1

Advanced Vocoder Idle Slot Exploitation for TIA IS-136 Standard

by
Ernest Nanjung Yeh

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degrees of

Bachelor of Science in Electrical Engineering and Computer Science
Master of Engineering in Electrical Engineering and Computer Science

at the
Massachusetts Institute of Technology

May 1998
[June 1998

Author..
Department of Electrical Engineering and Computer Science
May 26, 1998

Certified by..
G. David Forney, Jr.
Adjunct Professor of Electrical Engineering
Thesis Supervisor

Accepted by..
Arthur C. Smith
Chair, Department Committee on Graduate Students

Advanced Vocoder Idle Slot Exploitation for TIA IS-136 Standard

by
Ernest Nanjung Yeh

Submitted to the Department of Electrical Engineering and Computer Science
on June 26, 1998
in partial fulfillment of the requirements for the degrees of

Bachelor of Science in Electrical Engineering and Computer Science
Master of Engineering in Electrical Engineering and Computer Science

Abstract

BellSouth has proposed a modification to the IS-136 digital cellular standard, which they refer to as Advanced Vocoder Idle Slot Exploitation (ADVISE), in which base stations can transmit auxiliary coded (redundant) bits on otherwise unused time slots to assist certain subscriber units (SUs). In this master's thesis, we investigate two aspects of ADVISE: 1) adaptive redundancy schemes by which the auxiliary bits are generated, and 2) autonomous detection mechanisms by which the SUs detect the presence of such auxiliary bits without external signaling. The adaptive redundancy schemes are designed as an overlay to the existing IS-136 forward error correction (FEC) design. We propose and test a scheme which starts with a base code of a given rate and derives lower rate codes by adding FEC overlay. When the channel is favorable, for example, only the coded bits from the base code are sent; as the channel condition deteriorates, the FEC overlay bits are sent along with the existing coded bits to increase coding gain. On the other hand, we propose an autonomous detection method which exploits the fact that some of these FEC overlay bits are mere repetition of the standard slot bits. The detection method makes use of a distance-based metric which, after simplification, performs hypothesis testing based on the Hamming distance between the two sets of received bits. If the Hamming distance is smaller than a threshold, the detection method declares ADVISE to be present and allows the channel decoder to use the auxiliary data. Otherwise, the method declares ADVISE to be absent, and the channel decoder uses only the data received in the standard slot.

Thesis Supervisor:	Dr. G. David Forney, Jr.
Title:	Adjunct Professor of Electrical Engineering, MIT Laboratory for Information and Decision Systems
Industry Supervisor:	Dr. A. Roger Hammons
Title:	Principal Engineer, Hughes Network Systems

Acknowledgments

It has been an incredibly meaningful and fulfilling five years at MIT. I would like to extend my sincere gratitude to those who have made tremendous contributions to my intellectual development and my personal growth.

To Professor William Peake, for his guidance and sense of humor have been the highlights of my MIT experiences. To Dr. Roger Hammons, for his supervision over this thesis project. He has inspired me with his breadth of knowledge and his mind of inventiveness. To Professor David Forney, for his critique and encouragement throughout the revision of this thesis. He has given the thesis a magic touch.

To all my brothers and sisters in the Hong Kong Student Bible Study Group, especially to Kelvin, Sbwong, Oyip, Agnes and Belvin, for their friendship and prayer support.

To Sunnia, for her companionship throughout the days of joy and sorrow.

To Selina, my big sister, for being a good role model. To Teemwing, my little sister, for bringing laughter.

To my family, especially to my parents, Uncle Sam and Uncle Tony, for their sacrifice and firm support for my education. Mom and Dad, I *will* get my MD/PhD before you retire!

To my Lord Jesus Christ, for you are just awesome!

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The goal of the Advanced Vocoder Idle Slot Exploitation (ADVISE) project is to investigate first the optimal utilization of unused resources (idle channels) for voice quality improvement to specific subscriber units (SUs) on the downlink (base station to SU), and second, the best autonomous detection method by which the SUs can detect the presence or absence of ADVISE bits without explicit external signaling.

In IS-136, each radio frequency (RF) channel supports 3 full-rate users by time-division multiple access (TDMA). That is, each SU is assigned slots in a round-robin scheme (6 time slots per frame; 2 time slots per SU). Traditionally, as long as one user is active on any given RF channel, all other time slots are turned on with filler information at the same transmit power; therefore, the utilization of the idle slots does not add any additional interference to the system.

In general, the voice quality of the call is proportional to the bit rate sent to the SU. If it is possible to identify that a particular channel is idle, then extra speech information could be sent on this channel to the SU. Consider the following example illustrating how ADVISE might work in IS-136. Suppose in a given RF channel (with a capacity of 3 full-rate subscribers), there are only two subscribers, leaving the third time slots idle. One of the two subscribers, denoted here as Mobile1, experiences a poor quality radio link. The base station, after realizing that Mobile1 has a poor quality link, sends Mobile1 extra speech information on the idle time slots. Mobile1 combines all the speech information as inputs into its decoding process. The downlink error rate of this SU will thus be decreased, and the voice quality improved.

If some of this extra speech information is a simple repetition of that in the standard slot, then Mobile1 can autonomously detect the presence or absence of this extra information by comparing the two sets of data received. When Mobile1 decides such extra data are indeed available, it can proceed to use them in channel decoding. No handshaking occurs between the base station and Mobile1, and there is no latency due to message passing. Thus, the implementation of ADVISE can operate using the current IS-136 message protocol.

Chapter 2

Background

The concept of ADVISE is intended to be a modification to the existing IS-136 architecture. Our objective is to make as little perturbation as possible to the current system while designing ADVISE to best exploit the unused resources. In this chapter, we discuss the parts of the IS-136 standard that are relevant to the design and implementation of ADVISE.

2.1 IS-136 Standard: An Overview

IS-136 is a TDMA mobile digital cellular standard which specifies the use of a 7.4 kb/s algebraic-codebook code-excited linear predictive (ACELP) vocoder for speech source coding, a convolutional code (rate ½, K = 6) for channel coding, and quadrature phase-shift keying (QPSK) for modulation.

2.2 Ordering of the speech encoder bitstream

The existing IS-136 FEC scheme is diagrammed in Figure 2.1. It is an unequal error protection scheme in which the vocoder output bits are classified into three distinct categories. There are 48 Class 1A bits, which are protected by both a 7-bit cyclic redundancy check (CRC) and a K = 6, rate ½ convolutional code; 48 Class 1B bits, which are protected by the convolutional code alone; and 52 Class 2 bits, which are unprotected.

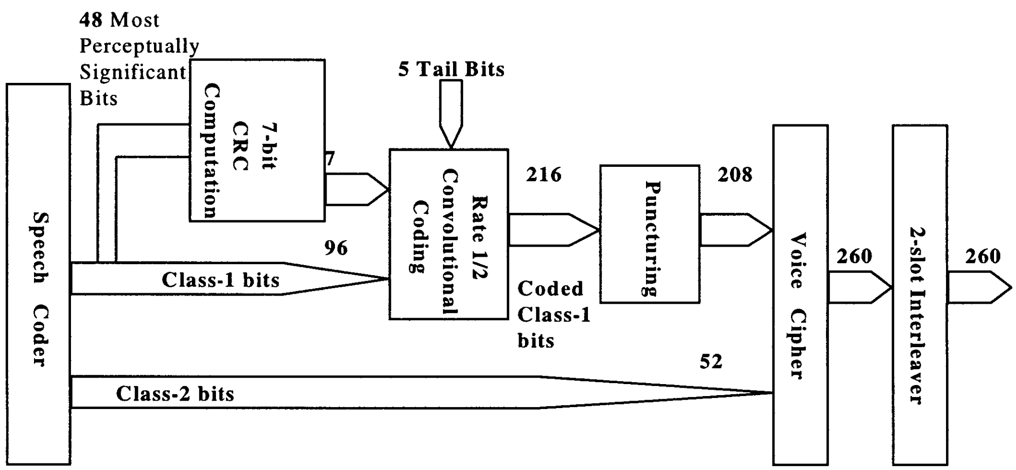

Figure 2.1 Flow Chart for IS-136 FEC

2.3 CRC for Error Detection

The 48 Class 1A bits are protected by seven parity bits used for error detection. The parity bits are calculated according to the CRC polynomial

$$g(x) = 1 + x + x^2 + x^4 + x^5 + x^7 .$$

At the transmitter, the 7-bit CRC is first computed. The Class 1A, CRC, Class 1B, and tail bits (48 + 7 + 48 + 5 = 108) are all encoded by a rate-½ convolutional encoder. The exact ordering of the input bit stream is shown in Figure 2.2.

At the receiver, channel decoding is performed. Based on the decoded Class 1A bits, another 7-bit CRC is calculated using the same CRC generating polynomial. This calculated CRC is compared to the decoded CRC. If the CRCs match, the receiver assumes no errors have occurred. Otherwise, a bad frame indicator is signaled to the speech decoder, and the speech decoder performs appropriate error concealment, such as attenuating the parameters or muting the output. This error detection method is illustrated in Figure 2.3.

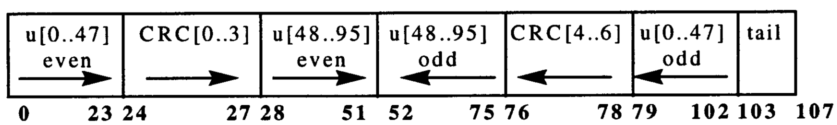

Figure 2.2 Input Bit Stream to the Convolutional Encoder

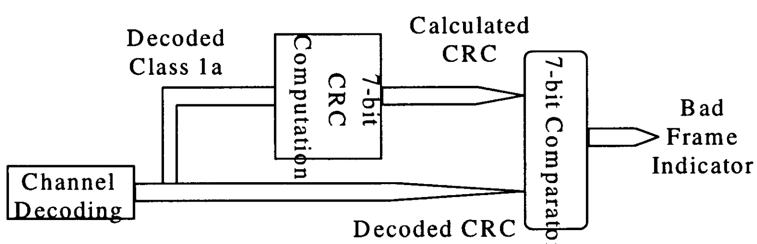

Figure 2.3 Error Detection and Bad Frame Indicator

2.4 Generators for the Rate-½ Convolutional Code

The rate-½ convolutional code used by IS-136 is the optimal nonsystematic K = 6 (32-state) code cited in standard texts (e.g., see Lin & Costello, Table 11.1, pp. 330-331), with generators

$$g_1(x) = x^5 + x^4 + x^2 + 1 \;; \qquad\qquad g_2(x) = x^5 + x^3 + x^2 + x + 1 \,.$$

An encoder for this code is illustrated in Figure 2.4. The d_{free} of this optimal rate-½ encoder is 8.

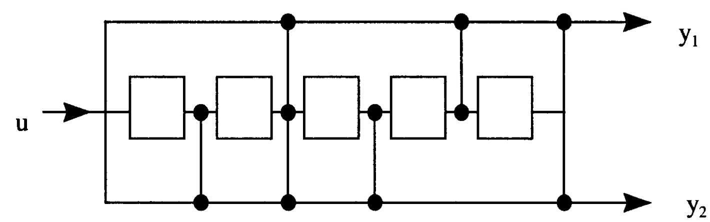

Figure 2.4 Encoder for Optimal Rate-½ Convolutional Code

2.5 Puncturing and Interleaving

The outputs of the convolutional encoder consist of 2 x 108 = 216 coded bits, of which 8 are punctured (deleted). The vocoder output is therefore represented by a total of 260 bits, corresponding to the 208 punctured coded bits and the 52 uncoded Class 2 bits.

$$Bits_{ch} = 2 \times (Class\ 1A + Class\ 1B + CRC + Tail) - PUNCT + Class\ 2 = 260,$$

where

$$Class\ 1A = 48,\ Class\ 1B = 48,\ Class\ 2 = 52,\ CRC = 7,\ Tail = 5, \text{ and } PUNCT = 8.$$

The pattern of puncturing, as illustrated in Figure 2.5, is arranged so that there is maximum separation between the punctured bits, to minimize the adverse effect due to puncturing.

The 260 bits are then passed to a two-slot interleaver in which the bits are rearranged in a 26-by-10 array. Of the 26 rows, the odd rows are sent during the current time frame, whereas the even rows are saved in a buffer to be transmitted during the next frame. This 2-slot interleaving scheme provides some time diversity so that in a slow fading channel, channel error events are delocalized. In most cases, the error correction ability of the convolutional code is enhanced, and the decoded word error rate is reduced. The 2-slot interleaver is illustrated in Figure 2.6.

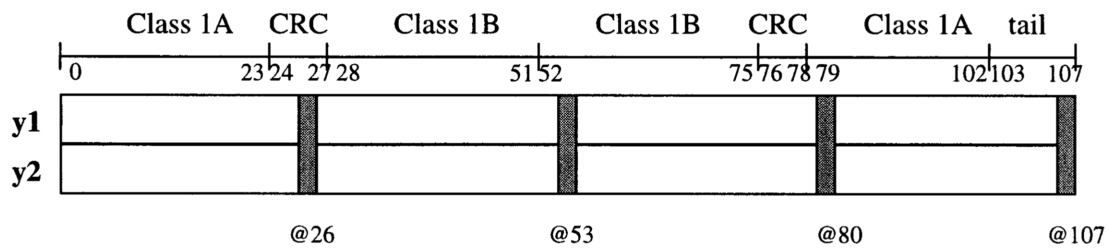

Figure 2.5 Encoder Output and Puncturing Pattern

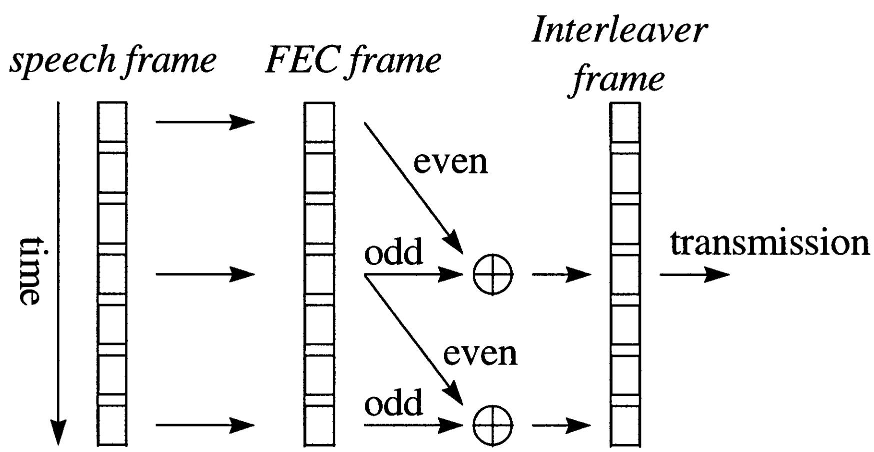

Figure 2.6 Two-Slot Interleaver

Chapter 3

Exploitation of Idle Slots

As discussed in previous chapters, the concept of ADVISE can be implemented when there is an idling time slot in the TDMA system. However, there are additional timing constraints which limit the window of time during which ADVISE can transmit. The width of this window is also handset-dependent, but we will leave that aspect to future detailed investigation.

3.1 Downlink and Uplink Timing

The IS-136 standard specifies a time-division structure of 40 ms/frame, or 6.7 ms/slot (six slots in one frame).

The uplink frame leads the downlink frame by 1¼ slot so that subscriber units will not have to employ duplexors. Suppose a full-rate subscriber unit uses slots 1 and 4. At the beginning of the uplink frame (slot 1), the subscriber unit tunes to the transmission frequency and subsequently transmits one slot worth of data. At the completion of transmission, it has ¼ of a slot to perform maintenance and to retune to the receiving frequency before receiving a slot of data from the downlink. After that, there will be approximately ¾ of a slot before the transmission during slot 4. During this period, the subscriber unit may be busy performing other maintenance functions. The exact length of time during which the subscriber unit is idle depends on the specifications of the handset provider. This overlapping of the uplink and downlink time slots is illustrated in Figure 3.1.

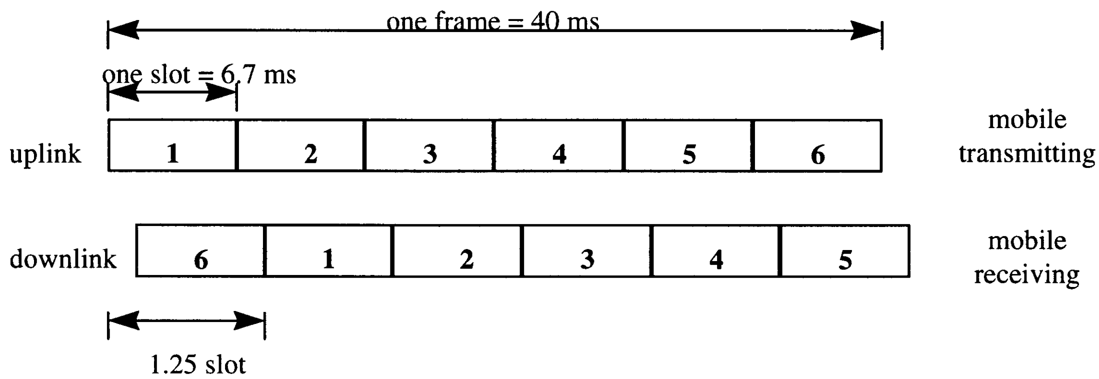

Figure 3.1 Uplink and Downlink Frame/Slot Structure

3.2 Number of Extra Bits Available

In the ADVISE concept, the base station is capable of making additional bits available to a particular subscriber unit in a neighboring idle time slot. Due to timing and other system requirements, the subscriber unit is believed to have access to up to 100 extra bits in the neighboring slot in the existing IS-136 mode of operation and up to a full slot (260 extra bits) in the Discontinuous Transmission (DTX) mode of operation under consideration for IS-136+.

3.2.1 A One-Hundred-Bit Solution

In IS-136, a frame can be divided into two subframes of three slots each, which simplifies the bookkeeping because the rules used in the second subframe are merely a repeat of those for the first subframe. In each subframe, there are two windows of time during which the mobile is neither transmitting or receiving. The first window starts at the end of uplink slot 1 and lasts until the beginning of downlink slot 1. The second window starts at the end of downlink slot 1 and lasts until the beginning of uplink slot 4. The first window lasts only ¼ of a slot, and is too short for ADVISE. However, the second window, about ¾ of a slot, can be used to send extra information to the subscriber unit (SU). According to the data provided by the SU manufacturer Nokia, approximately 100 channel bits can be sent during the second window.

3.2.2 A Full-Slot Solution

If the subscriber unit employs DTX on the uplink, then the subscriber unit will turn off transmission if there is a silence in the mobile user's speech. Once the mobile user speaks again, the uplink transmission will be turned on. When DTX is used, there is a potential of extending the second window because now the mobile station does not need to transmit during the uplink slot 4. Thus a full slot (260 channel bits) worth of extra information can be sent.

Chapter 4

FEC Overlay

The availability of the ADVISE slot allows us to adopt a lower-rate code. In this chapter, we propose and test different lower-rate convolutional codes. The objective is to achieve maximum coding gain while minimizing decoder complexity. Compatibility with existing systems is highly desirable.

4.1 Adaptive Rate-Compatible Convolutional Code

We investigate a variable-rate convolutional coding scheme in which the encoder is capable of generating a code of rate ½, 1/3 or ¼. In such an encoder, the first two connection polynomials are chosen to be the same as the two specified by IS-136. The third and fourth connection polynomials are chosen so that the rate 1/3 and rate ¼ codes have the best d_{free}, respectively. When the channel is good and ADVISE is not used, both the third and fourth coded bits will be punctured, so the resultant bit stream is identical (hence backward compatible) to the current IS-136. When the channel deteriorates and the 100 ADVISE bits are used for extra protection, the base station will transmit the third coded bits in the idle slot, lowering the overall coding rate to 1/3. If all 260 extra bits are available, all four coded bits will be transmitted, making the rate ¼ . The concept is illustrated in Figure 4.1.

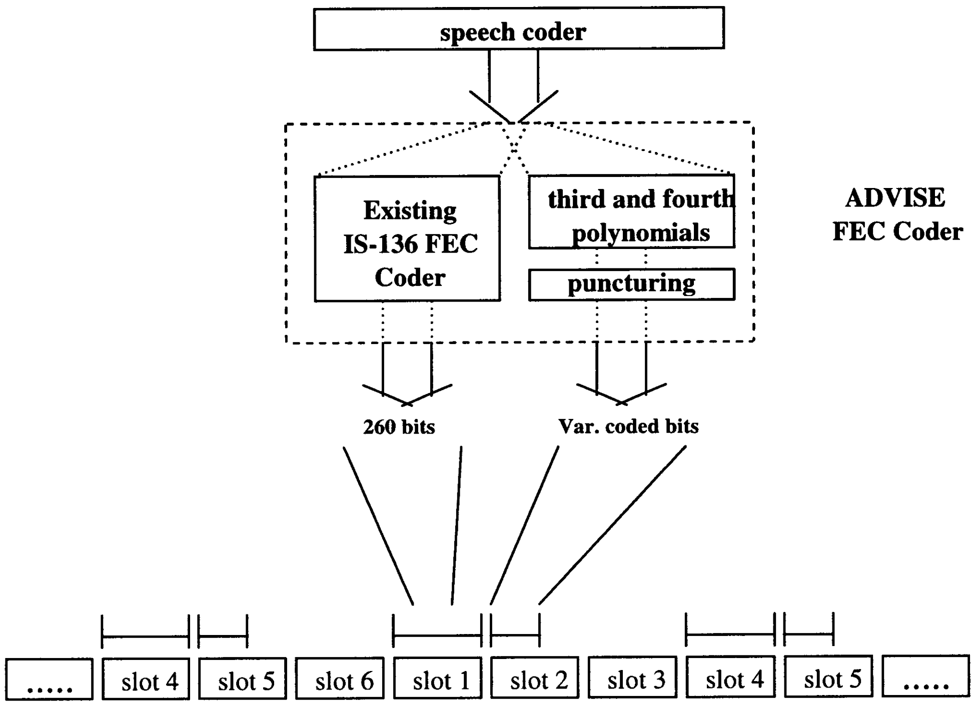

Figure 4.1 Rate-Compatible FEC Overlay

4.1.1 Rate-1/3 Convolutional Code

In the 100-bit ADVISE scenario, we propose to add a third connection polynomial to create a rate-1/3 code. Although not obvious, it is possible to augment the IS-136 rate-½ code to an *optimal* rate-1/3 code. Lin and Costello give the following optimal K = 6, rate-1/3 convolutional code generators:

$$g_1(x) = x^5 + x^3 + x + 1\ ;\ g_2(x) = x^5 + x^4 + x^3 + x^2 + 1;\ g_3(x) = x^5 + x^2 + x + 1.$$

We note that the first two of these generators are the reverse of the polynomials generating the rate-½ code. Since reverse polynomials generate a code with the same weight distribution, an optimal augmentation of the IS-136 code is achieved by using the reverse of the third polynomial. Therefore, for IS-136, we propose the use of the rate-1/3 code with generators:

$$g_1(x) = x^5 + x^4 + x^2 + 1 \; ; \; g_2(x) = x^5 + x^3 + x^2 + x + 1; \; g_3(x) = x^5 + x^4 + x^3 + 1 \, .$$

The d_{free} for this optimal rate-1/3 code is 13.

4.1.2 Rate-¼ Convolutional Code

For the rate-¼ version, we add the following additional connection polynomial:

$$g_4(x) = x^5 + x^4 + x^3 + x + 1 \, .$$

This rate-¼ code has $d_{free} = 18$, whereas the optimal rate-1/4 code has $d_{free} = 19$. We are unable to have an optimal encoder for rate-¼ because the first three generators restrict our choice. The trade-off is made in order to preserve backward compatibility.

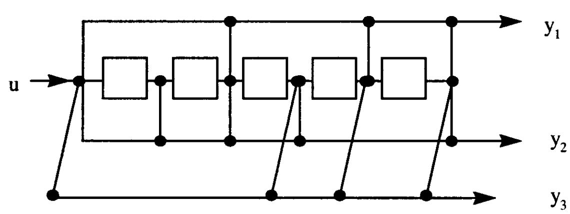

Figure 4.2a *Optimal* Rate-1/3 Encoder

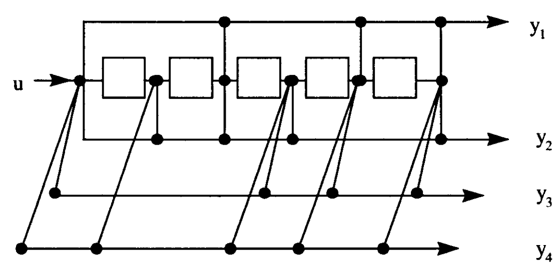

Figure 4.2b Rate-¼ Encoder

4.1.3 A Mix-and-Match Approach

The rate-compatible coding proposal above provides extra protection to the already protected bits, namely the Class 1 bits. The Class 2 bits are unprotected in the existing IS-136 FEC scheme. These bits are perceptually less important and do not play a significant role in the overall voice quality under normal channel conditions. However, when the channel deteriorates to the point that ADVISE is needed, badly corrupted Class 2 bits will adversely affect quality. An optimal allocation of the extra coded bits, therefore, is not to use them for Class 1 protection only. Rather, we investigate schemes which mix and match the extra check bits for both Class 2 and Class 1 protection.

4.1.4 Simple Full-Slot Repeat

When an extra full slot is available, one can merely repeat the original coded bits in the ADVISE slot. Despite its simplicity, the full-slot repeat provides significant performance improvement because of the 3 dB increase in signal energy obtained by sending twice as many channel bits. Implementation of slot repeat decoding is also straightforward. We can make use of the second set of data by using an antenna-diversity combining algorithm. Alternatively, we can optimally combine the extra set of data in the branch metric computation as though the code was a rate-¼ convolutional code, where the third and fourth generators are a repeat of the first and second. Both decoding methods are optimal; the former is slightly less computationally burdensome.

When we decode a simple slot-repeat using a rate-1/4 Viterbi algorithm, the effective d_{free} is 16, twice of the d_{free} of the original optimal rate-1/2 code. If we used the rate-1/4 code recommended in Section 4.1.2, then we would have $d_{free} = 18$ instead. In return, we would have to support a more complex decoding scheme. The trade-off between more complex encoding versus better performance, as in this case, is not entirely clear even after our simulations. We conclude that a simple full-slot repeat remains as an attractive candidate in comparison to other schemes. Table 4.1 summarizes the d_{free} of the proposed encoders.

Encoder Rate	d_{free}
rate-½	8
rate-1/3	13
rate-¼	18
full-slot repeat	16

Table 4.1 d_{free} Summary

4.1.5 A Split-and-Merge Trellis

If a full ADVISE slot repeat is available, we can achieve better voice quality by protecting the currently unprotected Class 2 bits. In addition, for the third and fourth generators, we can use the generators as proposed in Section 4.1.2. The basic scheme is illustrated in Figure 4.3. In this case we substitute the Class 2 bits for the Class 1B bits and re-encode the Class 1A + Class 2 bits for transmission in the ADVISE slot. The Class 1B bits are sent uncoded in the ADVISE slot.

In the Viterbi decoder, we optimally combine the received data from both transmissions to decode the various bits. This requires separate Viterbi decoding steps for the Class 1B and Class 2 bits. For the Class 1A bits, we effectively have rate-¼ coding. For the Class 1B and Class 2 bits, we effectively have systematic rate-1/3 coding. In order to optimally combine received data for the Class 1A bits at the end of the frame, we use the five Class 1B bits in both endings to ensure that the final encoder state is the same for both transmissions.

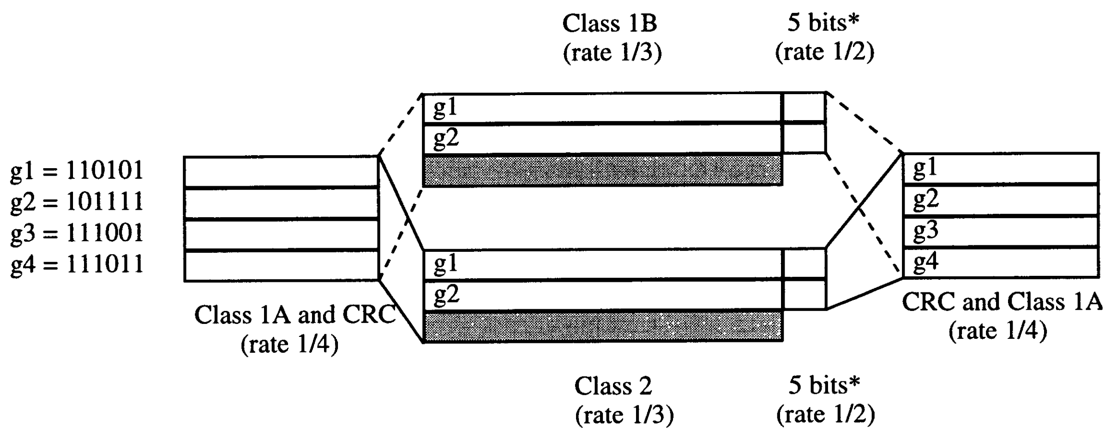

Figure 4.3 Split-and-Merge Trellis (systematic bits are shown shaded)

In Figure 4.4, we further improve upon this idea by using nonsystematic rate- 1/3 encoding for the Class 1B bits. That is, in the ADVISE slot, we do not send the Class 1B bits uncoded but perform additional coding using the optimal generators for the rate-1/3 code described earlier.

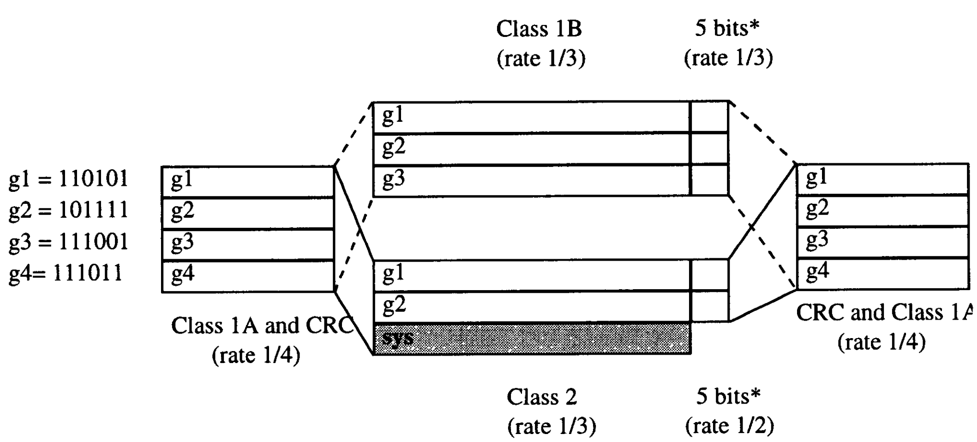

Figure 4.4 Optimal Rate-1/3 Code for Class 1B and Systematic Rate 1/3 Code for Class 2

4.2 Viterbi Decoders

4.2.1 Existing Viterbi Decoding Algorithm for Rate-½ Code

Given soft decisions from the receiver front end, the Viterbi decoder optimally uses the data to decode the various bits. The Viterbi decoder for the rate-½ code ideally computes branch metrics of the following squared-Euclidean distance form:

$$\mu(\lambda;b)=\sum_{i=1}^{2}\left|r_i\left(t_{\lambda,i}\right) - \alpha_i\left(t_{\lambda,i}\right)s_i(b)\right|^2$$

where:

- $\mu(\lambda;b)$ denotes the branch metric computed for branch b at trellis stage λ;
- $(r_1(t),r_2(t))$ denotes the received signals at time t corresponding to the transmitted code bits;
- $(s_1(b),s_2(b))$ denotes the ideal modulated signals corresponding to the coded bits for branch *b*;
- $\alpha_i(t)$ denotes the channel fade coefficient at time *t*;
- $t_{\lambda,i}$ denotes the transmission time associated with the i-th coded bit at the λ-th trellis stage.

Note that the times $t_{\lambda,i}$ are not necessarily uniformly spaced from one trellis stage λ to the next, or from one coded bit *i* to the next, due to interleaving. We note that there are well-known simplifications of the Viterbi branch metric that can be performed to reduce implementation complexity, so this form of metric is used primarily to illustrate the concept. In the event of puncturing, the decoder sets $r_i(t)$ to a neutral value favoring neither a 0-bit nor a 1-bit; e.g., for BPSK or QPSK, we set $r_i(t)$ = 0. The channel fade coefficient $\alpha_i(t)$ is usually not known at the receiver. In this case, the receiver may form an estimate of the fade coefficient and use the estimate in the branch metric calculations (decoding with channel state information) or may ignore fading altogether and set $\alpha_i(t) = 1$ in the branch metric calculations (decoding without channel state information).

4.2.2 Viterbi Decoder for Rate-1/3 and Rate-1/4 Codes

With minor modifications, the Viterbi decoding subroutine can be used "as is" from the existing system. In case of the augmented rate-1/3 code, the ideal branch metrics take the following form:

$$\mu(\lambda; b) = \sum_{i=1}^{3} \left| r_i(t_{\lambda,i}) - \alpha_i(t_{\lambda,i}) s_i(b) \right|^2 .$$

Therefore, after an additional term $\left| r_3(t_{\lambda,3}) - \alpha_3(t_{\lambda,3}) s_3(b) \right|^2$ is added to the branch metrics of the rate-½ code, the rest of the Viterbi decoder is the same. (If selected Class 2 bits are to be encoded, the number of trellis stages will also be increased somewhat.) This results in a negligible increase in decoder complexity.

Similarly, a rate-¼ Viterbi decoder adds a fourth term $\left| r_4(t_{\lambda,4}) - \alpha_4(t_{\lambda,4}) s_4(b) \right|^2$ to the branch metric computation, but otherwise is the same.

4.3 Simulation Results

As we lower the rate of the encoder, the corresponding d_{free} increases and therefore the CRC failure decreases. We perform simulations in two fading environments (slow-fading and fast-fading) to investigate the incremental improvement between each successive coding rate.

4.3.1 Slow Rayleigh Fading Channel

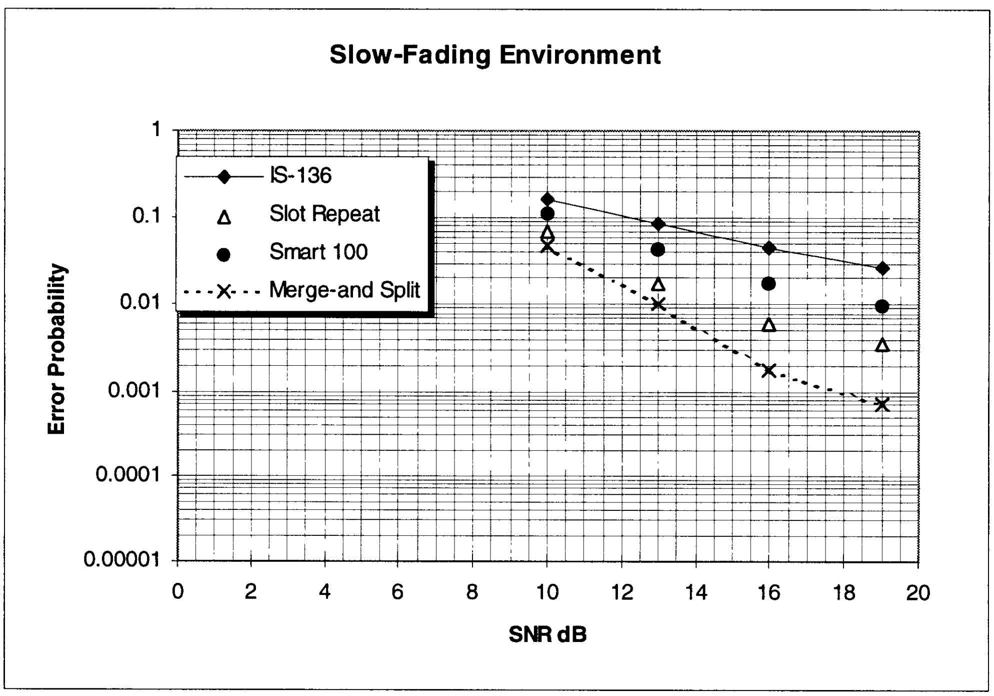

Figure 4.5 Comparison of Different Schemes in a Slow-Fading Environment.

As shown in Figure 4.5, we observe the performance curves improve as we lower the coding rate. The simulation results are consistent with our expectation. In this slow-fading environment, the performance curves are relatively flat. Improvement due to lowering the coding rate is insignificant. We conclude that ADVISE is not effective in such a slow-fading environment.

4.3.2 Fast Rayleigh Fading Channel

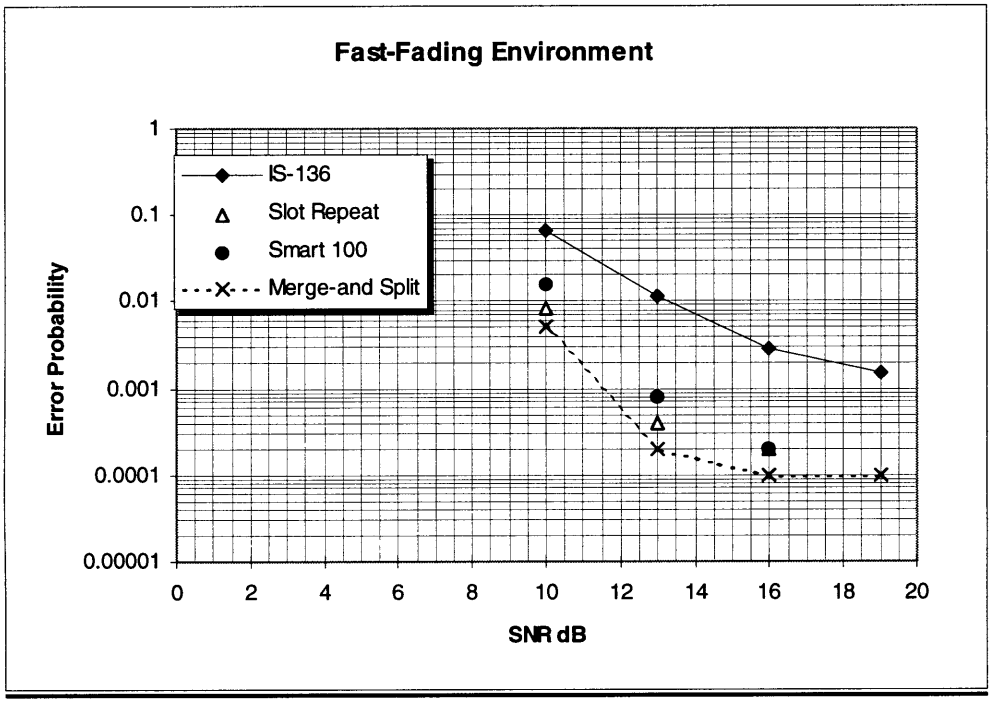

Figure 4.6 Comparison of Different Schemes in a Fast-Fading Environment.

In a fast-fading environment, as shown in Figure 4.6, ADVISE proves to be much more effective. The most significant improvement occurs between the original IS-136 (rate-½) and the Smart 100 (rate-1/3). The subsequent incremental improvements (i.e., between rate-1/3 to full-slot repeat and between full-slot repeat to merge-and-split) are not as significant.

4.3.2 Performance of Rate-1/4 versus Full-Slot Repeat

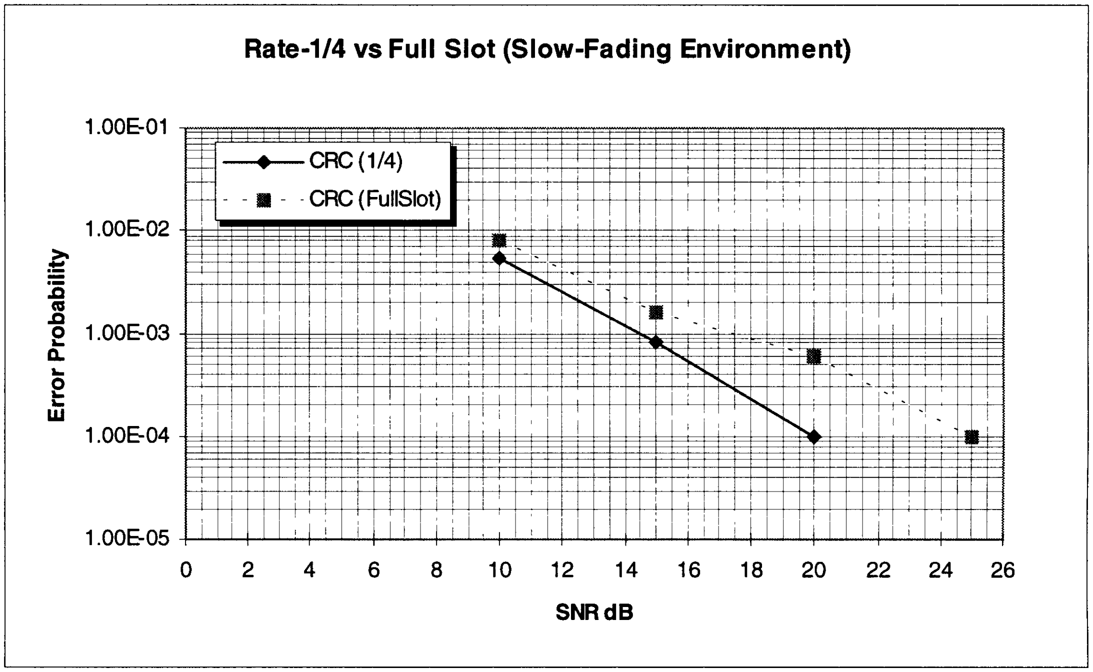

Figure 4.7 Rate-¼ Coding vs Full-Slot Repeat in a Slow-Fading Environment

(Note: Rate-¼ Coding and Full-Slot Repeat have virtually indistinguishable results in a Fast-Fading Environment.)

Further, we are interested to investigate the performance difference between full-slot repeat and the proposed rate-¼ code. As stated in previous sections, full-slot repeat has a smaller d_{free} but is less complex in decoding compared to the rate-¼ code. In a slow-fading environment, the full-slot repeat and the rate-1/4 code have virtually indistinguishable performance curves. In a fast-fading environment, as shown in Figure 4.7, the performance difference is marginal, which strongly suggests that the full-slot repeat solution is very attractive despite the minor trade-off of a smaller d_{free}.

Chapter 5

Autonomous ADVISE Detection Algorithm

The BellSouth proposal does not specify the signaling mechanism by which the base stations inform the SUs that auxiliary ADVISE bits are available. In this chapter, we describe a method by which the SUs can autonomously detect the presence or absence of auxiliary bits without explicit signaling.

5.1 Autonomous Detection Algorithm: An Overview

In our blind detection method, we assume that the auxiliary slot contains a certain number (approximately 50) of bits that are identical with those transmitted on the standard slot. At the mobile end, the receiver makes preliminary hard decisions regarding the channel bits that are repeated on both the standard and auxiliary (ADVISE) slots and counts the number of decisions that are the same for both slots, i.e., the Hamming distance. The set of repeated channel bits may be arbitrarily chosen from the coded bits (Class 1) and/or the uncoded bits (Class 2). If the Hamming distance is less than a threshold, the method declares ADVISE to be present and allows the channel decoder to use the auxiliary data. Otherwise, the method declares ADVISE to be absent, and the channel decoder uses only the data received in the standard slot. The threshold may be a predetermined constant or selected adaptively to maximize performance.

The proposed algorithm serves as a blind detection mechanism for any ADVISE FEC overlay in which some number of coded or uncoded bits are transmitted in an identical fashion on both the normal and the auxiliary slots (i.e., the same channel symbols are transmitted but in perhaps a different order). With only a modest increase in decoder and handset complexity, the method allows the utilization of the auxiliary bits (if available) without any external signaling or handshaking between the SU and base station.

Since the method can be applied on a slot-by-slot basis, it provides the base station greater flexibility in switching ADVISE on and off without the latency associated with traditional message passing. By employing the autonomous blind detection method, no modification is needed in the existing message standard, and backward compatibility is achieved.

5.2 Two-Stage Viterbi Decoding

An alternative non-distance-based method of autonomous blind detection is to perform two stages of decoding as depicted in Figure 5.1. In the first stage, the decoder assumes that the ADVISE bits are present and uses the data received on the auxiliary slot (by combining the auxiliary data with the normal slot data via maximal-ratio combining if the coded bits are identical on the two slots, or by incorporating both sets of data into the branch metric computations if the coded bits are different). At the completion of the first

Viterbi decoding, a cyclic redundancy check (CRC) is performed on the decoded word. If the CRC check fails, the decoder discards the ADVISE bits and performs a second stage of Viterbi decoding in which only the data from the normal slot are used in the branch metric computations. As is clear from Figure 5.1, the two-stage non-distance-based method disadvantageously requires a significant increase in decoder processing requirements and latency as compared with the proposed distance-based method. On the other hand, the distance-based method, imposes a constraint that a set of channel bits must be identical in the two slots, which reduces the flexibility of the encoder.

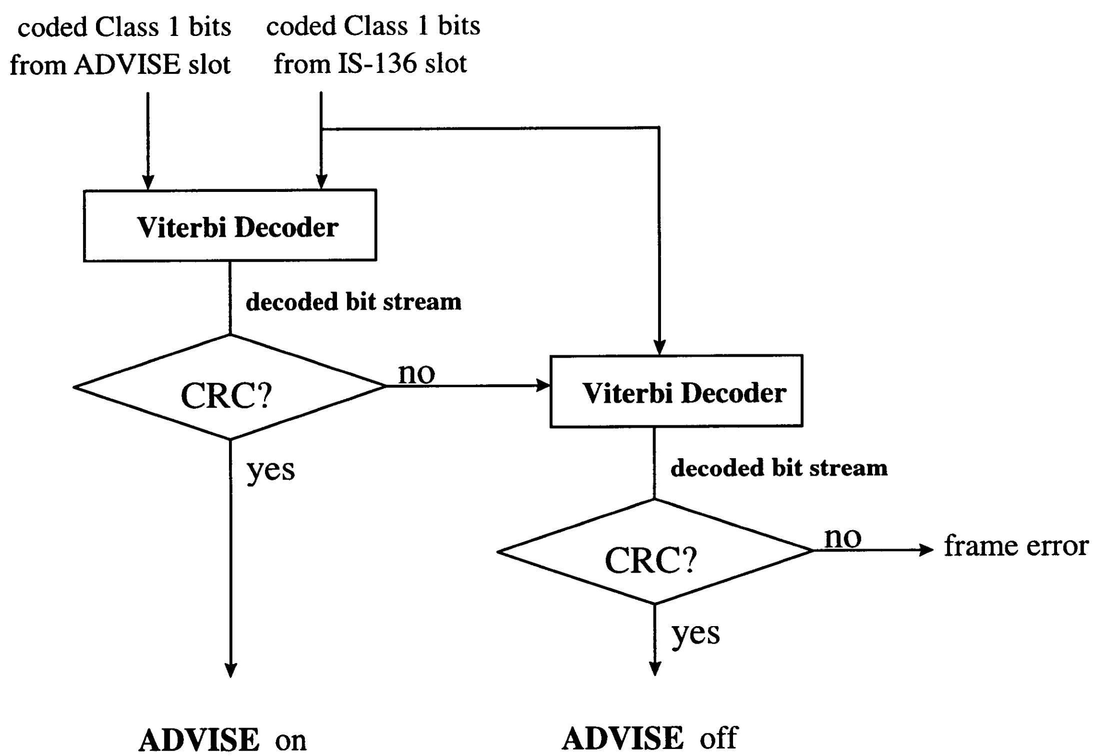

Figure 5.1 Two-stage Viterbi Decoding

5.3 Distance-based Metrics Algorithm

In the absence of explicit signaling to inform the SU that ADVISE is on or off, the SU needs to make such a determination autonomously. The ADVISE bits need not be transmitted in the same sequence as the original bit stream as long as the SU has prior knowledge of their reordering.

The distance metric can be adapted to suit implementation requirements. Useful metrics include squared Euclidean distance and Hamming distance. As is well known, distance metrics may be manipulated into many equivalent forms with corresponding modifications of the comparison rule. For example, before comparison, the metric value may be processed by applying an arbitrary monotonic increasing or montonic decreasing function. Common examples include taking the logarithm (often applied to reduce dynamic range requirements) and linear or affine transformations (to place metric values in a desired range of values).

5.3.1 Euclidean-Distance Metric

The proposed Euclidean-distance algorithm is shown in general form in Figure 5.2. The SU extracts the soft-decision values of the common bits on the auxiliary slot as one data vector and the soft-decision values of the same bits as received on the standard slot as a second data vector. The distance between the two data vectors is computed and compared to a detection threshold which is SNR-dependent. For fading channels, SNR estimation must also include estimation of the channel fading parameters $\alpha_i(t)$. If the distance between the vectors is less than the detection threshold, the receiver declares that useful ADVISE data are available and proceeds to use the ADVISE data in its decoding process. If the distance between the data vectors is large, indicating that the supposedly common bits are not the same, the receiver declares that ADVISE data are not available and proceeds to use its standard IS-136 decoder without processing the auxiliary data.

Maximum-likelihood statistical decision theory can be used to derive optimal rules for deciding between two hypotheses. For the autonomous detection algorithm, let

$$r_i(t) = \alpha_i(t)s_i(t) + n_i(t)$$

denote the complex baseband received signals for the common portion of the two slots, i = 1,2 (standard slot and auxiliary ADVISE slot, respectively). Here $s_i(t)$ denotes the transmitted QPSK symbol, $\alpha_i(t)$ denotes the fading coefficient, and $n_i(t)$ denotes the complex AWGN whose real and imaginary components each have variance σ^2. The receiver is assumed to estimate the $\alpha_i(t)$ and correct for the phase shift introduced by the fading channel. Ideally,

$$r_i'(t) = \frac{\alpha_i^*(t)r_i(t)}{|\alpha_i(t)|}$$

$$r_i'(t) = \alpha_i'(t)s_i(t) + n_i'(t)$$

where $\alpha_i'(t) = |\alpha_i(t)|$ and $n_i'(t)$ is again AWGN with real and imaginary components having variance σ^2.

To simplify notation, we drop the primes and use the following signal model:

$$r_i(t) = \alpha_i(t)s_i(t) \;+\; n_i(t)$$

where $r_i(t)$, $s_i(t)$ and $n_i(t)$ are complex, but $\alpha_i(t)$ is real. Taking L complex samples of $r_i(t)$ at multiples of the sampling interval T, we form a real-valued vector of dimension $2L$:

$$\bar{r}_i = \left[r_i^I(0), r_i^Q(0), r_i^I(T), r_i^Q(T), \ldots, r_i^I((L-1)T), r_i^Q((L-1)T)\right]$$

where $r_i^I(kT)$ and $r_i^Q(kT)$ are the real and imaginary (in-phase and quadrature) components of r_i at time $t = kT$. Similarly,

$$\bar{s}_i = \left[s_i^I(0), s_i^Q(0), s_i^I(T), s_i^Q(T), \ldots, s_i^I((L-1)T), s_i^Q((L-1)T)\right]$$

$$\bar{\alpha}_i = \left[\alpha_i(0), \alpha_i(T), \ldots, \alpha_i((L-1)T)\right].$$

Let H_0 denote the hypothesis that ADVISE is present; i.e., $\bar{s}_1 = \bar{s}_2$. The alternative hypothesis H_1 is that ADVISE is absent; i.e., $\bar{s}_1 \neq \bar{s}_2$.

A generalized likelihood ratio is given by the ratio of conditional probabilities averaged over the unknown signal pairs $(\bar{s}_1, \bar{s}_2)$:

$$\Lambda(\bar{r}_1, \bar{r}_2 \mid \bar{\alpha}_1, \bar{\alpha}_2) = \frac{\left\langle P(\bar{r}_1, \bar{r}_2 \mid \bar{s}_1, \bar{s}_2, \bar{\alpha}_1, \bar{\alpha}_2)\right\rangle_{\bar{s}_1 \neq \bar{s}_2}}{\left\langle P(\bar{r}_1, \bar{r}_2 \mid \bar{s}_1, \bar{s}_2, \bar{\alpha}_1, \bar{\alpha}_2)\right\rangle_{\bar{s}_1 = \bar{s}_2}}$$

where the angle brackets $\langle \cdot \rangle$ denote averaging with respect to the sequence $(\bar{s}_1, \bar{s}_2)$ satisfying the constraint.

In decision theory, assuming equal costs, hypothesis H_0 is accepted when $\Lambda \leq 1$. Since the conditional probability density functions are Gaussian, we have

$$P\left(\overline{r_1}, \overline{r_2} \mid \overline{s_1}, \overline{s_2}, \overline{\alpha_1}, \overline{\alpha_2}\right)$$
$$= C \cdot \exp\left\{-\frac{1}{2\sigma^2}\sum_k \left[r_1(kT) - \alpha_1(kT)s_1(kT)\right]^2\right\} \cdot \exp\left\{-\frac{1}{2\sigma^2}\sum_k \left[r_2(kT) - \alpha_2(kT)s_2(kT)\right]^2\right\}$$

where C is a constant.

After canceling common terms, Λ is proportional to the ratio

$$\frac{\left\langle \exp\left\{\frac{1}{\sigma^2}\sum_k \sum_{i=1}^{2}\left[r_i^I(kT)\alpha_i(kT)s_i^I(kT) + r_i^Q(kT)\alpha_i(kT)s_i^Q(kT)\right]\right\}\right\rangle_{\overline{s_1} \neq \overline{s_2}}}{\left\langle \exp\left\{\frac{1}{\sigma^2}\sum_k \left[s_1^I(kT)\sum_{i=1}^{2} r_i^I(kT)\alpha_i(kT) + s_1^Q(kT)\sum_{i=1}^{2} r_i^Q(kT)\alpha_i(kT)\right]\right\}\right\rangle_{\overline{s_1} = \overline{s_2}}}.$$

For QPSK, s_i^I and s_i^Q are proportional to ± 1. Hence, the dominant term in the numerator occurs when

$$s_i^I(kT) = \operatorname{sgn}\left\{r_i^I(kT)\alpha_i(kT)\right\}$$

and

$$s_i^Q(kT) = \operatorname{sgn}\left\{r_i^Q(kT)\alpha_i(kT)\right\}.$$

The dominant term in the denominator occurs when

$$s_1^I(kT) = s_2^I(kT) = \operatorname{sgn}\left\{\sum_{i=1}^{2} r_i^I(kT)\alpha_i(kT)\right\}$$

and

$$s_1^Q(kT) = s_2^Q(kT) = \operatorname{sgn}\left\{\sum_{i=1}^{2} r_i^Q(kT)\alpha_i(kT)\right\}.$$

Therefore, the dominant term in the numerator is

$$\exp\left\{\frac{1}{\sigma^2}\sum_k \sum_{i=1}^{2}\left[\left|r_i^I(kT)\alpha_i(kT)\right| + \left|r_i^Q(kT)\alpha_i(kT)\right|\right]\right\}$$

and in the denominator, it is

$$\exp\left\{\frac{1}{\sigma^2}\sum_k\left[\left|\sum_{i=1}^{2} r_i^I(kT)\alpha_i(kT)\right|+\left|\sum_{i=1}^{2} r_i^Q(kT)\alpha_i(kT)\right|\right]\right\}.$$

After taking logarithms, we have the approximate maximum likelihood test in which the distance-like measure

$$D=\sum_{k=0}^{L-1}\left\{\sum_{i=1}^{2}\left|r_i^I(kT)\alpha_i(kT)\right|-\left|\sum_{i=1}^{2} r_i^I(kT)\alpha_i(kT)\right|+\sum_{i=1}^{2}\left|r_i^Q(kT)\alpha(kT)\right|-\left|\sum_{i=1}^{2} r_i^Q(kT)\alpha_i(kT)\right|\right\}$$

is compared to a threshold λ. If $D < \lambda$, then H_0 is accepted. In an actual implementation, the receiver would use estimates $\tilde{\alpha}_i$ of the α_i. Also, more complex tests could be derived by including additional terms other than the dominant ones identified above.

In this formulation, the threshold should ideally be a function of the effective SNR after fading since one would usually prefer to have approximately constant false-alarm rate (CFAR) operation. We note that, in the absence of noise, the term

$$\sum_{i=1}^{2}\left|r_i^I(kT)\alpha_i(kT)\right|-\left|\sum_{i=1}^{2} r_i^I(kT)\alpha_i(kT)\right|$$

has minimum value 0 and maximum value $2\sqrt{E_b}\,\min\{\alpha_1^2(kT),\alpha_2^2(kT)\}$. With noise, at high SNR, the standard deviation of the above term is approximately 2σ. Hence, to achieve approximately CFAR operation, we propose the following threshold

$$\lambda=\mu\sqrt{S\tilde{N}R}\sum_{k=0}^{L-1}\min\{\tilde{\alpha}_1^2(kT),\tilde{\alpha}_2^2(kT)\},$$

where μ is a constant multiplier chosen to meet desired operating points with respect to missed detection and false-alarm rates, and the $\tilde{\alpha}_i$ and $S\tilde{N}R$ are the receiver's best estimates of the channel fading and signal-to-noise ratio, respectively. Depending on implementation constraints and specific operating conditions, other functions of the $\tilde{\alpha}_i$ and $S\tilde{N}R$ would also be possible.

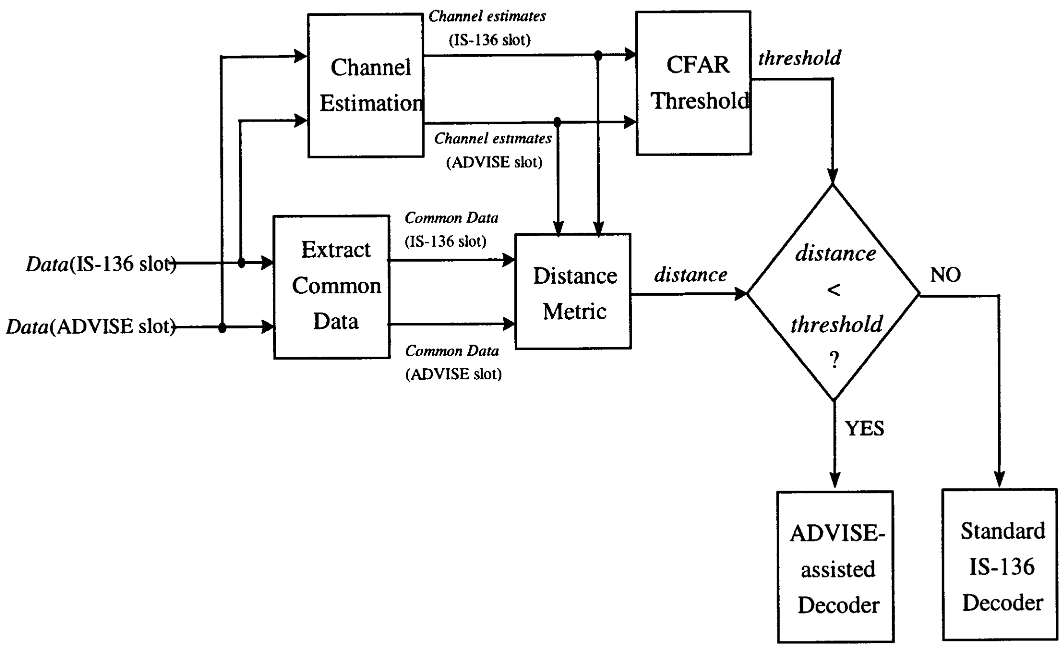

Figure 5.2 Distance-based Detection Algorithm

5.3.2 Hamming-Distance Metric

ADVISE pre-processors derived from squared Euclidean-distance metrics can be expected to provide optimal performance under AWGN conditions but, because of the real-valued multiplications involved in the metric computations and the necessity of adapting the threshold using constant false-alarm rate (CFAR) techniques, may be too computationally burdensome to be attractive for all applications.

The Hamming-distance metric involves making hard decisions as to the bit values and then counting disagreements between the two hard-decision data vectors. In fact, the measure D from the previous section is closely related to the Hamming-distance metric.

$$D = \sum_{k=0}^{L-1}\left\{\sum_{i=1}^{2}\left|r_i^I(kT)\alpha_i(kT)\right| - \left|\sum_{i=1}^{2} r_i^I(kT)\alpha_i(kT)\right| + \sum_{i=1}^{2}\left|r_i^Q(kT)\alpha(kT)\right| - \left|\sum_{i=1}^{2} r_i^Q(kT)\alpha_i(kT)\right|\right\}$$

By making hard decisions on the $r_i^I(kT)\alpha_i(kT)$ and $r_i^Q(kT)\alpha_i(kT)$ before computing D, the result is twice the Hamming distance. Hamming distance has the advantage that it is independent of SNR and the fading amplitude $|\alpha_i(t)|$, so setting the detection threshold is a simpler matter. The receiver makes a favorable decision with regard to similarity of the two data vectors only when the number of disagreements is small. Equivalently, the receiver can count the number of *agreements* and declare the two data vectors to be similar when the number of agreements is *large*. A constant threshold λ is appropriate to be effective with this decision statistic.

Pre-processors derived from Hamming-distance metrics would be less powerful than the Euclidean-distance counterparts, but the simplicity of their implementation (involving only binary logic operations) and their robustness to varying channel conditions may make them particularly attractive when processor loading is a major concern and/or channel estimates of adequate quality are not readily available. Operation of ADVISE within a handset is likely to be such an application.

A proposed design of the Hamming-distance type is shown in Figure 6.4. Because the Hamming-distance metric is simple to compute, we assume the handset has sufficient computational power to make decisions on a frame-by-frame basis. However, in practice, ADVISE will persist for many slots. We can easily add some memory to a frame-by-frame detection method.

For illustrative purposes, we consider the case in which the 52 Class 2 bits are repeated on the ADVISE slot. For detection, we require N of the 52 bits to produce agreements for ADVISE to be detected. In general, the threshold N can be made adaptive to the channel condition. We first consider a non-adaptive implementation in which a constant threshold is used. Then we consider a simple adaptive implementation that is well suited to the IS-136 application.

5.3.3 Non-adaptive Hamming-Distance Threshold

For a non-adaptive implementation, we propose a nominal threshold value of $N = 35$. We note that, by lowering the value of N, we increase the ADVISE detection rate but also increase the false-alarm rate. Conversely, by increasing the value of N, we decrease both the detection rate and the false-alarm rate. A low detection rate is undesirable from the system point of view since the ADVISE bandwidth is wasted when the targeted SU is unable to make use of the auxiliary bits. A high false-alarm rate is undesirable in that it degrades the performance of the already disadvantaged SU: false ADVISE detections will usually result in loss of the burst since the Viterbi decoder is unlikely to produce a valid codeword that passes the CRC test. The relative costs of these two different error events depend upon the application, so the setting of an optimal threshold will depend on detailed system trade-offs. The values of N reported here are offered as nominal values, expected to be roughly correct for the IS-136 application.

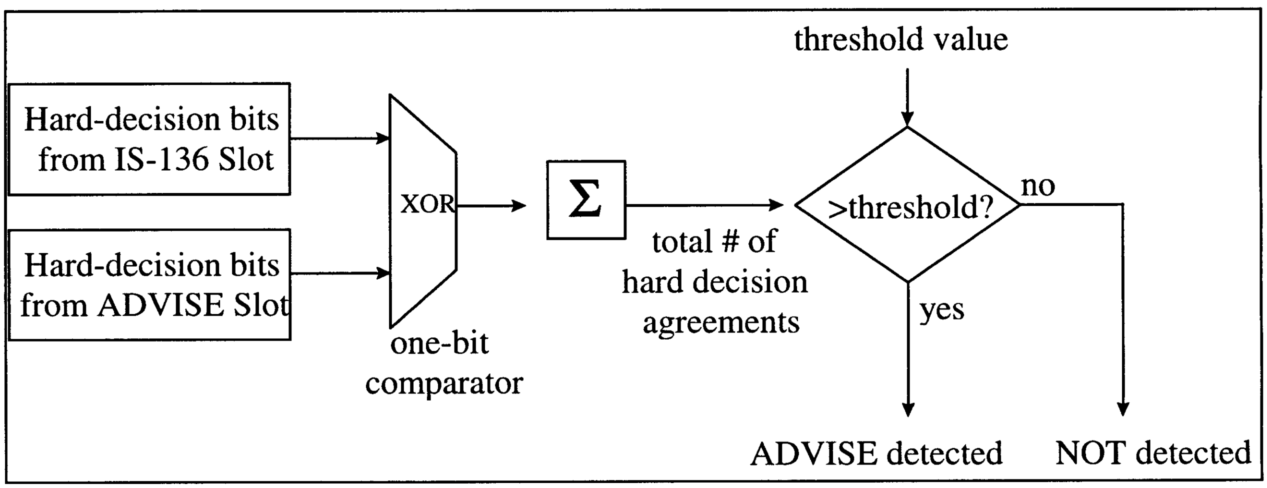

Figure 5.3 Hamming-Distance Threshold Detection

In Figures 5.5a-b and 5.6a-b, we compare the false-alarm and missed-detection rates versus SNR under two fading rates (slow and fast) for two non-adaptive thresholds, $N = 35$ and $N = 39$. We show that when we lower the threshold N, the detection rate improves whereas the false-alarm rate degrades. For a fixed N, we achieve approximately constant false-alarm rate (CFAR) in both slow-fading and fast-fading environments.

Representative performance results for ADVISE using the proposed non-adaptive autonomous blind detection method are presented in Figures 5.6c and 5.7c. In addition, we compare the CRC failure rates for the non-adaptive implementation versus the ideal case in which the SU has perfect knowledge about whether or not the ADVISE bits are available. From these results, we see that, when ADVISE is present, the lower threshold $N = 35$ results in virtually no performance degradation since the auxiliary data are almost always detected. When ADVISE is absent, however, the lower threshold causes some degradation due to the occasional acceptance of a bad frame as valid auxiliary data. For the higher threshold $N = 39$, the opposite is true: There is no degradation due to false acceptance of invalid data, but there is less ADVISE performance gain due to the higher rate at which valid ADVISE data are rejected.

5.3.4 Adaptive Hamming-Distance Threshold

It is possible to enhance the robustness of the basic detection method by making the threshold adaptive. For the IS-136 application, the SU can reasonably determine when it is a candidate for the auxiliary ADVISE bits and, once it begins receiving ADVISE bits, can reasonably expect the help to continue over an extended period of time (even though the actual duration is a random variable depending on traffic load). Thus, a simple finite-state machine representation of system operation would enable the ADVISE detection threshold to be set higher for initial acquisition and then reduced once sustained ADVISE service begins.

Let us designate three system states as follows:

State	**Description**	**Threshold**

NORMAL	Standard IS-136 mode of operation – SU does not look for ADVISE	–
EXPECTANT	SU with disadvantaged RF channel searches for ADVISE	$N_{\text{expectant}}$
ENHANCED	SU enjoys sustained ADVISE service	N_{enhanced}

Table 5.1 Table of Dual-Threshold Mode State Transition

The SU begins in the NORMAL state. As part of its normal IS-136 processing, it provides the base station with information as to the perceived quality of the radio downlink. While the channel is favorable, the SU has no expectation that the base station will begin offering it the ADVISE service and therefore will not look for it. When the channel deteriorates, however, the SU can expect the base station to begin providing ADVISE service, subject to the availability of resources. The SU then enters the EXPECTANT state and begins autonomously looking for the ADVISE bits using the threshold value $N_{\text{expectant}}$. Since the SU does not know when, if ever, the ADVISE service will begin, the threshold $N_{\text{expectant}}$ must be set relatively high to keep the false detection rate low. Once ADVISE is successfully detected, the SU uses the auxiliary bits to enhance the performance of its channel decoder. It continues in the EXPECTANT state, using the ADVISE bits as they are detected, until sustained ADVISE service has been achieved. When this condition is reached, the SU enters the ENHANCED state and, in order to maximize detection success, continues to look for the ADVISE bits using the lower threshold N_{enhanced}. Under these circumstances, the overall false-alarm rate is not adversely affected by the lower threshold, since the ADVISE service is usually present. Once the base station stops providing ADVISE, the SU quickly detects its absence and returns either to the EXPECTANT state with the higher detection threshold (if the unaided RF channel is still disadvantaged) or to the NORMAL state (if the unaided RF channel is no longer disadvantaged).

As a specific example design, we set $N_{\text{expectant}} = 39$ and $N_{\text{enhanced}} = 35$, based on the simulation results for the non-adaptive method. In the EXPECTANT mode, the SU uses the detection method to check for the presence of ADVISE bits. If the detection threshold is exceeded and the resultant decoded word passes the CRC test, the SU records a hit. If Q of the last P frames score a hit, then the SU declares that sustained ADVISE service has been established. It then switches to the ENHANCED state and uses the higher detection threshold. Similarly, the SU falls out of the ENHANCED state if the history of the last P frames suggests that ADVISE is no longer on. The state diagram for this these transition rules is illustrated in Figure 5.4.

In Figures 5.5d and 5.6d, we show how the three-state adaptive implementation using two thresholds provides enhanced detection/false-alarm rate operation. These results have been confirmed by tests in which the autonomous blind detection method has been simulated and applied to actual speech samples in controlled laboratory tests.

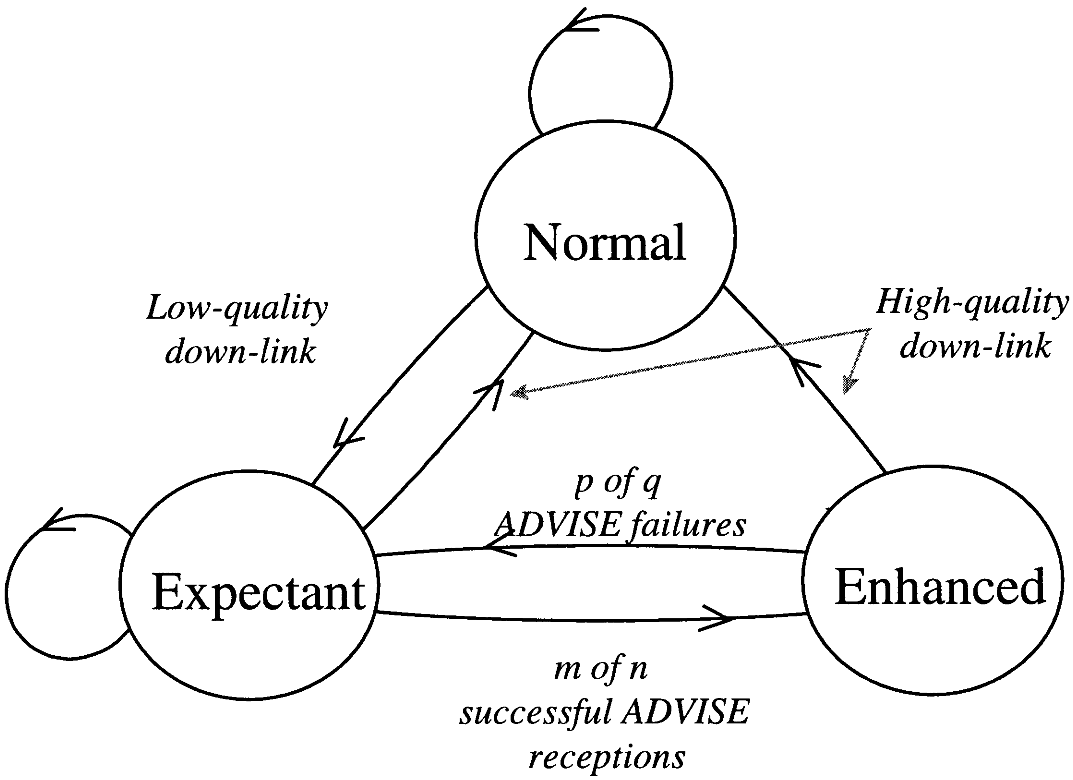

Figure 5.4 Dual-Threshold-Mode State-Transition Diagram

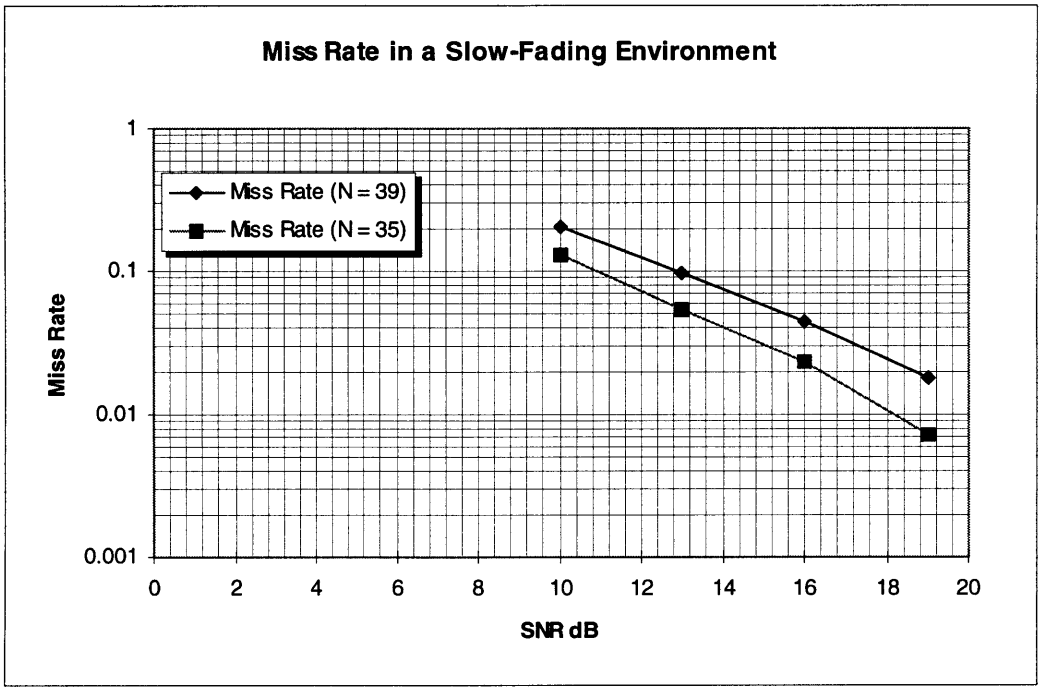

Figure 5.5a Missed-Detection Rate in a Slow-Fading Environment

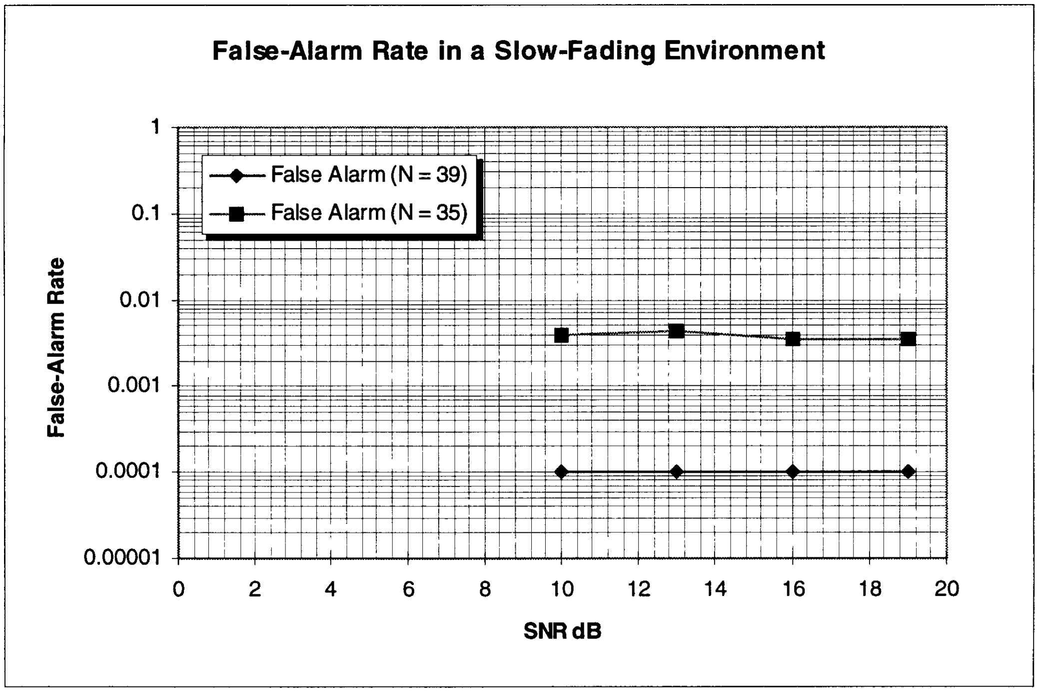

Figure 5.5b False-Alarm Rate in a Slow-Fading Environment

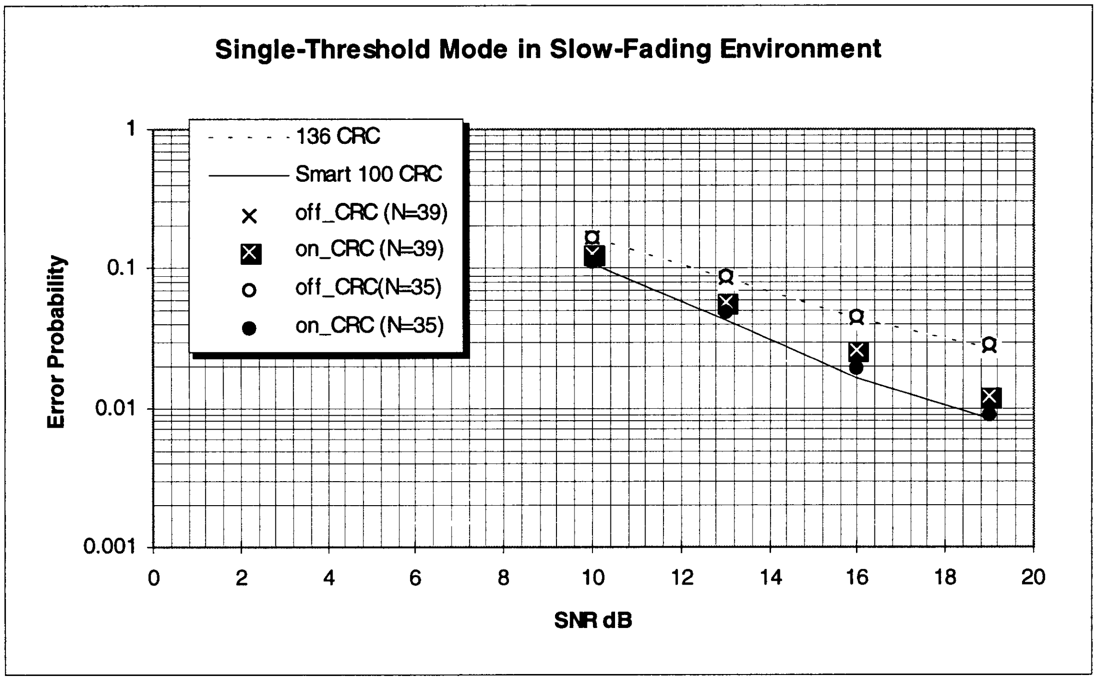

Figure 5.5c Single-Threshold Mode in Slow-Fading Environment

- 136 CRC: Standard IS-136 Coding/Decoding (zero false-alarm);
- Smart 100: 100 ADVISE Bits are available (perfect detection);
- Off_CRC (N = 39): CRC Error Rate when ADVISE is off; threshold = 39;
- On_CRC (N = 39): CRC Error Rate when ADVISE is on; threshold = 39;
- Off_CRC (N = 35): CRC Error Rate when ADVISE is off; threshold = 35;
- Off_CRC (N = 35): CRC Error Rate when ADVISE is on; threshold = 35.

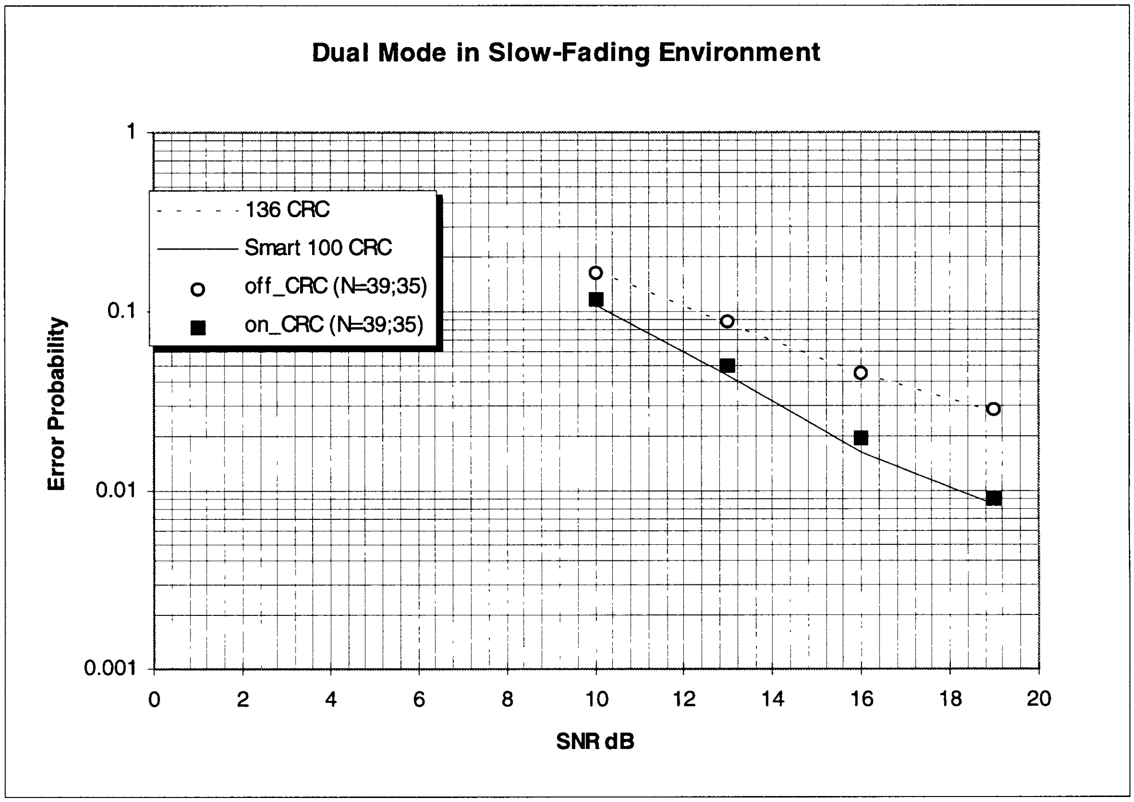

Figure 5.5d Dual-Threshold Mode in Slow-Fading Environment

- 136 CRC: Standard IS-136 Coding/Decoding (zero false-alarm);
- Smart 100: 100 ADVISE Bits are available (perfect detection);
- Off_CRC (N = 39; 35): CRC Error Rate when ADVISE is off;
  Threshold of Expectant State = 39; Threshold of Enhanced State = 35;
- On_CRC (N = 39; 35): CRC Error Rate when ADVISE is on;
  Threshold of Expectant State = 39; Threshold of Enhanced State = 39.

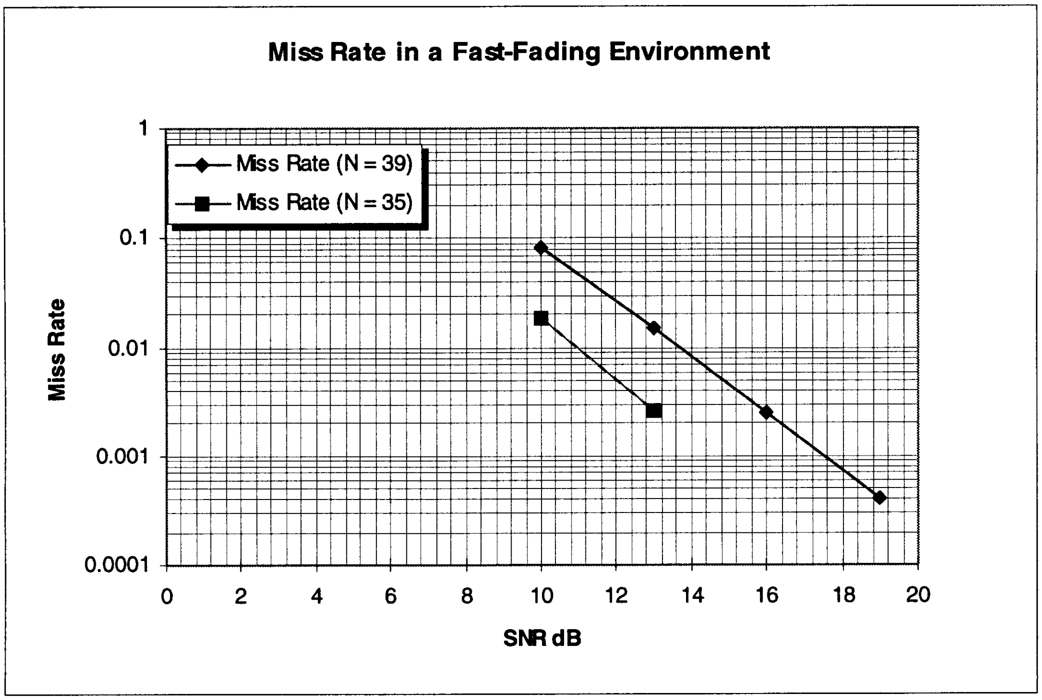

Figure 5.6a Missed-Detection Rate in a Fast-Fading Environment

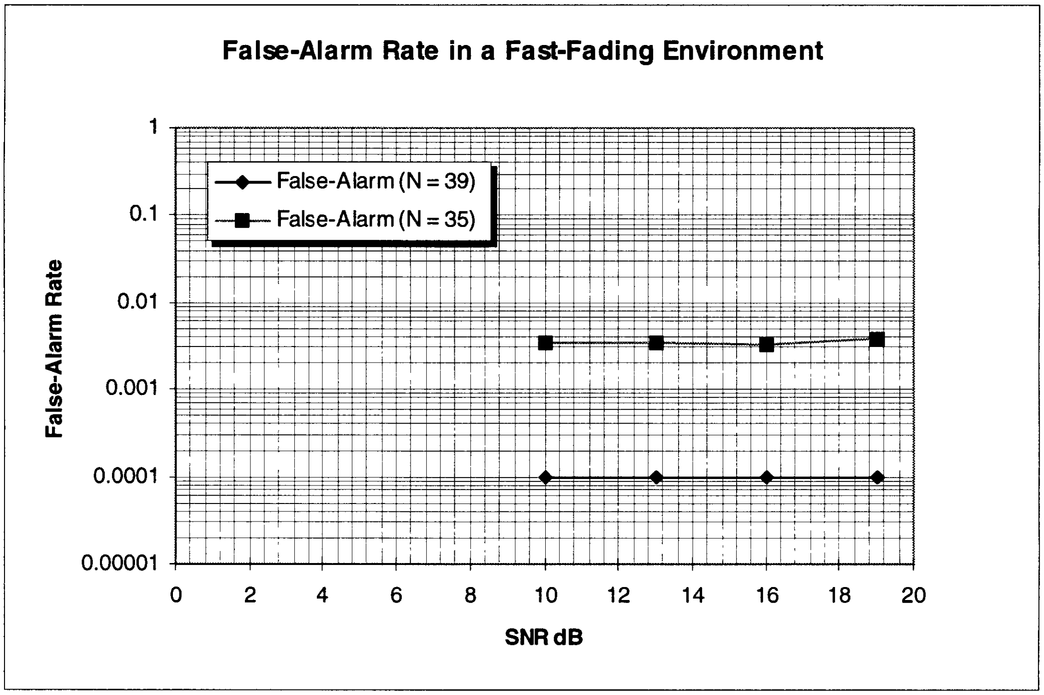

Figure 5.6b False-Alarm Rate in a Fast-Fading Environment

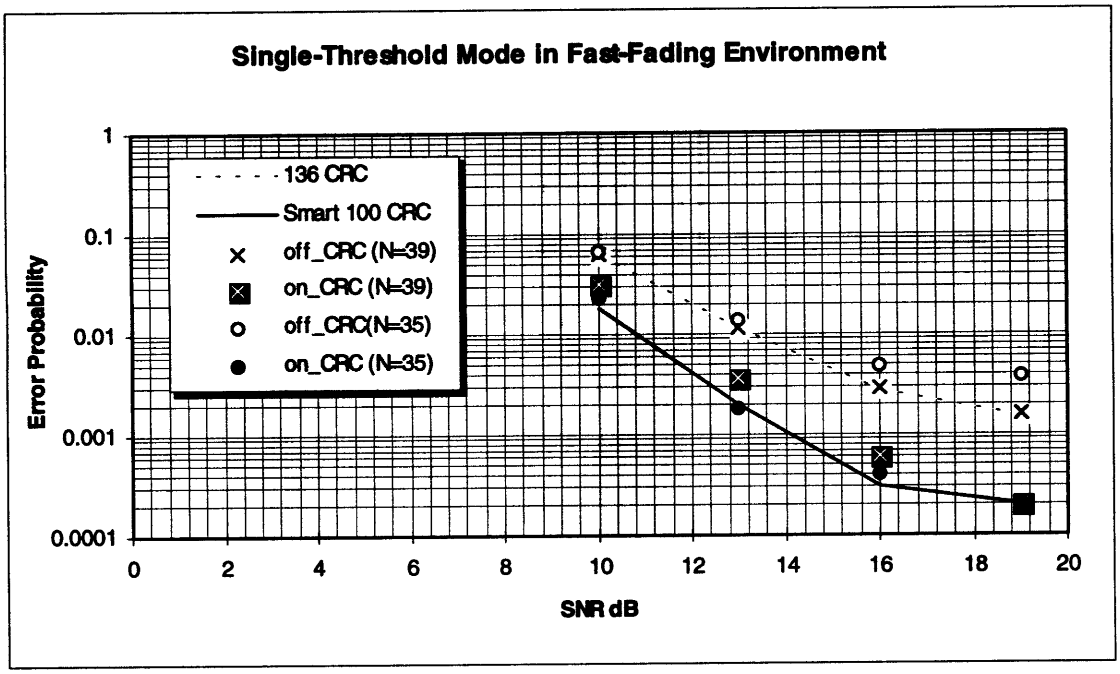

Figure 5.6c Single-Threshold Mode in Fast-Fading Environment

- 136 CRC: Standard IS-136 Coding/Decoding (zero false-alarm);
- Smart 100: 100 ADVISE Bits are available (perfect detection);
- Off_CRC (N = 39): CRC Error Rate when ADVISE is off; threshold = 39;
- On_CRC (N = 39): CRC Error Rate when ADVISE is on; threshold = 39;
- Off_CRC (N = 35): CRC Error Rate when ADVISE is off; threshold = 35;
- Off_CRC (N = 35): CRC Error Rate when ADVISE is on; threshold = 35.

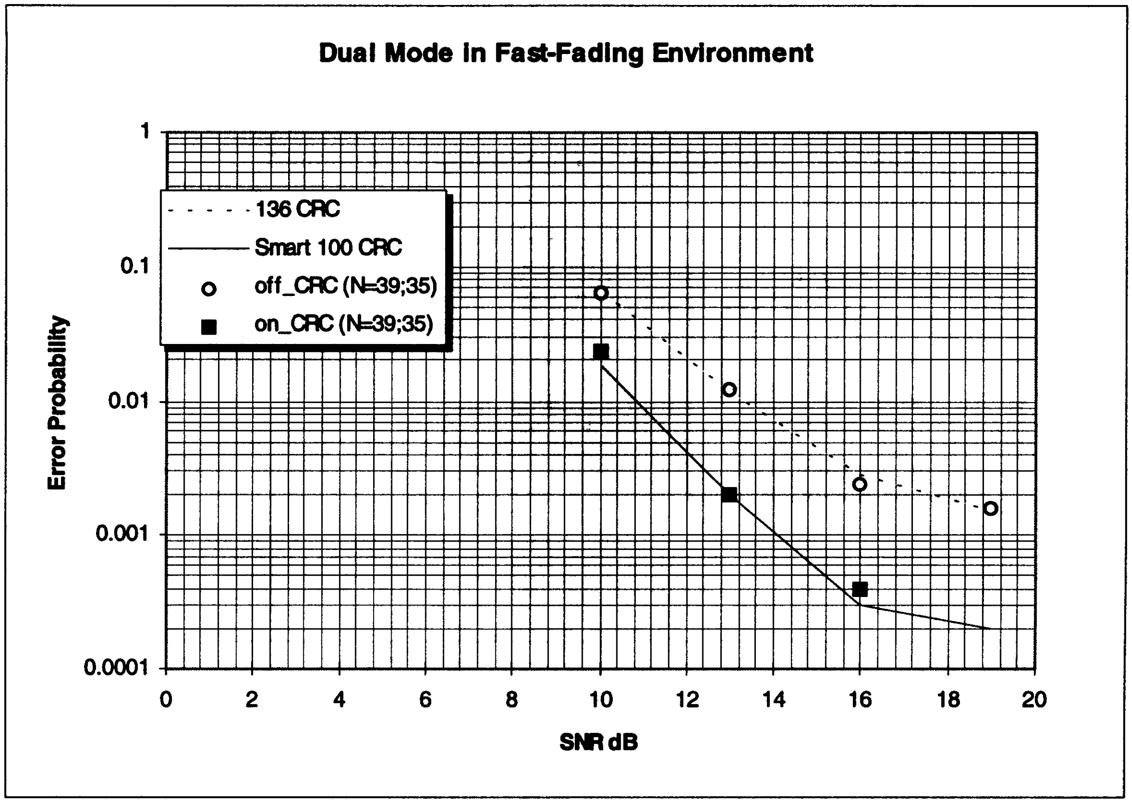

Figure 5.6d Dual-Threshold Mode in Fast-Fading Environment

- 136 CRC: Standard IS-136 Coding/Decoding (zero false-alarm);

- Smart 100: 100 ADVISE Bits are available (perfect detection);

- Off_CRC (N = 39; 35): CRC Error Rate when ADVISE is off;
  Threshold of Expectant State = 39; Threshold of Enhanced State = 35;

- On_CRC (N = 39; 35): CRC Error Rate when ADVISE is on;
  Threshold of Expectant State = 39; Threshold of Enhanced State = 39.

Chapter 6. Conclusion and Suggested Work

The various FEC overlay schemes proposed in Chapter 4 have been tested by simulations. We conclude that the ADVISE concept of utilizing the idle slot by transmitting variable lower-rate codes is both attractive and practical. With few added computations, the existing decoding firmware can be modified to work with rate-1/3, rate-¼ codes, etc. All of the proposed ADVISE implementations are backward-compatible, so that old handsets without ADVISE support can continue to operate.

The most attractive ADVISE coding scheme, according to our results, is the mix-and-match scheme. In addition to extra Class 1A and 1B protection, the mix-and-match scheme provides protection to previously unprotected Class 2 bits.

The autonomous detection algorithm discussed in Chapter 5 is also an important consideration for selecting an optimal ADVISE scheme. Augmented codes such as rate-1/3 and rate-¼ codes transmit different coded bits in the ADVISE slots, therefore cannot exploit the benefit of the autonomous detection algorithm. On the other hand, the autonomous detection algorithm is applicable with the mix-and-match scheme or for simple full-slot repeat because a set of identical bits is sent twice to the mobile. As shown in our simulations, the proposed detection algorithm using dual-threshold-mode with the Hamming distance metric provides a higher detection rate and a lower false-alarm rate. This detection algorithm is computationally simple, and requires only minor modifications of the existing firmware.

We recommend further analysis of the mix-and-match scheme by more refined simulations and by subjective human listening tests. Forward error correction (FEC) for voice is full of surprises. Incremental improvements in word error rate are often too subtle for human ears to distinguish. Consequently, we believe that simple schemes such as full-slot repeat, though inferior in simulation performance, may offer voice quality that is competitive with more sophisticated FEC schemes. Future investigations should address trade-off issues in voice quality, decoder complexity and signaling complexity, as well as backward compatibility.

References

[1] Hagenauer, Joachim. "Rate-Compatible Punctured Convolutional Codes (RCPC Codes) and their Applications." IEEE Transactions on Communications. Vol. 36, pp. 389-400, April 1988.

[2] Harvey, Bruce A. and Wicker, Stephen. "Packet Combining Systems Based on the Viterbi Decoder." IEEE Transactions on Communications. Vol. 42, pp. 1544-1557, Feb/March/April 1994.

[3] Kallel, Samir and Haccoun, David. "Sequential Decoding with ARQ and Code Combining: A Robust Hybrid FEC/ARQ System." IEEE Transactions on Communications. Vol. 36, pp. 773-779, July 1988.

[4] Kallel, Samir and Haccoun, David. "Sequential Decoding with an Efficient Partial Retransmission ARQ Strategy." IEEE Transactions on Communications. Vol. 39, pp. 208-213, Feb 1991.

[5] Lin, Shu and Costello, Daniel J. Jr. *Error Control Coding: Fundamentals and Applications.* Prentice Hall, 1983.

[6] IS-641 Manual. TDMA Cellular/PCS - Radio Interface - Enhanced Full-Rate Speech Codec. TIA/EIA, March 1996.