1.[13]

1.4.2 Multiple access

Another way cellular systems increase capacity is with multiple access schemes, which allow many users in one bandwidth.[18][19][20] Analog cellular systems use a scheme called Frequency-Division Multiple Access (FDMA). Each base station splits the allocated bandwidth into smaller frequency bands and puts one user on each band.

Digital cellular systems use Time-Division Multiple Access (TDMA), in which there are several users per frequency band. Data from each user is partitioned into blocks

and sent over the channel staggered in time. If there are n users, every nth block sent will belong to the same user. TDMA systems have high transmission rates compared with FDMA, and designers must worry about synchronization between mobile station and base station.

Recently, spread spectrum techniques, also known as code-division multiple access (CDMA), have become a popular topic. In direct-sequence (DS-CDMA), each user transmits over the entire frequency band. One user distinguishes itself from other users by unique codes. In frequency-hopping (FH-CDMA), there is only one user per frequency, but a user moves from one frequency to another. The advantages of frequency hopping are described in Chapter 3.

Chapter 2

GSM System

Groupe Speciale Mobile (GSM) was adopted in 1987 as the digital cellular phone standard in Europe.[18][20] It operates at 900 MHz and has cells up to 35 km in radius. Figure 4 shows the steps involved in transmitting speech data. Control data is processed similarly.

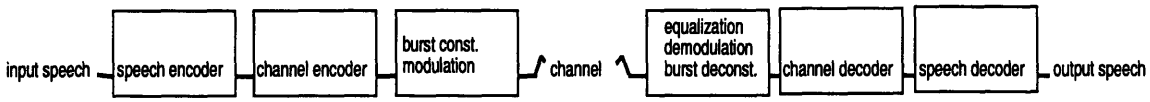

Figure 4: GSM speech transmission path.

2.1 Speech Coder

In the first block, the speech coder compresses digitized speech.[20][22][20] Speech is produced when air passes over the vocal chords and is shaped by the mouth and lips. If the vocal chords vibrate, voiced sounds such as a and o are produced. The frequency of the vibration is the pitch. If the vocal chords do not vibrate, unvoiced sounds such as s and f are produced. Speech can be modeled with an all-pole filter representing the vocal tract (Figure 5). The filter coefficients are often found with linear prediction techniques. The filter is excited by a signal containing pitch and voiced/unvoiced information. The types of excitation signals are widely varied. Common structures include codebook, pulse, and mixed excitation.

A vocoder is a speech coder that uses a speech model for compression. A vocoder computes the parameters for the filter and the excitation signal. These parameters, rather than the actual speech waveform, are sent to the receiver, where the decoder reconstructs the speech. Thus, the decoded speech is not the original speech waveform, but a modeled

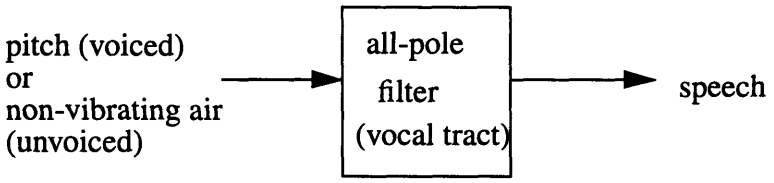

Figure 5: Model of speech production.

representation. Vocoders are commonly used in military applications because they produce intelligible speech at low bit rates (2 to 4 kilobits per second).[20] Recently, they have also been used in commercial standards such as GSM.

The GSM speech coder is a pulse-excited vocoder called RPE-LTP.[4][20] It takes 160 samples (20 ms) of speech at a time and outputs 260 bits. This gives it an output bit rate of 13 kilobits per second (kbps). The encoder (Figure 6) consists of two analysis filters, followed by an excitation analyzer. The encoder first puts the speech samples through a pre-processor, which removes offset and shapes the signal. It then computes filter coefficients (reflection coefficients) to model the vocal tract. The process, which is done by linear prediction, is called STP analysis. With these coefficients, the encoder filters the speech samples to get the short-term residual signal. This residual is split into 4 sub-frames of 40 samples each.

Long-term prediction (LTP) parameters, consisting of lag and gain, are computed for each sub-frame. The parameters model the pitch periodicity in the signal. They are based on the present sub-frame, plus the estimated short-term residual of the three previous sub-frames (a total of 160 samples). The LTP parameters are used to obtain the long-term residual signal.

In the final section of the encoder, the long-term residual is represented as one of four possible sequences. Each sequence has 13 regularly-spaced pulses of varying ampli-

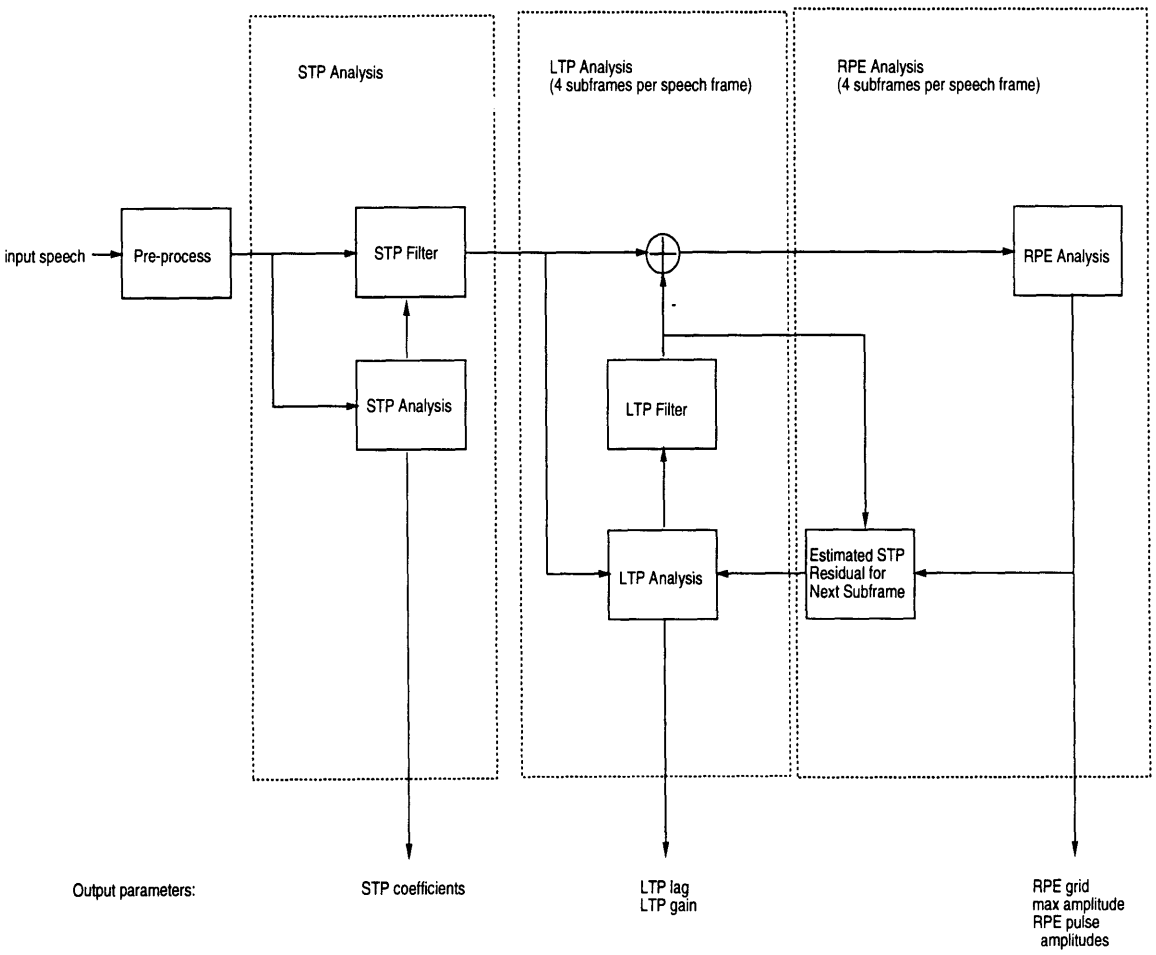

Figure 6: RPE-LTP speech encoder.

tudes. The choice of sequence is called the grid position. The pulse amplitudes and maximum sub-frame amplitude are also outputted. The pulse sequence and previously computed short-term residual estimates are used to reconstruct the short-term residual signal. This is saved for LTP analysis of the next sub-frame.

Table 1 shows the bit allocations of the parameters. Of the 260 output bits, 36 bits are for the short-term filter coefficients, 8 bits for the LTP gains, 28 bits for the LTP lags, 8 bits for the RPE grid positions, 24 bits for the sub-frame maximum amplitudes, and 156 bits for the pulse amplitudes.

Parameter	Number of bits
8 STP filter coefficients	36
Each of 4 sub-frames:	
LTP gain	2 (x 4 = 8)
LTP lag	7 (x 4 = 28)
RPE grid	2 (x 4 = 8)
max amplitude	6 (x 4 = 24)
13 pulse amplitudes	39 (x 4 = 156)
Total	260

Table 1: Bit allocation of GSM speech coder.

To reconstruct the speech, the GSM speech decoder (Figure 7) creates the excitation signal and passes it through the two filters.

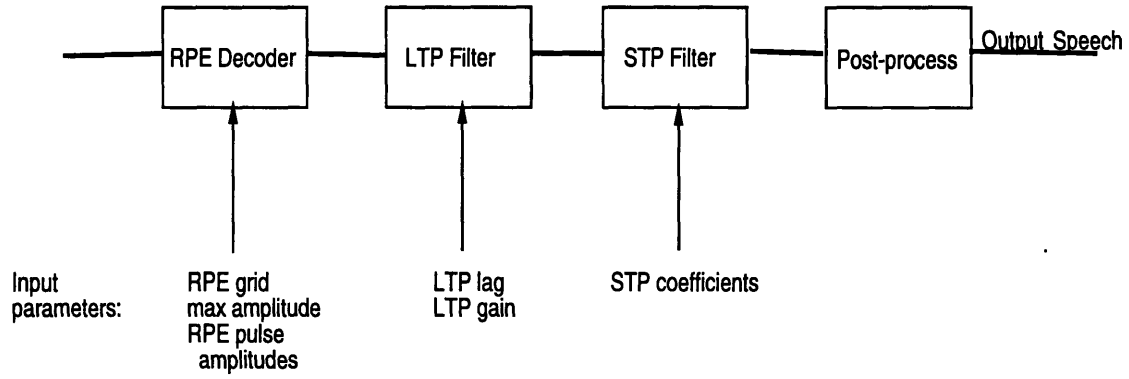

Figure 7: RPE-LTP speech decoder.

2.2 Channel coding and interleaving

Next, channel coding adds error protection bits to the data, bringing the total bit rate to 22.8 kbps. GSM channel coding is discussed in more detail in Chapter 4. Interleaving, another error protection measure, is also discussed.

2.3 Burst construction and modulation

After channel coding, the bits are arranged into eight bursts, each 0.58 ms long. The bursts will be placed in separate TDMA frames for transmission. Thus, it takes 8 TDMA frames for all the data from one speech frame to be sent over the channel. The bursts are modulated with Gaussian Minimum Shift Keying (GMSK), in which the data is processed through Gaussian filter and then a voltage-controlled oscillator (VCO).[20]

2.4 Channel

The modulated signal is then sent across the channel at a rate of 270.8 kbps. The GSM system uses the frequency bands 935-960 MHz for forward (base station to mobile station) transmission, and 890-915 MHz for reverse (mobile station to base station) transmission. These bands contain many carrier frequencies. Each carrier has a bandwidth of 200 kHz and maximum capacity of 8 users.[20]

2.5 Receiver

The receiver reverses the transmitter process. It demodulates the signal, disassembles bursts, deinterleaves, channel decodes, source decodes, and converts the bits back into speech. One key addition is the equalizer in the demodulation block. The equalizer, based on a Viterbi algorithm, reduces interference between successive data bits.

Chapter 3

Mobile Radio Environment

When compared with wireline phones, the GSM standard -- indeed, all cellular phones --has bad sounding speech and high noise levels. Another common complaint is "dropped" calls, when the phones hang up in the middle of a call. These problems are due to noise on the channel. The cellular channel can be modeled in a number of ways.

3.1 Gaussian channel

The simplest channel is a Gaussian, or additive white Gaussian noise (AWGN), channel.[13][20] It is an ideal channel in that no noise comes from the channel, only from the receiver. It occurs when the signal travels between base station and mobile station on a single path, as might happen in microcells. The amplitude of the noise has a normal distribution and is constant over the entire bandwidth. An important consideration is path loss, the attenuation of signal strength over distance. In general, the cellular environment has an attenuation of 40dB/decade.[13]

3.2 Fading channel

A more realistic channel model is a fading channel, also called a multipath interference channel.[9][13][18][20] Fading occurs if the signal travels on more than one path. This can happen when the signal bounces off buildings in an urban area, for instance. The signals along the multiple paths differ in delay time, phase, and doppler shift. They interfere with each other and create a standing wave pattern. A mobile station that is stationary in the pattern will receive a constant signal, which can be modeled as Gaussian.[13] However, for a mobile station that moves through the pattern, as is more often the case, the

received signal strength changes over time (Figure 8). When the signal strength is low, the mobile station is said to be in a fade. The frequency of fades depends on the speed of the mobile station. The higher the speed, the more fades per second. The depth of the fades depends on the standing wave pattern. [13] The two main fading patterns are Rician and Rayleigh fading.

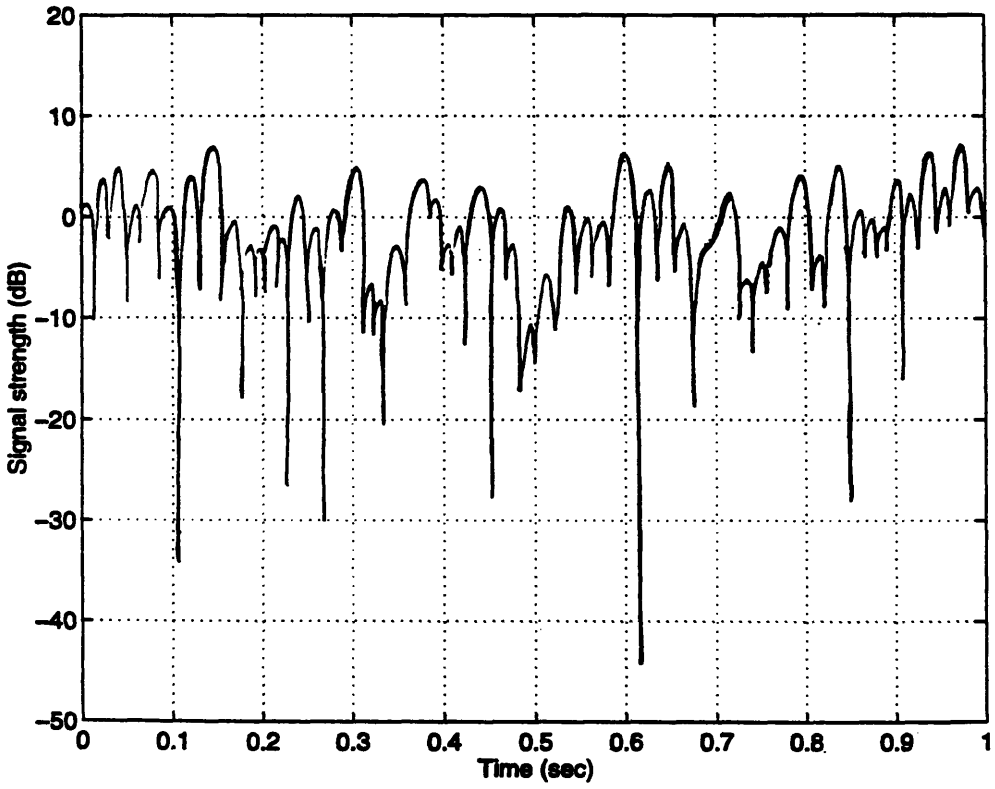

Figure 8: Fading signal.

3.2.1 Rician fading

When a line-of-sight path exists between mobile station and base station, that path will dominate. The condition where one path is much stronger than any other results in Rician fading. At any particular moment, the signal strength has a Rician distribution:

$$\rho_{rice}(a) = \frac{a}{\sigma^2} e^{\frac{-a^2}{2\sigma^2}} e^{-K} I_o\left\langle \frac{a}{\sigma}\sqrt{2K}\right\rangle \quad (1)$$

where I_0 is the zeroth order Bessel function, and σ^2 is the variance. The depth of the fades depends on $K = \frac{power-in-dominant-path}{power-in-other-paths}$. The smaller K, the higher the probability of deep fades.

3.2.2 Rayleigh fading

In the extreme of $K = 0$, all signal paths are equally important. This situation is called Rayleigh fading and is the most severe. The signal strength has a Rayleigh distribution:

$$\rho_{rayleigh}(a) = \frac{a}{\sigma^2} e^{\frac{-a}{2\sigma^2}} \quad (2)$$

Some work has been done to simulate Rayleigh fading channels from Gaussian sources.[2][9]

3.2.3 Frequency-selective fading

Often, fading does not occur equally across all frequencies. Some frequencies may have normal power levels at the same time that other freqeuncies are in a deep fade. This condition is called frequency-selective fading.[13][18] It is the result of differences in delay times of the multiple signal paths. The closer frequencies are to each other, the more

correlated their fading. Many systems employ FH-CDMA to reduce the probability of long fades. The signal is transmitted on a certain set of frequencies for a while, then hops to a different set of frequencies. This way, even if one set of frequencies is in a fade the signal will not be in the fade for long. The GSM standard, for example, calls for pseudo-random frequency hopping at 217 hops per second.[20]

3.3 Noise in the cellular environment

Noise is any non-signal energy in the bandwidth of a channel. Although cellular bandwidths are narrow, they nevertheless have many sources of noise, due to frequency reuse. That is, signals in one cell will encounter interference from the many other cells using the same frequency. This co-channel interference is the most significant noise problem in the cellular environment.[13][18][20] The extent of the interference is measured by carrier-to-interference (C/I) ratio.[13] A C/I of 13 dB is considered a clean channel, while 3 dB is quite noisy.

$$(C/I) = \frac{signal-level-of-desired-cell}{\sum signal-levels-from-interfering-cells} \quad (3)$$

The problem of co-channel interference, as well as other forms of noise, is intensified by the fading characteristics of cellular channels. During fades, the signal power is low, so that any noise becomes more noticeable. The C/I ratio can decrease considerably, making co-channel interference particularly troublesome. In addition, fading is time-varying; there are periods of high signal power and low signal power. This gives rise to time-varying noise characteristics. The periods of high noise create many errors in the signal, while the periods of low noise create fewer errors. Thus, the errors tend to occur in clus-

ters, which are called burst errors. The next chapter discusses ways to combat random bit errors and burst errors.

Chapter 4

Error Protection

Many methods are available to reduce the effects of co-channel interference. Some, such as cell management or antenna size, depend on the physical design of the system; while others involve manipulation of data bits. The latter is called channel coding.[1][14][23] The goal of channel coding is to add redundant bits to the data so that channel errors will have the least effect on the decoded data bits. Two channel coding algorithms will be discussed here, along with interleaving.

4.1 Cyclic redundancy check codes

A simple channel code that detects errors is a cyclic redundancy check code (CRC).[17] A CRC (Figure 9) takes blocks of data bits (D1) as input. For each block, a small number of parity bits are computed (P1). The data block and parity bits are sent over the channel, where both can be hit by errors. At the decoder, a new set of parity bits (P2) is computed from the received data block (D1'). If P2 is different from the received parity bits (P1'), a channel error probably occurred in the data bits and appropriate steps can be taken to mask its effect. The steps may consist of retransmitting the data, as in automatic-repeat-request (ARQ) schemes, or using data from a previous block.

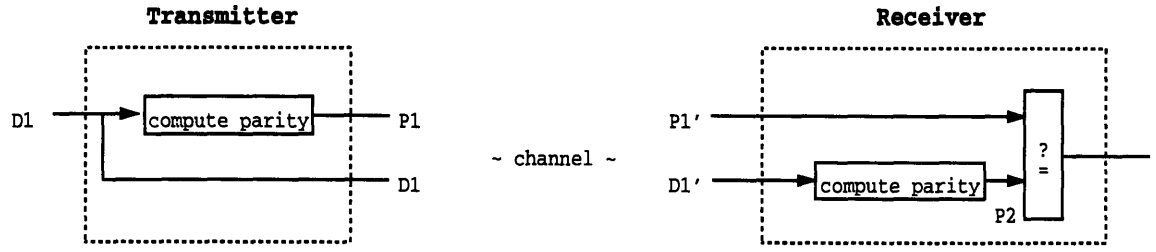

Figure 9: CRC code.

Figure 10 shows how to calculate the parity bits. D1 can be viewed as a polynomial over the integers modulo 2. The polynomial is of order n, where n is the number of bits in the block, and the data are the coefficients of the polynomial. D1 = 100010 would correspond to $x^5 + x$. D1 is divided by a generator polynomial, G. G is primitive, meaning its only factors are itself and 1, much like a prime number. The remainder from the division is the parity bits.

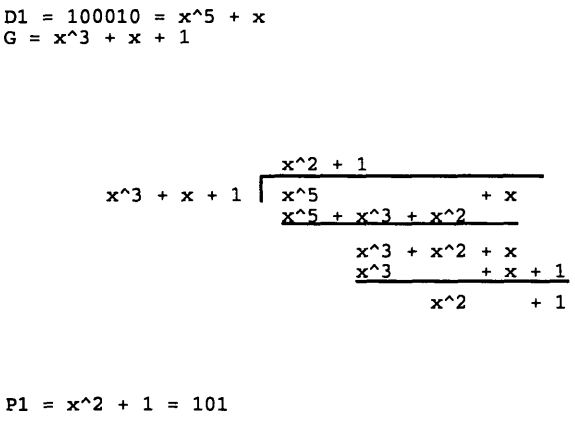

Figure 10: Parity calculation using generator polynomial $x^3 + x + 1$.

Suppose $G = x^3 + x + 1$. For an input of D1 = 100010, P1 = 101. D1 and P1 are sent over the channel. If no errors occur, P2 will be the same as P1', and D1' will be marked correct. On the other hand, if the data has errors, P2 will not equal P1'. For example, if D1' = 100000, P2 = 111. The errors will then be detected.

The CRC code can fail under two conditions. Different data blocks can result in the same parity bits. For instance, the data block D2 = 101001, when divided by $x^3 + x + 1$, will also give a parity of 101. If the channel errors are such that D1' = D2, P2 will equal P1', and the errors will be undetected. The other case where CRCs do not work is if errors hit the parity bits but not the data bits. P2 will not equal P1' because of errors in P1'. The

CRC will mark the data as incorrect, although there are no errors in the data. These two conditions are important tradeoffs to consider in the design of a CRC code. The more parity bits in the design, the less likely two data blocks will give the same parity. However, having more parity bits means the parity is more likely to be hit with channel errors.

4.2 Convolutional codes

4.2.1 Shift register encoding

A popular channel code that corrects errors is a convolutional code.[3][23] Convolutional codes view input bits as streams of data. To encode, the data stream is passed through a shift register (Figure 11). Generator polynomials add together elements of the shift register to produce output bits. In this example, an input of d = 1011000 would result in an output of 11101001110100. For ease of decoding, the shift register usually starts and ends with all memory elements being zero. This requires extra zeros (tail bits) to be added to the end of the data stream.

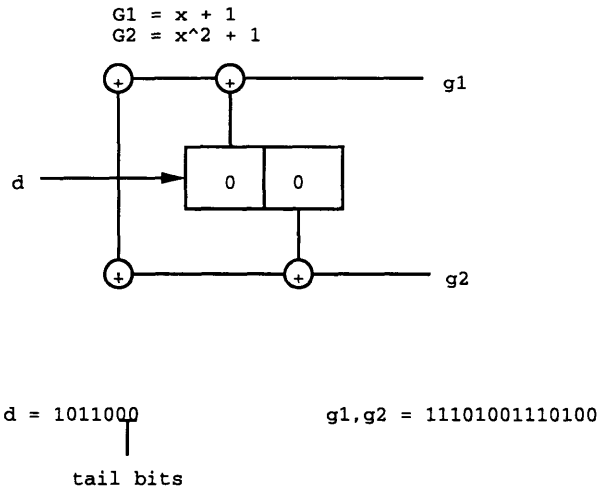

Figure 11: Convolutional code encoder.

The two characteristics of a convolutional code are rate and constraint length. The number of generator polynomials determines the rate of the code, the ratio of input bits to output bits. In Figure 11, the code is a rate 1/2 code because for every 1 input bit, 2 output bits are generated. The constraint length of a code is the number of memory elements in the shift register. This code has a constraint length of 2. In general, the longer the constraint length and the higher the rate, the better the code.

4.2.2 Trellis representation

Convolutional codes can be represented as a trellis diagram (Figure 12). The states of the trellis are the memory states of the shift register. N paths are possible to and from each state, where N is the number of possible input values. In Figure 12, the code has 2 possible input values, 0 and 1. Thus, two exit branches are possible, each giving a different state change and output bits. The input bit determines which branch to take. For example, if the current state is 1, an input of 1 would result in an output of 10 and a change to state 3, while an input of 0 would result in an output of 01 and a change to state 2. An input bit increments the trellis stage, forming a path of states through the trellis. This path is unique for each input stream. In Figure 12, the highlighted path corresponds to an input of d = 1011000.

4.2.3 Viterbi decoding

Decoding involves finding the correct path through the trellis. The most popular trellis search method is the Viterbi algorithm.[3][16] The algorithm goes through the trellis stages and, based on the received data bits, finds the most likely branch to enter each state. The branch is then saved for each state. At the last stage, the algorithm traces the saved branches back to the first stage, illuminating the best path. Any of the states in the

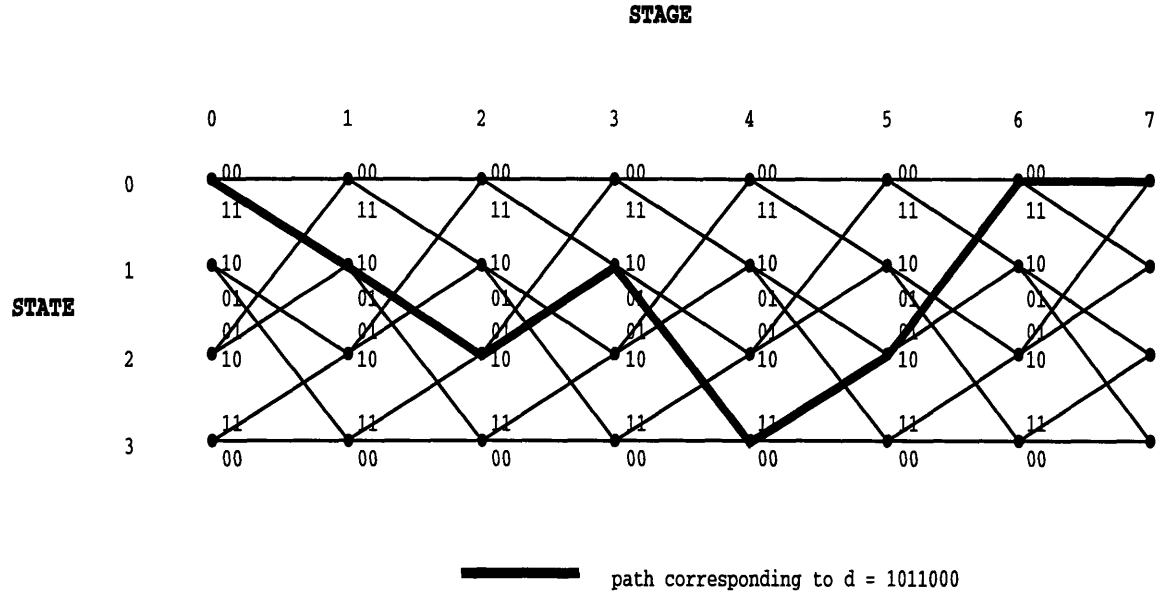

Figure 12: Convolutional code trellis diagram.

last stage could be a starting point for the trace-back process. However, the best performance is obtained if the starting point is known, i.e., by clearing the encoder shift register after all the data bits have been coded.

The most likely branch to enter each state is determined by comparing the path metrics that the possible branches would give. The path metric is related to the likelihood of being in the state. Figure 13 shows the two possible branches (A and B) entering state 1 in stage Y of the trellis in Figure 12. Branch A exits from state 0 and has output bits a1 = 1, a2 = 1. Branch B exits from state 2 and has output bits b1 = 1, b2 = 0. The output bits are compared to the data bits that were transmitted over the channel from stage Y - 1 in the encoder. The closer the bits, the more likely the branch was taken. A common distance measure is Hamming distance, which counts the number of differing bits. For example, if the received data bits were y1 = 1 and y2 = 0, the Hamming distance of branch A is 1 and the Hamming distance of branch B is 0.

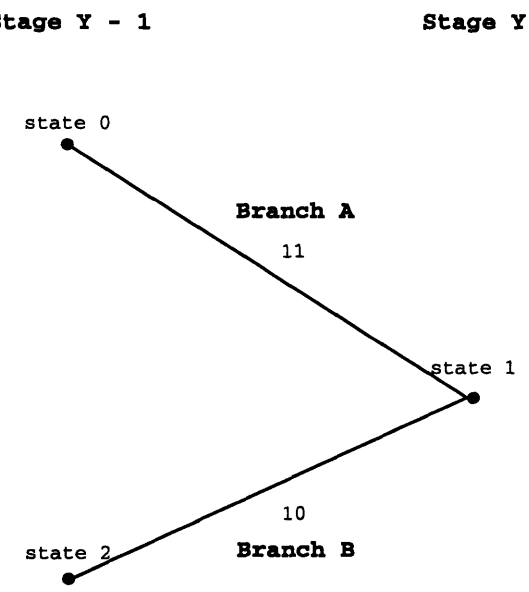

Figure 13: Branches entering state 1 at Stage Y of trellis.

The path metric, then, is the cumulative Hamming distance of the most likely branches from stage 0 to state S in stage Y (in this case S = 1). To find the path metric of a state, candidate path metrics are computed for each entering branch by adding the Hamming distance of the branch to the path metric of the branch's previous state. The branch with the best path metric is then chosen. For example, if the path metric for state 0 in stage Y - 1 is 12, branch A would give a candidate path metric of 13. If the path metric for state 2 in stage Y - 1 is 20, branch B would give a candidate path metric of 20. Thus, the path metric of this state is 13, and the most likely branch is branch A. The Viterbi algorithm calculates the path metric and the most likely branch for every state in the trellis. To ensure that all decoded paths will trace back to the correct starting state, the algorithm initializes the path metrics at stage 0, so that the starting state (usually state 0) has a large negative metric.

Other distance measures besides Hamming distance can also be used. A popular one is Euclidean distance:

$$d = \sum_{J} (y_j - a_j)^2 \tag{4}$$

where J is the number of output bits per branch, y_j is a received data bit, and a_j is the corresponding output bit on branch A. With Euclidean distance, the Viterbi algorithm is called soft-decision because each received bit can take on many values. (With binary distance measures such as Hamming distance, the algorithm is hard-decision.) The Viterbi algorithm is a maximum-likelihood method if Euclidean distance is used.[16]

4.2.4 Punctured codes

A code rate of 2/3 means that 2 bits are input to the shift register each time. But this means that each state in the trellis has 4 possible branches, rather than 2. This increases decoding complexity considerably. One way to solve the problem is with punctured codes.[8][12][23] That is, use a code rate with only 1 input bit per stage, such as rate 1/2 or 1/3, and discard a set pattern of the output bits. The discard pattern is called the puncturing pattern, and the number of input bits to realize it is the puncturing period. For example, if every fourth output bit of a rate 1/2 code is discarded, the puncturing period is 2 and the puncturing pattern is 1110. Thus, for 2 input bits, only 3 bits are transmitted, giving a code rate of 2/3. At the decoder, the discarded bits are inserted as erasures, and the Viterbi algorithm proceeds as if the code rate were 1/2.

4.2.5 Error correction

The path metrics in Figure 14 illustrate how the Viterbi algorithm corrects errors. For simplicity, suppose the all-zero path is sent over the channel. At any trellis stage, the decoder will choose a branch that diverges from the all-zero path only if the path metric of the incorrect branch is lower. For example, if one bit error occurs in the seventh received bit, the key decision is the choice of branches at state 0 of stage 4. Using the code of the

previous example and assuming an initial path metric of -20 at state 0, the path metric of the top branch is -19, while the path metric of the bottom branch is -15. Thus, the correct branch will be chosen, despite the bit error.

input: 000000
transmitted: 000000000000
received: 000000100000

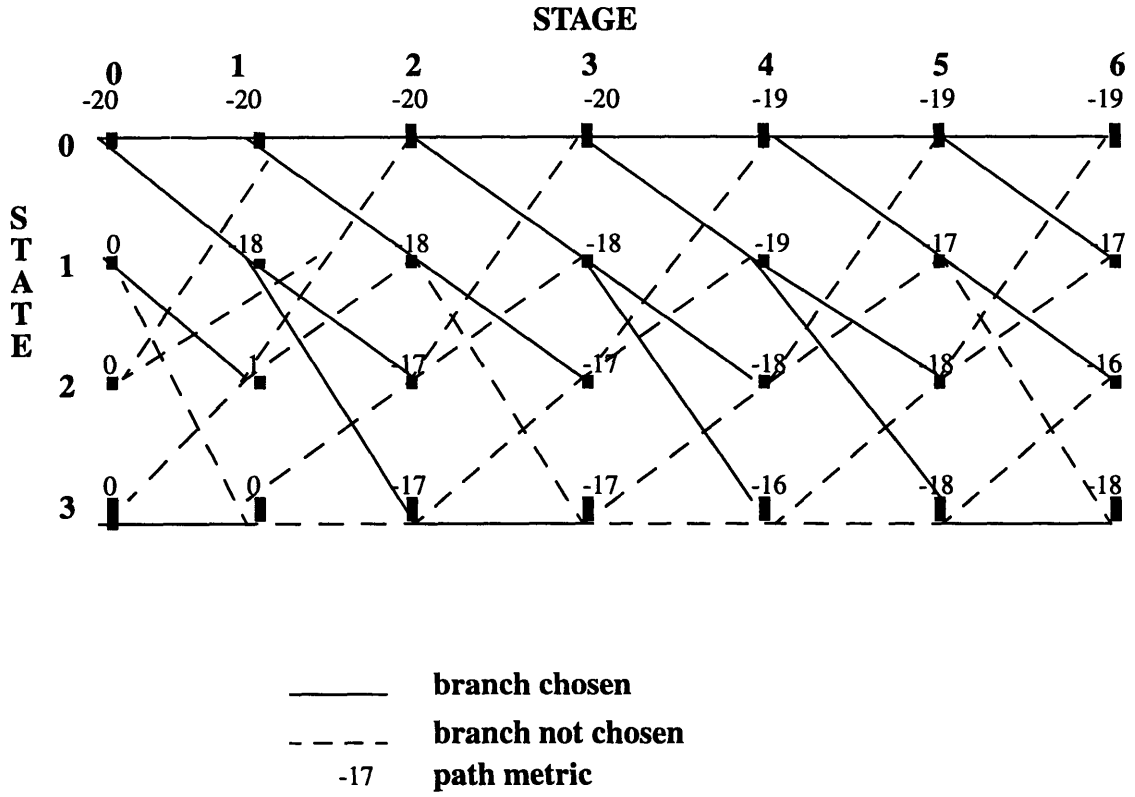

Figure 14: Example of Viterbi error correction.

Another way of looking at it is that the received bits are closer in Hamming distance to the all-zero path than to any other possible trellis path. For an incorrect trellis path to be chosen, then, the received bits must be closer to the incorrect path than to the all-zero path. But every possible incorrect path is several consecutive errors away from the all-zero path. This explains why convolutional codes can correct random bit errors

more easily than burst errors.[23] Unfortunately, burst errors are characteristic of cellular channels.

4.3 Interleaving

To combat burst errors, many systems mix the data bits before modulation, so that adjacent bits are separated. The bits are unmixed at the receiver. Any burst errors that occurred will then be separated due to the unmixing and appear more random. This mixing is called interleaving, and it could occur within one frame of data (intraframe) or between two or more frames (interframe).[20][23]

Figure 15 shows a simple example of interframe block interleaving. Bits from two frames are placed in a matrix row by row. The output, read out column by column, is mixed. On the channel, the data is hit by a burst error of length four. The incorrect bits are separate after deinterleaving, making them easier to correct. The disadvantage of interleaving is that it increases delay, since deinterleaving cannot begin until all the bits in both frames have been received.

4.4 GSM channel coder

The GSM channel coding scheme for speech data (Figure 16) incorporates all the methods discussed above.[5][20] For each 20 ms frame of speech, the speech coder outputs 260 bits. In the channel coder, these bits are divided into three classes, based on their importance to perceptual speech quality. Each class of bits is processed differently.

The most important bits are in class 1a (C1a). There are 50 bits in this class. They are protected by a CRC of 3 bits, with generator polynomial x^3+x+1. The next most important bits are in class 1b (C1b). There are 132 bits in this class. The C1a and C1b bits are intraframe interleaved. Even-numbered bits (91 bits) are placed in front, and odd-num-

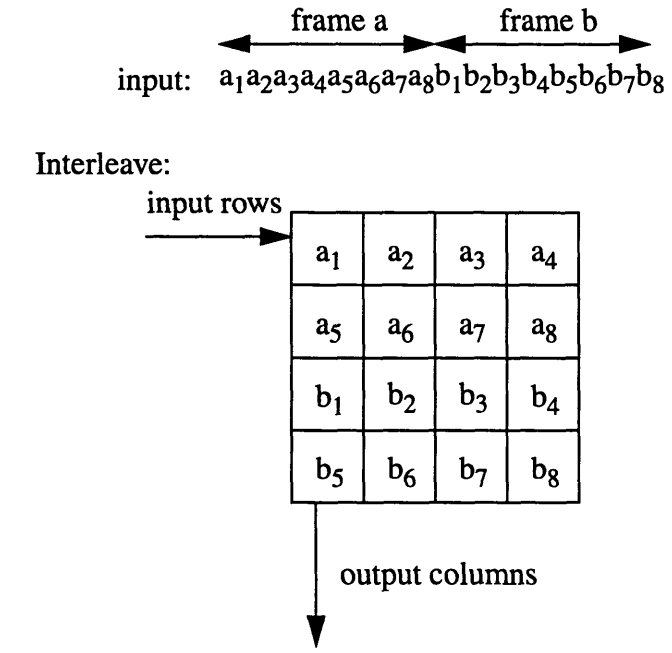

Figure 15: Block interleaving over two frames.

bered bits (91 bits) behind, with the three parity check bits in the middle. Then the bits are put through a convolutional code of constraint length 4 and rate 1/2. The two generator polynomials are $1+x^3+x^4$ and $1+x+x^3+x^4$. The total number of bits processed through the convolutional coder is 189 bits (C1a, C1b, parity, and 4 tail bits to clear to shift register), giving an output of 378 bits. Finally, class 2 bits (C2), the least important bits, are appended without coding, producing a total output of 456 bits per frame. (For more detail on bit classes, see Appendix A.)

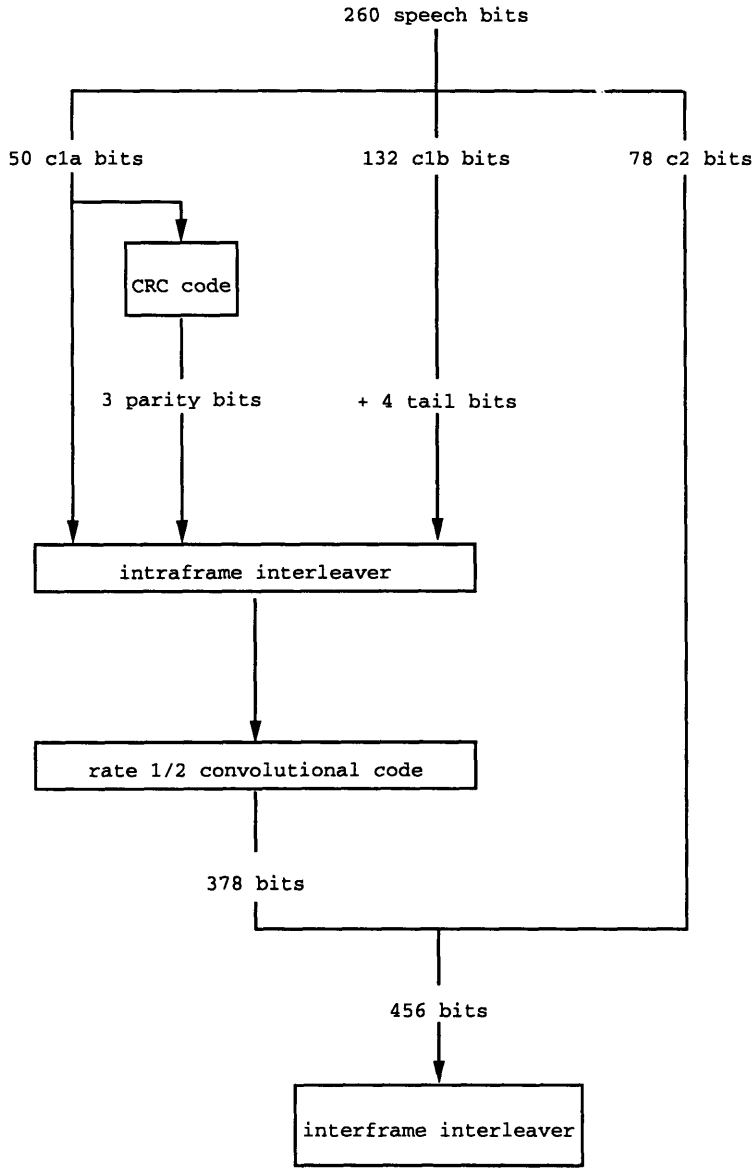

Figure 16: GSM channel coding scheme.

The 456 bits are interleaved with bits from the 2 adjacent frames. The method of interleaving is called block diagonal. It involves filling an 8x114 matrix. The first frame begins at [0,0] and places elements in [i,j] position of the matrix according to

$$i = (k) mod(8) \tag{5}$$

$$j = 2[(49k) mod(57)] + [((k) mod(8)) div(4)], \tag{6}$$

where $k = [0...455]$ is the number of the bit to be interleaved. At this point, only half the matrix elements are filled. The first 4 columns, containing half the bits from frame 1, are outputted to the modulator. The next frame begins at [0,4] and places elements according to

$$i = (k)mod(8) + 4 \tag{7}$$

$$j = 2[(49k)mod(57)] + [((k)mod(8))div(4)]. \tag{8}$$

The last 4 columns are now outputted. They contain the other half of the bits from frame 1 and half the bits from frame 2. The bits from frame 3 are then placed starting from [0,0], overwriting frame 1's bits. The first 4 columns are outputted again. This time, they contain the rest of frame 2 and half of frame 3. Interleaving continues, with alternating sets of output columns, until all frames have been processed.

The GSM standard gives no specifications for channel decoding.

Chapter 5

Design

Significant advances in speech coding have occurred in the decade since the GSM standard was passed. This project involved the design of a channel coding scheme for a new speech coder. The speech coder gives improved speech quality in background noise and burst errors. The intent was to propose the combined speech and channel coder system in an open competition for a new European standard, called enhanced full rate GSM. This chapter describes the speech coder, the design goals, and the rationalization behind the final channel coding scheme.

5.1 Speech coder

The new speech coder is a Code-Excited Linear Predictive (CELP) coder (Figure 17).[20][22][20] Like the RPE-LTP coder in the current standard, CELP coders model speech as a filter excited by a signal. The filter coefficients are first found through linear prediction. Then the coefficients are used to determine the excitation parameters. The excitation signal is composed of two parts that are analyzed separately. The first part of the signal is the adaptive codebook excitation. It relates to the pitch period and is found through an analysis-by-synthesis process. Different pitch values are used to construct candidate excitation signals. The candidate signals are put through the LPC filter to synthesize the input speech. The synthesized speech is compared to the actual input. The pitch value that gives the least error is chosen. The second part of the excitation is called the fixed codebook excitation. It is chosen from a fixed set of candidate vectors, again through analysis-by-synthesis. The fixed codebook candidate and the previously determined adaptive

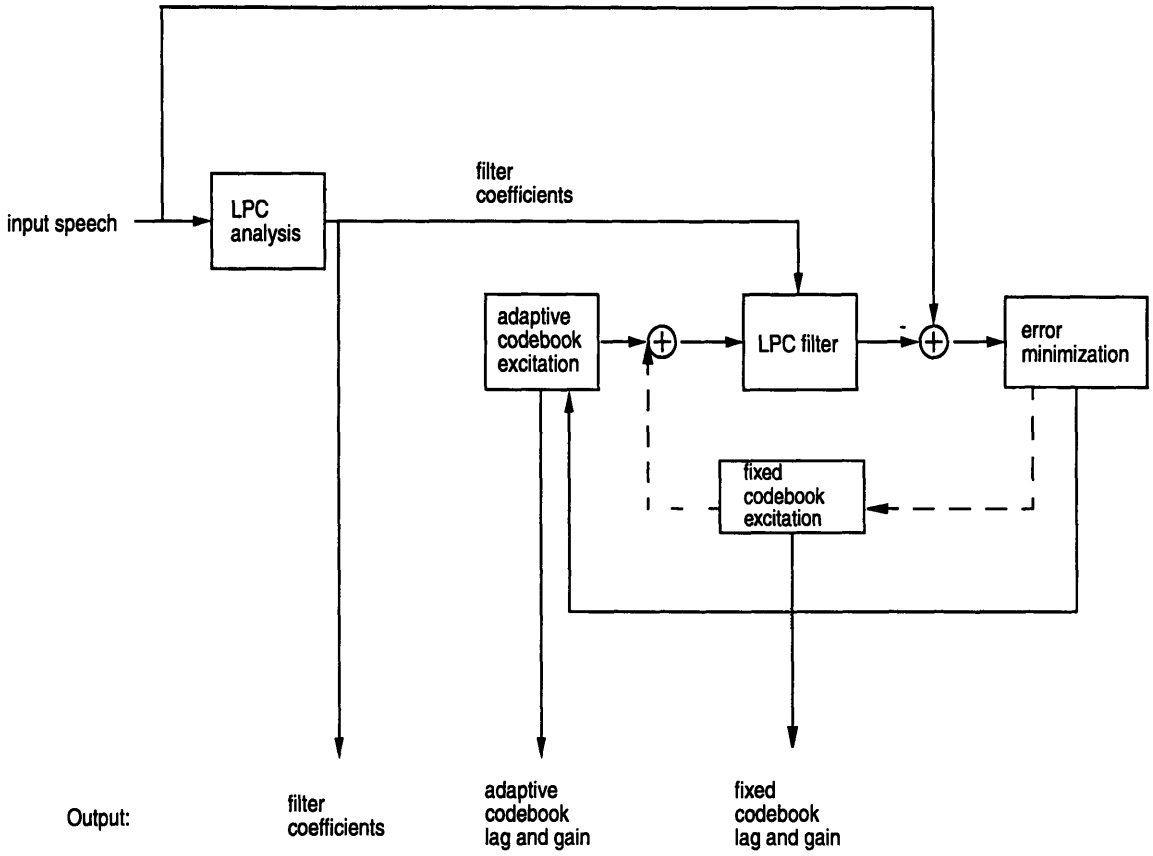

Figure 17: CELP speech encoder.

codebook excitation are added together to form a full excitation signal. The excitation is passed through the LPC filter to create synthesized speech. The fixed codebook vector that gives the least error between synthesized and actual speech is chosen.

The new coder improves on the current GSM speech coder in several ways.[11] The quantization of the parameters is designed for robustness in background noise. The lower bit rate allows for more error protection. In addition, the error concealment algorithm is improved. In the case of parity check failures, instead of repeating the entire frame, the speech coder repeats only the affected parameters.

Two versions of the coder were considered, one with 95 bits per frame and one with 119 bits. As the coder's frame size is 10ms, this gives bit rates of 9.5 kbps and 11.0

kbps, respectively. The main difference is that the 119 bit coder has more bits for excitation. Table 2 shows the bit allocations for both versions.

Parameter	Number of bits	
	95 bit	119 bit
10 LPC filter coefficients	24	24
adaptive codebook lag	13	13
adaptive codebook gain	8	8
fixed codebook info	40	64
fixed codebook gain	10	10
Total	95	119

Table 2: Bit allocation of two versions of new speech coder.

5.2 Design goals

The goal of the project was to design a channel coder so that the overall system would produce output speech of much better quality than the current GSM standard. The design specifications were set by the European Telecommunications Standards Institute (ETSI). The main limitations were complexity, bit rate, and delay. Specifically, the relevant ETSI requirements on the entire system are as follows:[6][7]

- Speech quality -- better than the current standard
- Complexity -- no more than the current half-rate standard
- Bit rate -- 22.8 kbps
- Delay -- no more then the current standard

5.3 Channel model

This project assumed a Rayleigh fading channel with frequency hopping every TDMA burst. Three error conditions were modeled -- C/I ratios 10dB, 7dB, and 4dB. This meant the average bit error rates were 3%, 10%, and 13%, respectively. The model was realized in the form of error mask files that were applied directly to the output bits of the channel coder. The files contained 8-bit values indicating the probability that a bit was corrupted. These files replaced the modulator, channel, and demodulator blocks in Figure 4.

5.4 Coding schemes

Several candidate schemes were implemented for each version of the speech coder. All had the same basic structure of a CRC and convolutional code, but differed in details such as code rate and number of parity bits. Since no proven objective method of judging speech quality exists at present, the schemes were evaluated with a subjective listening test. Speech was processed through the speech encoder, channel encoder, error mask function, channel decoder, and speech decoder. The output speech was then played. Attention was paid to clarity of speech and the existence of unwanted artifacts, such as pops or clicks. The schemes with the best output speech quality over all channel types were chosen.

Figure 18 and Figure 19 show the final schemes for the two coders. The data bits are grouped into classes, depending on their importance to speech quality. As in the current GSM standard, the more important bits are given more protection. The most important bits are in class 0 (C0). These 29 bits are the MSBs of the filter parameters. They are put through a CRC, then a convolutional code. The next most important bits, the class 1 (C1) bits, mostly consist of the excitation bits and the LSBs of the filter parameters. These

are processed through a less powerful convolutional code. The least important bits are class 2 (C2) bits, which are left uncoded. Only the 95 bit coder has class 2 bits. (For more detail on bit classes, see Appendix A.) The following sections present the reasons behind the two schemes.

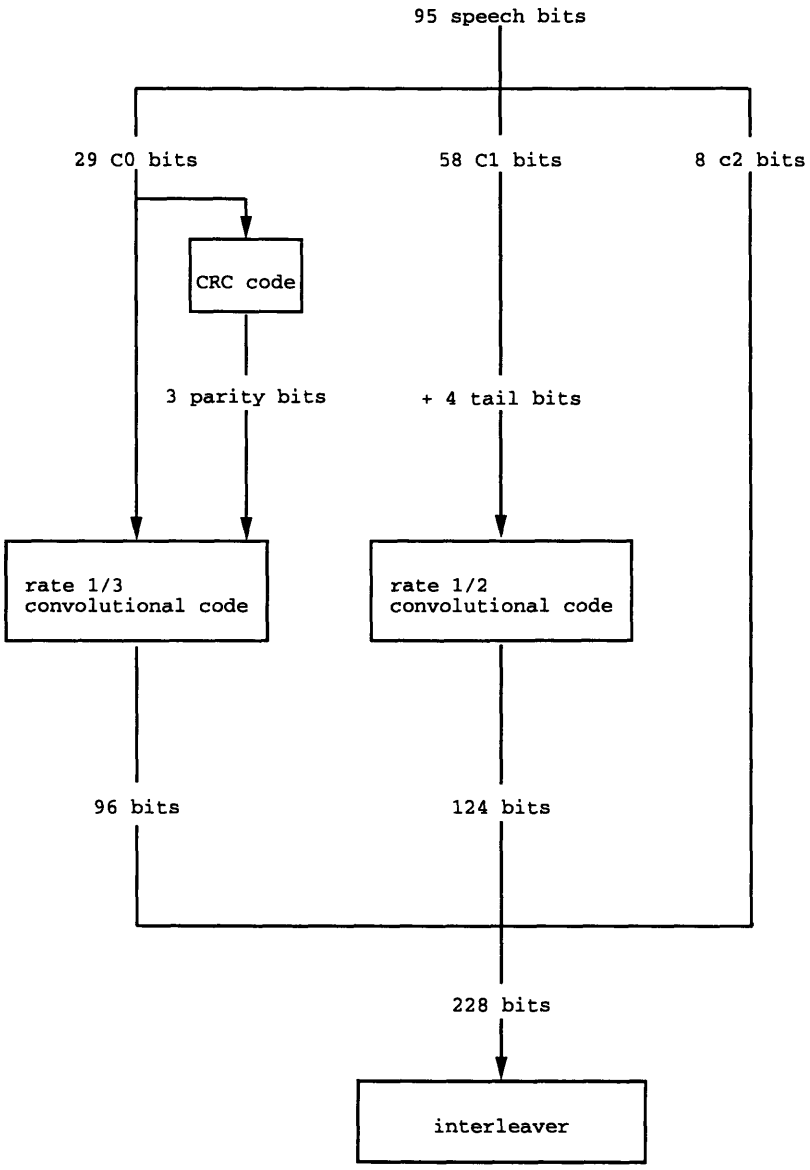

Figure 18: Channel coding scheme for 95 bit coder.

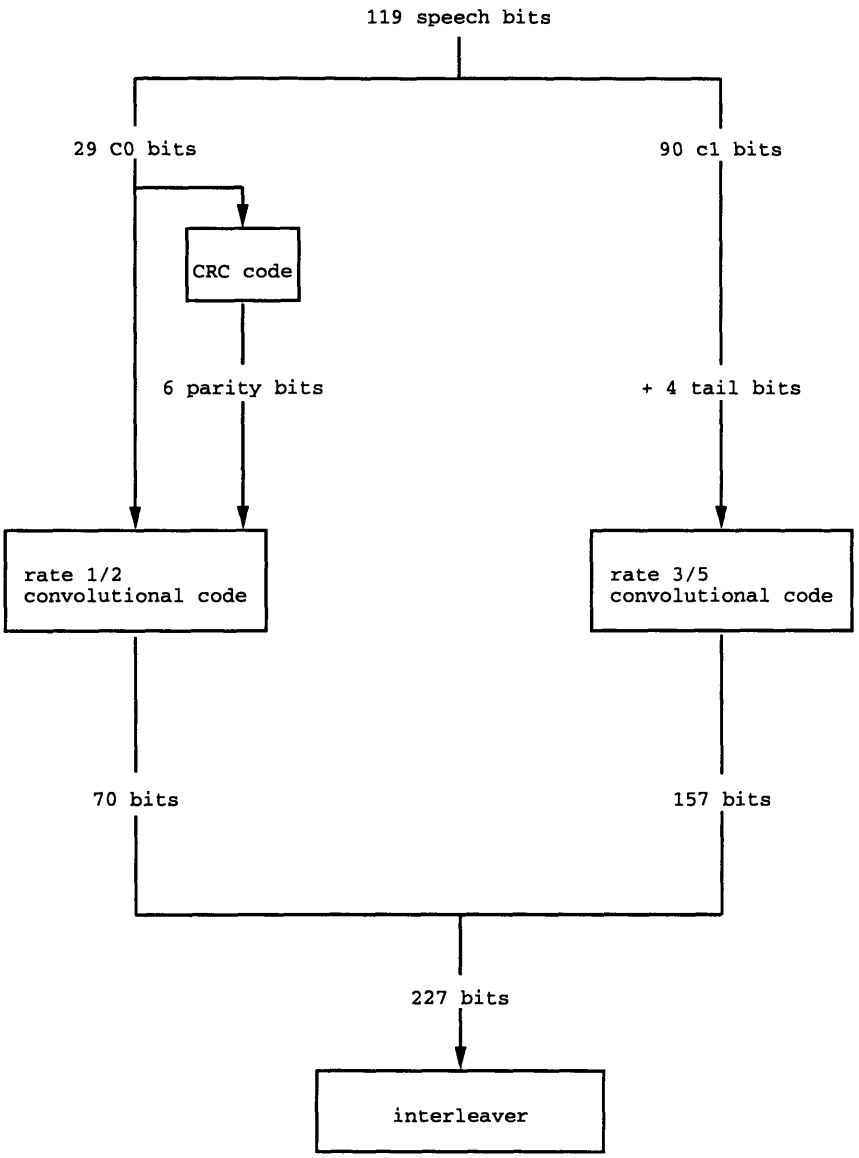

Figure 19: Channel coding scheme for 119 bit coder.

5.4.1 Convolutional code

The design began with the convolutional code. It has 3 characteristics: constraint length, rate, and generator polynomial.

The constraint length was limited by the complexity requirement. The most computationally intensive part of the channel coder is the Viterbi decoder. Increasing the constraint length by 1 doubles the number of states in the trellis, which in turn doubles the

computation for decoding. Thus, to meet the complexity requirement, the constraint length was chosen to be 4, the same as the current half-rate standard.

The code rate was limited by the bit rate requirement and the output bit rate of the speech coder. It was found experimentally that a code of rate 1/3 corrects almost all the channel errors, even at a C/I ratio of 4dB. The ideal solution would be to put all the data bits through a rate-1/3 code. However, this would result in too high an output bit rate. A logical scheme is to use a rate-1/3 code on some bits and code the rest of the bits with a different rate. This is the coding scheme for the 95 bit coder. A rate-1/3 code was used for the class 0 bits, while a rate-1/2 code was used for class 1. This left 8 bits uncoded.

For the 119 bit coder, it was found experimentally that the excitation bits were more perceptually important than originally thought. In fact, for good speech quality, all the bits in this version of the coder needed some protection. Thus, a rate-1/2 coder was used on C0 bits. While not as effective at correcting errors as a rate-1/3 code, the code had a lower bit rate, allowing protection for all data bits. The rest of the bits were coded a rate-1/2 code, punctured to rate-3/5. A puncturing period of 3 and a pattern of 111110 was chosen, based on listening tests.

As for generator polynomials, the two final schemes need only 2 sets of polynomials, one for each code rate. The current GSM standard has generator polynomials for codes of rate 1/2 and 1/3. These were chosen because they are proven to be effective. For the rate-1/3 code, the polynomials are $1+x+x^3+x^4$, $1+x^2+x^4$, and $1+x+x^2+x^3+x^4$. For the rate-1/2 code, the polynomials are $1+x^3+x^4$ and $1+x+x^3+x^4$.[5]

A detail of note is that a total of only 4 tail bits were required to clear the shift register in each scheme, although the schemes called for two different rate codes. This is

because the shift register state was continuous from one rate to the next. In the 95 bit coder, for example, the 29 C0 bits were shifted in and coded with the rate 1/3 code. When the 30th bit was shifted in, the rate 1/2 code began processing immediately, with bits 26-29 still in the shift register. Thus, for the two code rates, the shift register is cleared only once. This saves four bits for other uses.

5.4.2 CRC

The key issue in the design of the CRC was the number of parity bits required. Too few parity bits would not catch enough errors, while too many bits would cause false detection. Two schemes were tested with this channel model, one with 3 parity bits (polynomial x^3+x+1) and one with 6 parity bits (polynomial x^6+x+1). The generator polynomials were taken from Blahut.[1] It was found that 6 parity bits caught considerably more bit errors, while not resulting in false detection too often.

Thus, 6 parity bits were used in the 119 bit coder. In the 95 bit coder, however, 3 parity bits were sufficient because of the high performance of the rate 1/3 code.

5.4.3 Interleaving

The interleaving used was the same as that of the current standard. This is sufficient to meet the delay requirement.

5.4.4 Decoder

The decoder used a Viterbi algorithm, as described in Chapter 4. Soft-decision was possible because the channel model had 8-bit sensitivity. A Euclidean distance measure was chosen for maximum-likelihood decoding.

The decoder also included a modification to the Viterbi algorithm. This is a significant contribution of the project. The metrics of the Viterbi algorithm were used to provide

a reliability measure on the output speech parameters, supplementing the parity check (Figure 20). Each step along the decoded trellis path increments the overall path metric.

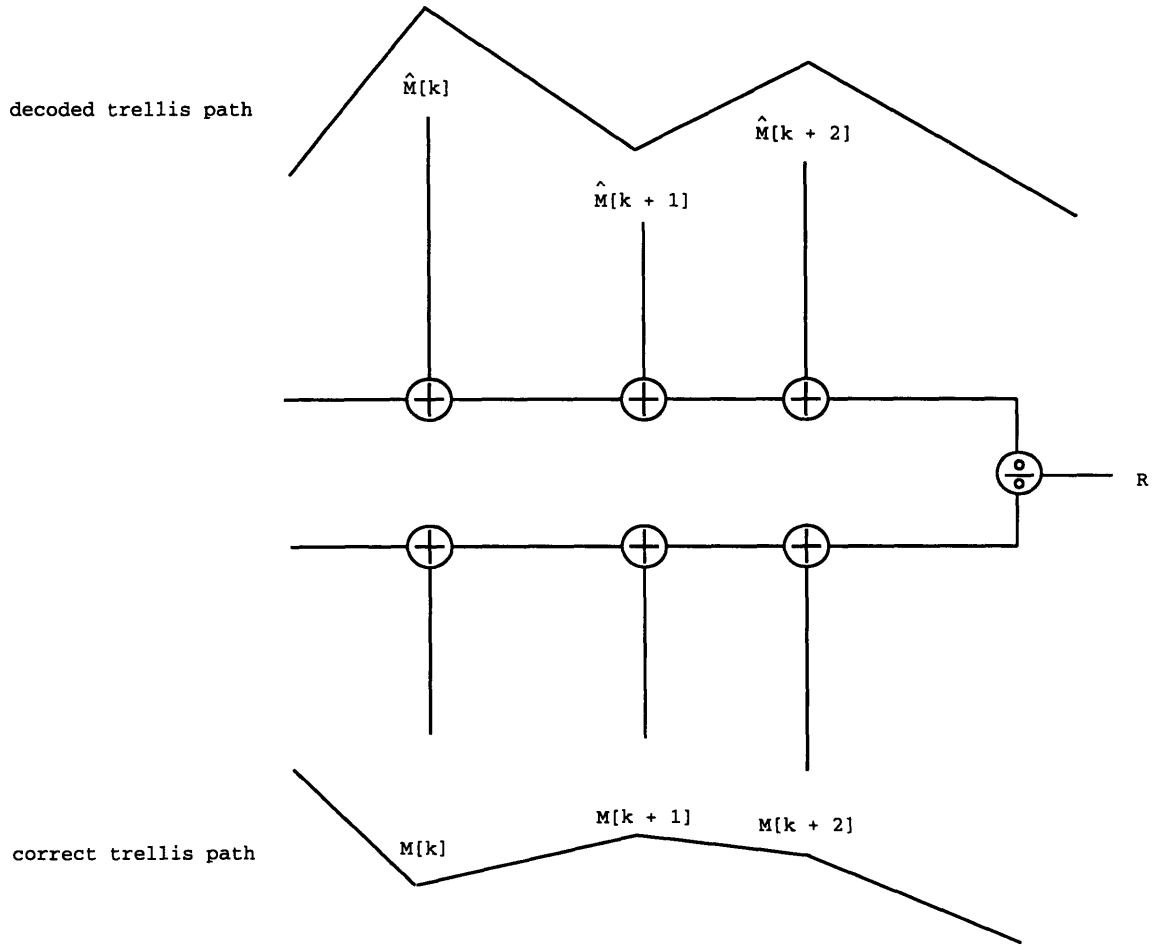

Figure 20: Error detection on a three-bit parameter using Viterbi metrics.

The amount of increment depends on the reliability of the received bits. The reliability of a speech parameter is computed by comparing the sum of the path metric increments of each bit to the ideal sum. The reliability values can then be sent to the speech coder, where appropriate action can be taken. In Figure 20, $\hat{M}[k]$ represents the actual path metric increment for bit k in the trellis, while $M[k]$ represents the ideal increment. For a parameter that begins at bit k and ends at $k + 2$, the reliability value R is

$$R = \frac{\sum_{i=0}^{2} \hat{M}[k+i]}{\sum_{i=0}^{2} M[k+i]} \tag{9}$$

In this design, reliability value was computed only from C0 bits. The value was scaled to be a number between 0 and 15 (4-bit sensitivity), 15 being the least reliable. A parity failure on the C0 bits automatically set the value to 15. In the speech coder, the value was compared to a threshold. The threshold could be different for each parameter. If the value was higher than the threshold, the current parameter was discarded and the previous frame's parameter was used. This modification thus allowed error detection on individual speech parameters.[15]

5.4.5 Code performance

The theoretical upper and lower bounds for post-decoding bit error probability (P_b) are shown in Table 3.[21][23] (For equations, see Appendix B.) The upper bounds are considerably generous at high channel error rates; the post-decoding bit error rates (BER) exceed unity. At low channel error rates, the rate 1/3 code shows better performance than the other two codes, and the rate 1/2 code performs better than the rate 3/5 punctured code.

The bounds were calculated assuming hard-decision decoding and random channel errors. In the Rayleigh fading channel of this design, channel errors are bursty rather than

random. This will result in higher post-decoding BERs. On the other hand, the interleaving and soft-decision decoding in this design will mitigate the effects of the burst errors.

Code rate	Channel BER					
	3%		10%		13%	
	low	high	low	high	low	high
1/2	<0.0001	0.006	0.003	0.5	0.007	>1
1/3[1]	<0.0001	<0.0001	0.0003	0.026	0.001	0.098
punctured 3/5[2]	0.0001	0.04	0.003	>1	0.006	>1

Table 3: Bounds on post-decoding BER.

1. The upper bounds for this code were calculated using only the d_{free} term of the weight distribution. (See Appendix B.)

2. The rate 1/2 code that was punctured to produce this code has generator polynomials $x^4 + x^3 + 1$ and $x^4 + x^2 + x + 1$. This is not the code used in this design, but one with the same d_{free}.

Chapter 6

Results

6.1 Bit error rates

To show the effects of channel coding, the average bit error rate (BER) was found for the two coding schemes under different channel conditions (Figure 21). The error rates were averaged over 8 files of 300 speech frames each. The average channel BER is included for reference. Coding reduces the error rate considerably at 10dB and less so at 4dB. In fact, for the 119 bit coder, the BER at 4dB is close to the channel BER. This means that the 4dB channel condition approaches the limits of the error correcting capability of the rate 1/2 code.

Comparing the two versions of the speech coder, the 95 bit coder has a lower BER at a C/I of 4dB. This is predictable, since it uses the more powerful rate 1/3 code. However, the 119 bit coder has a somewhat lower error rate at 10dB. This is most likely due to the fact that the C2 bits of the 95 bit coder are unprotected.

6.2 Mean-opinion-score tests

The ultimate test of the channel coder is how good the decoded speech sounds. To show this, mean-opinion-score (MOS) tests were performed. In the test, participants listen to speech sentences and rank the quality of the speech on a scale of 1 to 5, 5 being excellent and 1 being poor. In this test, there were 25 listeners, and the sentences included both female and male speakers. The average scores are shown in Figure 22. Four channel conditions were tested: EP0 is no errors, EP1 is 10dB, EP2 is 7dB, and EP3 is 4dB. TI-EFR1 was the 95 bit coder. TI-EFR2 was the 119 bit coder. In concurrence with the graph of

average bit error, the 95 bit coder was rated better at a C/I of 4dB, while the 119 bit coder sounded better at 10dB.

The MOS scores may seem surprisingly high, given the high post-decoding BER (in the EP2 condition, for example). This can be explained by the coding of the bit classes. C0 bits, the most important to speech quality, are given the most error protection. Thus, more of the bit errors occur in C1 and C2 bits, which are not as crucial. Also, the error concealment scheme of the speech coder plays an important role in speech quality. If an error is detected in the C0 bits, the speech coder repeats the data of the previous frame. To the ear, the speech will not sound bad.

For comparison, two other speech coders were also tested. TCH-FS is the current standard. Both TI-EFR coders performed much better this coder. PCS1900 is a competing coder. It is currently accepted in the US as a standard at 1.9 GHz. The TI-EFR coders performed similarly to PCS1900, with no coder a clear winner.[10]

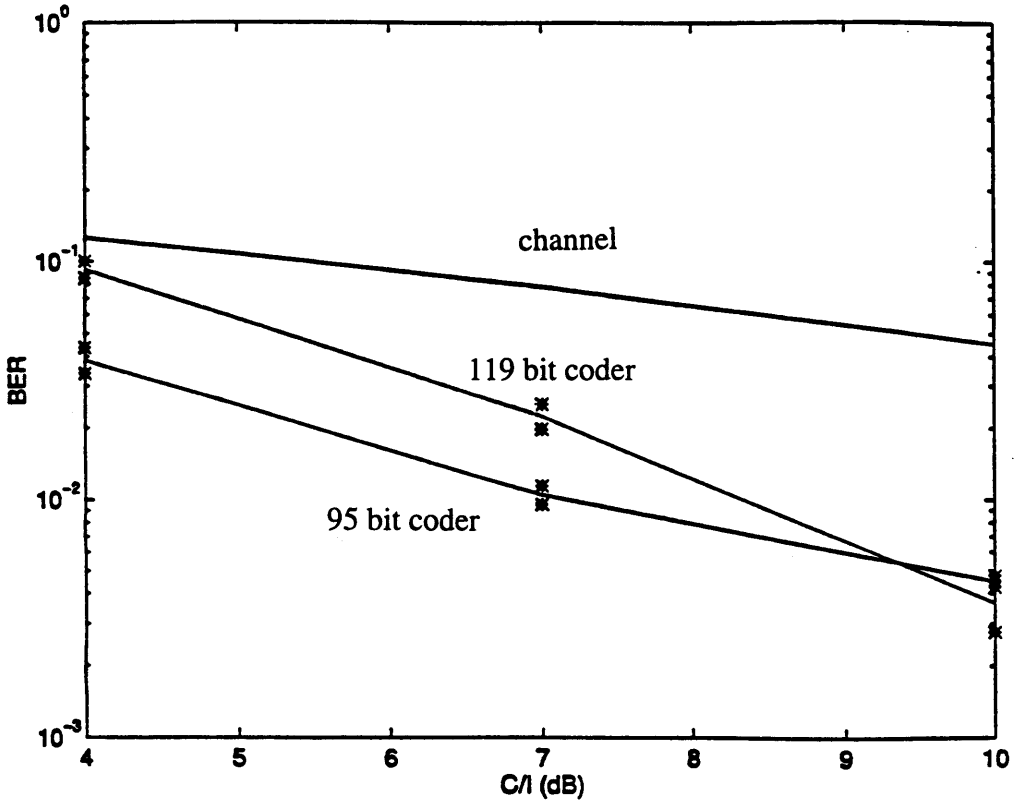

Figure 21: Average BER for two coders with channel coding.

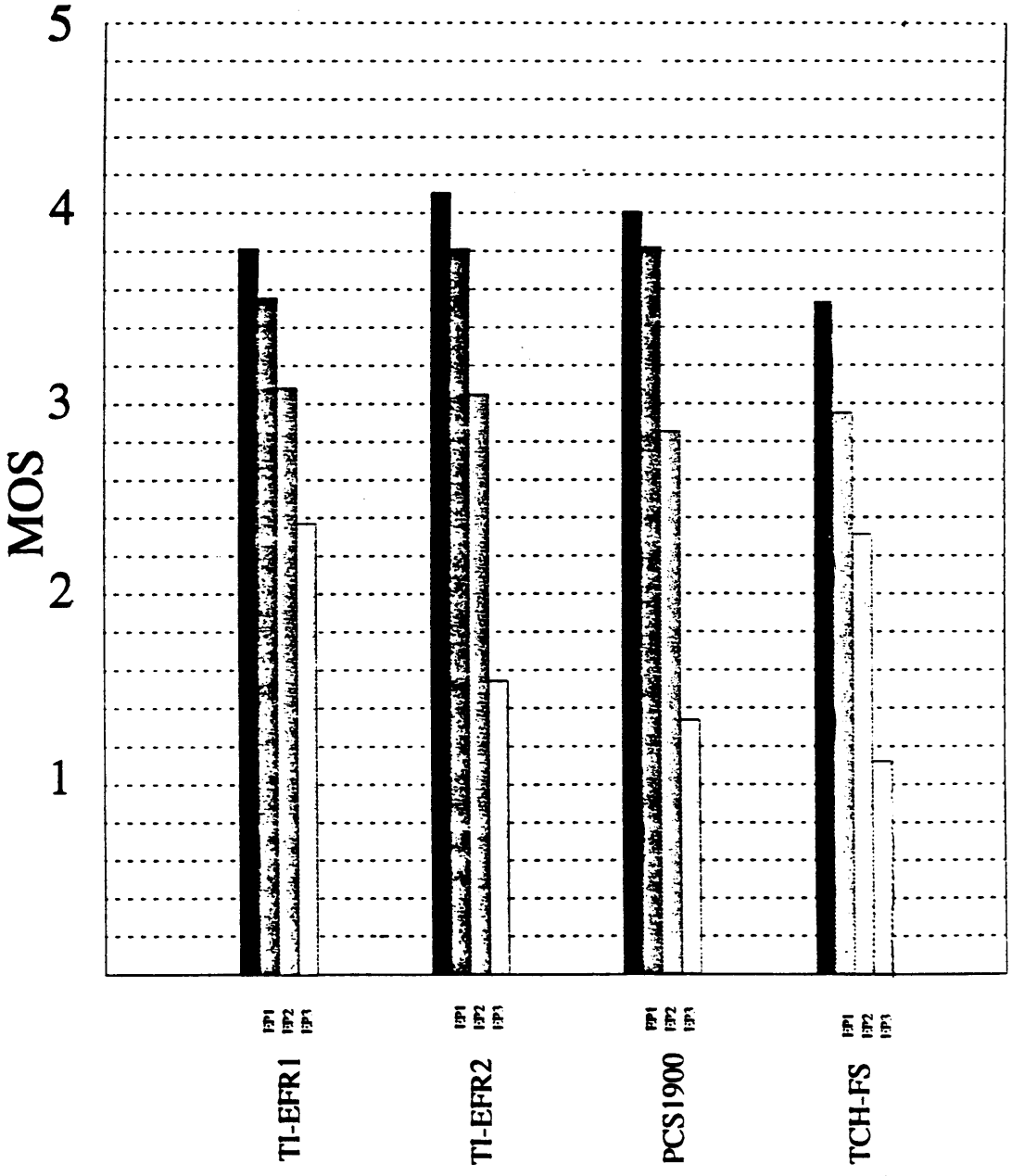

Figure 22: MOS results.[1]

1. This plot made by W. P. LeBlanc.

Chapter 7

Conclusions

The main focus of the design was on speech quality of the overall system. In that context, the results were quite good. Judging by the MOS scores, both TI-EFR coders performed considerably better than the current GSM standard and were comparable to the competing PCS1900 coder. The channel coder itself performed well, reducing average bit error rates significantly, especially at a C/I of 10dB. Of the two versions of the TI-EFR coder, the 119 bit coder performed better at 10dB in both BER and MOS tests, and the 95 bit coder performed better at 4dB.

The system should meet the delay and complexity requirements set by ETSI, although only estimates are available. For delay, the speech coder introduces a delay of 15 ms (10ms frame size + 5 ms look ahead). The current standard has a speech coder delay of 20 ms. The channel coder introduces the same delay as the current standard, due to interleaving. The total delay is thus 5 ms less than the current standard. For complexity, the speech coder has approximately the same complexity as the speech coder in the current half-rate standard. The channel coder is also similar, with most of the complexity residing in the Viterbi algorithm.

Most of the improvements in MOS scores, as compared to the current GSM standard, were due to the performance of the new speech coder, rather than advances in channel coding. In fact, the channel coding scheme implemented in this design closely followed the scheme in the current standard. One important difference is in the use of the Viterbi algorithm for error detection. For future investigation, I would like to look at the modified Viterbi in more detail. For example, what is its effect on MOS scores and BER?

How does its performance compare with that of more traditional methods of error detection, such as CRC codes? Also, it would be valuable to develop a channel coder with lower interleaving delay, since this would allow more flexibility in other parts of the system. Given the new speech coder's smaller frame size, a different interleaving scheme could reduce delay considerably, but at what cost to speech quality? This is another area for future development.

Appendix A: Classes of speech coder bits

Parameter name	Number of bits	Bit classes (MSB-LSB)
STP 1	6	C1a, C1a, C1a, C1a, C1b, C2
STP 2	6	C1a, C1a, C1a, C1b, C2, C2
STP 3	5	C1a, C1a, C1b, C2, C2
STP 4	5	C1a, C1a, C1b, C2, C2
STP 5	4	C1a, C1b, C1b, C2
STP 6	4	C1a, C1b, C2, C2
STP 7	3	C1a, C1b, C2
STP 8	3	C1b, C2, C2
Subframe 1:		
LTP gain	2	C1b, C1b
LTP lag	7	C1a, C1a, C1a, C1a, C1a, C1a, C1b
RPE grid	2	C1b, C1b
max amplitude	6	C1a, C1a, C1a, C1b, C1b, C2
RPE pulse 1	3	C1b, C1b, C2
RPE pulse 2	3	C1b, C1b, C2
RPE pulse 3	3	C!b, C1b, C2
RPE pulse 4	3	C1b, C1b, C2
RPE pulse 5	3	C1b, C1b, C2
RPE pulse 6	3	C!b, C1b, C2
RPE pulse 7	3	C1b, C1b, C2
RPE pulse 8	3	C1b, C1b, C2
RPE pulse 9	3	C!b, C1b, C2
RPE pulse 10	3	C1b, C1b, C2
RPE pulse 11	3	C1b, C1b, C2
RPE pulse 12	3	C!b, C1b, C2
RPE pulse 13	3	C1b, C1b, C2
Subframe 2:		
LTP gain	2	C1b, C1b
LTP lag	7	C1a, C1a, C1a, C1a, C1a, C1a, C1b
RPE grid	2	C1b, C1b
max amplitude	6	C1a, C1a, C1a, C1b, C1b, C2
RPE pulse 1	3	C1b, C1b, C2
RPE pulse 2	3	C1b, C1b, C2
RPE pulse 3	3	C!b, C1b, C2
RPE pulse 4	3	C1b, C1b, C2
RPE pulse 5	3	C1b, C1b, C2
RPE pulse 6	3	C!b, C1b, C2
RPE pulse 7	3	C1b, C1b, C2

Table 4: GSM speech coder bit classes.[5]

Parameter name	Number of bits	Bit classes (MSB-LSB)
RPE pulse 8	3	C1b, C1b, C2
RPE pulse 9	3	C!b, C1b, C2
RPE pulse 10	3	C1b, C1b, C2
RPE pulse 11	3	C1b, C1b, C2
RPE pulse 12	3	C!b, C1b, C2
RPE pulse 13	3	C1b, C1b, C2
Subframe 3:		
LTP gain	2	C1b, C1b
LTP lag	7	C1a, C1a, C1a, C1a, C1a, C1a, C1b
RPE grid	2	C1b, C1b
max amplitude	6	C1a, C1a, C1a, C1b, C1b, C2
RPE pulse 1	3	C1b, C1b, C2
RPE pulse 2	3	C1b, C1b, C2
RPE pulse 3	3	C!b, C1b, C2
RPE pulse 4	3	C1b, C1b, C2
RPE pulse 5	3	C1b, C1b, C2
RPE pulse 6	3	C!b, C1b, C2
RPE pulse 7	3	C1b, C1b, C2
RPE pulse 8	3	C1b, C1b, C2
RPE pulse 9	3	C!b, C1b, C2
RPE pulse 10	3	C1b, C1b, C2
RPE pulse 11	3	C1b, C1b, C2
RPE pulse 12	3	C!b, C1b, C2
RPE pulse 13	3	C1b, C1b, C2
Subframe 4:		
LTP gain	2	C1b, C1b
LTP lag	7	C1a, C1a, C1a, C1a, C1a, C1a, C1b
RPE grid	2	C1b, C1b
max amplitude	6	C1a, C1a, C1a, C1b, C1b, C2
RPE pulse 1	3	C1b, C1b, C2
RPE pulse 2	3	C1b, C1b, C2
RPE pulse 3	3	C!b, C1b, C2
RPE pulse 4	3	C1b, C1b, C2
RPE pulse 5	3	C1b, C2, C2
RPE pulse 6	3	C!b, C2, C2
RPE pulse 7	3	C1b, C2, C2
RPE pulse 8	3	C1b, C2, C2
RPE pulse 9	3	C!b, C2, C2
RPE pulse 10	3	C1b, C2, C2
RPE pulse 11	3	C1b, C2, C2

Table 4: GSM speech coder bit classes.[5]

Parameter name	Number of bits	Bit classes (MSB-LSB)
RPE pulse 12	3	C!b, C2, C2
RPE pulse 13	3	C1b, C2, C2

Table 4: GSM speech coder bit classes.[5]

Parameter name	Number of bits	Bit classes (MSB-LSB)
LPC stage 1	6	C0, C0, C0, C0, C0, C0
LPC stage 2	6	C0, C0, C0, C0, C0, C0
LPC stage 3	6	C1, C1, C1, C1, C1, C1
LPC stage 4	6	C1, C1, C1, C1, C1, C1
adaptive codebook lag 1	8	C0, C0, C0, C0, C0, C1, C1, C1
adaptive codebook lag 2	5	C0, C0, C1, C1, C1
adaptive codebook gain 1	4	C0, C0, C1, C1
adaptive codebook gain 2	4	C0, C0, C1, C1
fixed codebook info	40	C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C1, C2, C2, C2, C2, C2, C2, C2, C2
fixed codebook gain 1	5	C0, C0, C0, C1, C1
fixed codebook gain 2	5	C0, C0, C0, C1, C1

Table 5: 95 bit coder bit classes.

Parameter name	Number of bits	Bit classes (MSB-LSB)
LPC stage 1	6	C0, C0, C0, C0, C0, C0
LPC stage 2	6	C0, C0, C0, C0, C0, C0
LPC stage 3	6	C1, C1, C1, C1, C1, C1
LPC stage 4	6	C1, C1, C1, C1, C1, C1
adaptive codebook lag 1	8	C0, C0, C0, C0, C0, C1, C1, C1
adaptive codebook lag 2	5	C0, C0, C1, C1, C1
adaptive codebook gain 1	4	C0, C0, C1, C1
adaptive codebook gain 2	4	C0, C0, C1, C1
fixed codebook info	64	all C1
fixed codebook gain 1	5	C0, C0, C0, C1, C1
fixed codebook gain 2	5	C0, C0, C0, C1, C1

Table 6: 119 bit coder bit classes.

Appendix B: Calculation of bounds on post-decoding BER

B.1 Weight distribution

The weight distribution of a convolutional code is used to determine bounds on the post-decoding BER (P_b).[21] It is defined by

$$W(d) = \sum_{d = d_{free}}^{\infty} W_d D^d \tag{10}$$

d is the Hamming distance between two trellis paths. There can be several paths that are a distance d away from a given path. d_{free} is the free distance, the minimum Hamming distance between any two trellis paths. The higher d_{free}, the better the code. W_d is the total number of bit errors in all the paths of distance d. The sequence of W_d's is referred to as c_d. For example, a code with $W(d) = D^7 + 4D^8 + 18D^9$ has $d_{free} = 7$ and $c_d = [1, 4, 18]$. Table 7 shows d_{free} and c_d for the codes analyzed in the text.

Code rate[1]	d_{free}	c_d
1/2	7	[4,12,20,...]
1/3	12	[12,...]
puncture 3/5	5	[1,39,104,...]

Table 7: Weight distribution parameters of convolutional codes.

1. Parameters for rate 1/2 code taken from [21], for rate 1/3 taken from [3], for rate 3/5 taken from [12].

B.2 Upper bound

For hard-decision decoding, the upper bound on P_b is given by

$$P_b = \frac{1}{k} \sum_{d = d_{free}}^{\infty} W_d [4p(1-p)]^{\frac{d}{2}}, \tag{11}$$

where p is the channel BER and k is the number of bits inputted to the encoder at a time. For punctured codes, k is the puncturing period. (For derivation, see [21].)

B.3 Lower bound

A binary symmetric channel, or BSC, (Figure 23) is a memoryless channel with two symbols, 0 and 1.[23] The probability of bit error is the same, regardless of the symbol transmitted. A BSC is an accurate model of hard-decision decoding on a channel with random bit errors.

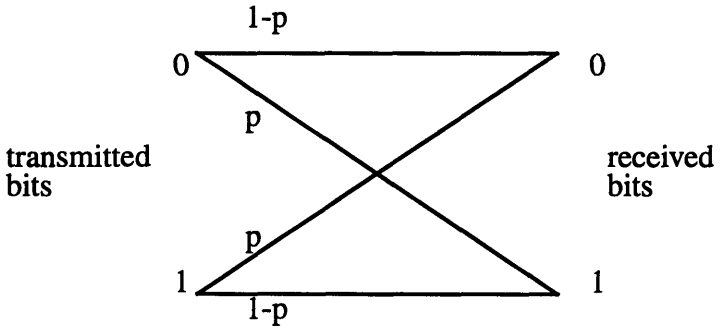

Figure 23: Binary symmetric channel

For a BSC, the lower bound on P_b is given by

$$P_b = \begin{cases} \frac{1}{k} \sum_{i = \frac{d_{free}+1}{2}}^{d_{free}} \binom{d_{free}}{i} p^i (1-p)^{d_{free}-i} & d_{free} odd \\ \frac{1}{2k} \binom{d_{free}}{\frac{d_{free}}{2}} p^{\frac{d_{free}}{2}} (1-p)^{\frac{d_{free}}{2}} + \frac{1}{k} \sum_{i = \frac{d_{free}}{2}+1}^{d_{free}} \binom{d_{free}}{i} p^i (1-p)^{d_{free}-i} & d_{free} even \end{cases} \tag{12}$$

where p is the channel BER and k is the number of bits inputted to the encoder at a time. For punctured codes, k is the puncturing period. (For derivation, see [23].)

References

[1] R. E. Blahut. *Theory and Practice of Data Transmission Codes, 2nd Edition.* 1994.

[2] E. F. Casas and C. Leung. A simple digital fading simulator for mobile radio. IEEE Vehicular Technology Conference, June 1988.

[3] A. Dholakia. *Introduction to Convolutional Codes with Applications.* Kluwer Academic Publishers, Norwell, Massachusetts, 1994.

[4] ETSI TC-SMG. *European Digital Cellular Telecommunications System (Phase 2): Full Rate Speech Transcoding (GSM 06.10).* European Telecommunications Standards Institute, October 1993.

[5] ETSI TC-SMG. *European Digital Cellular Telecommunications System (Phase 2): Channel Coding (GSM 05.03).* European Telecommunications Standards Institute, October 1993.

[6] ETSI SMG2 Speech Experts Group. Comparison of the PCS1900 EFR codec against ETSI EFR requirements based on COMSAT test results. Ericsson, Nokia, Nortel. TDOC 33/95, Cambridge, UK, June 1995.

[7] ETSI SMG2 Speech Experts Group. Selection criteria for the enhanced full rate speech coding algorithm -- speech quality requirements. France Telecom, CNET. TDOC 54/95.

[8] J. Hagenauer. Rate-compatible punctured convolutional codes (RCPC codes) and their applications. *IEEE Transactions on Communications*, V. 36, No. 4, April 1988, pp. 389-400.

[9] W. C. Jakes. *Microwave Mobile Communications.* John Wiley and Sons, Inc., New York, 1974.

[10] W. P. LeBlanc. GSM-EFR: MOS results from preliminary test. Internal paper, Texas Instruments. December 1995.

[11] W. P. LeBlanc, C. N. Liu, and V. R. Viswanathan. An enhanced full rate speech coder for digital cellular applications. International Conference on Acoustics, Speech and Signal Processing. May 1996.

[12] L. H. C. Lee. New rate-compatible punctured convolutional codes for Viterbi decoding. *IEEE Transactions on Communications*, V. 42, No. 12, December 1994, pp. 3073-3079.

[13] W. C. Y. Lee. *Mobile Cellular Telecommunications Systems.* McGraw-Hill Book Company, New York, 1989.

[14] S. Lin and D. J. Costello. *Error Control Coding: Fundamentals and Applications.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[15] C. N. Liu, W. P. LeBlanc, V. R. Viswanathan. Error detection and error concealment of convolutionally encoded data. U.S. Patent application, TI-224444, submitted 1996. Robert L. Troike, *et al*, attorney.

[16] H. Lou. Implementing the Viterbi algorithm. *IEEE Signal Processing Magazine*, V. 12, No. 5, September 1995, pp. 42-52.

[17] W. H. Press, *et al. Numerical Recipes in C: The Art of Scientific Computing, 2nd Edition.* Cambridge University Press, New York, 1992.

[18] T. S. Rappaport. *Wireless Communications Principles and Practices.* Prentice-Hall, Inc., Upper Saddle River, New Jersey, 1996.

[19] N. Seshadri, C. W. Sundberg, V. Weerackody. Advanced techniques for modulation, error correction, channel equalization, and diversity. *AT&T Technical Journal*, July/ August 1993.

[20] A. S. Spanias. DSP and speech coding. Lecture notes, presented at Texas Instruments, Dallas, Texas, September 1995.

[21] R. Steele, ed. *Mobile Radio Communications.* Pentech Press, London, 1992.

[22] V. R. Viswanathan. A tutorial on speech coding. Internal report, Texas Instruments. 1995.

[23] S. B. Wicker. *Error Control Systems for Digital Communication and Storage.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1995.