AN ADAPTIVE CODING SCHEME FOR EFFECTIVE BANDWIDTH AND POWER UTILIZATION OVER NOISY COMMUNICATION CHANNELS

Million Fekadu and Hailu Ayele

Department of Electrical & Computer Engineering

Addis Ababa University

ABSTRACT

Codes for communication channels are in most cases chosen on the basis of the signal to noise ratio expected on a given transmission channel. The worst possible noise condition is normally assumed in the choice of appropriate codes such that a specified minimum error shall result during transmission on the channel. However, as the noise conditions on communication channels vary from time to time, the use of a particular code that meets the specified error criteria under the assumption of worst noise condition may not result in the optimum utilization of the channel. One can exploit the variations of the noise level on the channel to increase the throughput on a fixed bandwidth channel without increasing the transmitter power by the use a coding scheme that adapts itself to the noise conditions on the channel.

An adaptive coding scheme that utilizes the different merits of more than one fixed rate codes and the noise level variations on a communication channel to achieve an increased throughput has been implemented in this work. An adaptive coding scheme using three fixed-rate BCH codes was developed and implemented on a computer. The results obtained in transmitting a given text over a simulated channel indicate an improved performance in terms of download time and coding gain. The use of the adaptive coding scheme results in a saving of 28% in download time and a coding gain of 2.61 dB when compared to the performance of a given reference code.

INTRODUCTION

Error control coding (ECC) uses controlled redundancy to detect and correct errors. The main issue in error control coding research is to find an effective means of adding redundancy to the message to be sent over the communication channels so that the receiver can fully utilize that redundancy to detect and correct the errors and improve the coding gain. In this introductory section and the section that follows the nature and characteristics of linear codes and the basic concepts behind the type of coding scheme used in the implementation part of this study will be presented.

There are many types of error control codes. Each code is distinguished by the method used to add redundancy and how much of this redundancy is added to the information going out of the transmitter. In the implementation of the coding scheme in this study, the BCH codes that belong to the class of cyclic codes are employed.

The block diagram shown in figure-1 represents a typical transmission system of a digital communication system that involves coding. The transmitter generates a sequence of binary bits called the message sequence, designated by **u**.

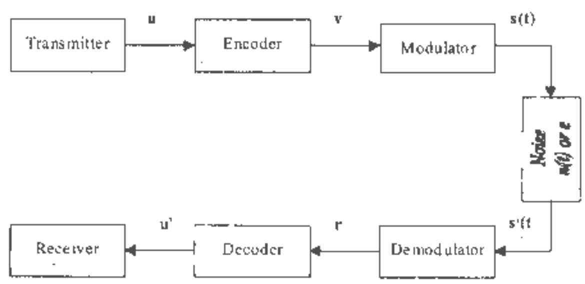

Figure 1 Error Control Coding path

The encoder transforms the information or message sequence **u** into an encoded seqnence **v** called the codeword. A codeword includes the information bits (may be in a slightly altered form) plus the redundancy in the form of parity checks. As the discrete symbols are not suitable for transmission over a physical channel, a modnlator is used to convert each ontput symbol of the channel encoder into a waveform **s(t)** of duration T seconds. The transmitted signal is therefore an encoded and modulated waveform.

The waveform s(t) that enters the communication channel may be corrupted by noise, **n(t)** as it travels over the channel. The received signal represents a modulated signal which is corrupted by noise, **n(t)** and the demodulated received signal represent the received sequence, **r**. The consequence of the additive noise signal on the received sequence, **r** is to create an error pattern in the binary tuple domain. Thus, if the transmitted codeword v has an error, what arrives at the receiving end would be v+e, which will have an error in every position where **e** has a '1'. A decoded received sequence **u'** is an estimate of a transmitted message tuple. If there were no errors or whatever errors did occur were correctable **u'** will equal **u**. However, if there was an undetectable error or the decoder makes an error, **u'** will be different from u. Even though there may be many types of noise that can be present on a given communication channel, the Additive White Gaussian Noise (AWGN) is the most commonly encountered and this is assumed for the communication channel considered in this work.

The following section presents a review of the basic concepts and theory of linear block coding schemes that would enable the automatic detection and correction of errors that are introduced because of the noise on the channel. This presentation is aimed at providing sufficient background to the basic principles of linear block coding and the methods or procedures employed in the coding and decoding of the *Bose, Chaudhuri and Hocquenghen* (BCH) codes.

BASIC CONCEPTS

A binary block code is a set consisting of 2^k codewords, such that modulo-2 sum of any two codewords is also another codeword. k is any positive integer that is greater than or equal to 1. In block coding the output of information source, a sequence of binary digits "0" or "1", is segmented into message blocks of fixed length. Each message block denoted by **u**, consists of k information digits. There are 2^k possible message blocks that correspond to the 2^k codewords of the block code.

In the development of the algebraic properties of a block code the components of a code vector $\mathbf{v}=(v_0, v_1,..,v_{n-1})$ are treated as the coefficients of a polynomial $\mathbf{v}(X)=v_0 + v_1X + \ldots + v_{n-1}X^{n-1}$, where $\mathbf{v}(X)$ is called the *code polynomial* of **v**. Each code vector corresponds to a polynomial of degree n-1 or less.

One of the most important subclass of linear block codes is the *cyclic code*. It possesses an algebraic structure that makes the encoding and decoding scheme easier than the other codes [2]. An (n, k) linear block code C is called a cyclic code if every cyclic shift of a code vector in C is also another code vector in C. Another important subclass of linear block codes, which is frequently encountered in practice, is the *Hamming code*, which is one of the first linear codes devised for error correction. The Hamming code comprises of a class of codes having a block length for a codeword $n=2^m-1$, and information length $k=2^m-m-1$, where m is any positive integer greater than or equal to three and corresponds to the number of parity check digits. A Hamming code with these parameters is capable of correcting one error per block of message [2].

Bose, Chaudhuri and Hocquenghem (BCH) Codes are also one of a large class of powerful random error correcting cyclic codes. BCH codes can be viewed as a generalization of Hamming codes for multiple-error correction. BCH codes are characterized by the following parameters for any positive integer, m $(m\geq3)$ and t $(t<2^{m-1})$.

Block Length:	$n - 2^m - 1$
Parity Check Digits:	$n - k \leq mt$
Minimum Distance:	$d_{min} \geq 2t+1$

These codes are capable of correcting any combination of t or fewer errors in a block of n digits, and hence they are also known as *t-error-correcting BCH codes* [2].

Since the implementation part of this work is based on BCH codes, we will hereinafter concentrate on encoding and decoding schemes of BCH codes. In the following some terminologies that will help us understand linear block codes and concepts of theGalois fields are defined and discussed.

Let **v** and **w** be binary n-tuples. The **Hamming weight** of **v**, denoted by $\omega(\mathbf{v})$, is defined as the number of non-zero components of **v** and the **Hamming distance** between v and **w**, denoted by $d(\mathbf{v},\mathbf{w})$ is defined as the number of places where they differ. Given a block code C, the **minimum distance** d_{min} is defined as

$$d_{min} = \mathrm{Min}\ \{d(\mathbf{v},\mathbf{w}): \mathbf{v},\mathbf{w} \in C, \mathbf{v}\neq\mathbf{w}\}.$$

This minimum distance of a linear block code is equal to the minimum weight of its non-zero codewords.

If the minimum distance between codewords in a block code C is d_{min}, any two distinct code vectors of C differ in at least d_{min} places; no error pattern of d_{min}-1 or fewer errors can change one code vector into another [2]. Hence, a block of code with minimum distance d_{min} is capable of detecting all error patterns of d_{min}-1 or fewer. An (n, k) linear code is capable of detecting $2^n - 2^k$ error patterns of length n. There are 2^k-1 undetectable error patterns. For large n, 2^k-1 is much smaller than 2^n. The undetectable error patterns therefore represent only a small fraction of the total error patterns.

A block code with minimum distance d_{min} guarantees correction of all error patterns of $t = [(d_{min}-1)/2]$ or fewer errors, where $[(d_{min}-1)/2]$ denotes the largest integer not greater than $(d_{min}-1)/2$. The parameter t is called the random-error-correcting capability of the code. A t-error correcting (n, k) linear code is capable of correcting a total of 2^{n-k} error patterns, including those with t or fewer errors [2]. *From the above discussion we see that the minimum distance of a block code determines the error detection and error correction capabilities of the code. Clearly, for a given n and k one would like to construct a code whose minimum distance is as large as possible.*

A set consisting of two integers {0, 1} is called a Galois Field denoted by GF (2). It is a commutative group under binary addition and binary multiplication. A polynomial $f(X)$ over GF(2) with one variable X and with coefficients from GF(2) has the form, $f(X) = f_0 + f_1X + f_2X^2 + \ldots + f_nX^n$, where $f_i = 0$ or 1 for $0 \leq i \leq n$. A polynomial p(X) over GF(2) of degree m is said to be irreducible if p(X) is not divisible by any polynomial over GF(2) of degree less than m but greater than zero. On the other hand any irreducible polynomial over GF(2) of degree m divides $X^{2^m-1}+1$. An irreducible polynomial p(X) of degree m is said to be primitive, if the smallest positive integer n for which p(X) divides $X^n + 1$ is $n=2^m-1$. It is not easy to recognize a primitive polynomial. However, there are tables for irreducible polynomials in which primitive polynomials are listed in power, polynomial or m-tuple forms of representation [2].

Galois Field denoted by $GF(2^m)$ is an extension field of GF(2) and has 2^m elements. In addition to '0' and '1' symbols of GF(2), $GF(2^m)$ has a new symbol 'α'. It is generated from a primitive polynomial over GF(2) of degree m. The new symbol α is a root of the primitive polynomial over $GF(2^m)$ which consists the set $F = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^{2^m-2}\}$ with 2^m distinct elements. A smallest degree polynomial, $\phi(X)$, over GF(2) such that $\phi(\beta)=0$ is called the minimal polynomial of element β in a Galois field. It can be obtained using Equation 1, where *e* is the smallest integer such that, $\beta^{2^e}=\beta$. β^{2^e} are called the **distinct conjugates** of β.

$$\phi(X) = \prod_{i=0}^{e-1} \left(X + \beta^{2^i}\right) \qquad (1)$$

Encoding & Decoding Schemes of BCH Codes

Encoding and decoding schemes of BCH codes are based on generator polynomials and the systematic structure of codewords. In an (n, k) BCH code there exists a minimum degree code polynomial, **g**(X), known as the generator polynomial of the code. Every other code polynomial is a multiple of **g**(X) and every linear polynomial of degree n-1 or less that is a multiple of **g**(X) is a code polynomial. The degree of **g**(X) is equal to *the number of parity check digits of the code.* Generator polynomials have the following form:

$$\mathbf{g}(X) = 1 + g_1X + \ldots + g_{n-k-1}X^{n-k-1} + X^{n-k} \qquad (2)$$

A binary n-tuple $\mathbf{v} = (v_0, v_1, v_2, \ldots, v_{n-1})$ is a codeword of a t-error-correcting BCH code if and only if the polynomial $v(X) = v_0 + v_1X + v_2X^2 + \ldots + v_{n-1}X^{n-1}$, has $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t}$ as its roots. Generator polynomial, **g**(X), of a t-error-correcting BCH code of length 2^m-1 is the lowest-degree code polynomial over GF(2).
For BCH codes **g**(X) is the least common multiple of $\phi_i(X)$ and is given by

$$\mathbf{g}(X) = \mathrm{LCM}\{\phi_1(X), \phi_3(X), \ldots, \phi_{2t-1}(X)\} \qquad (3)$$

where $\phi_i(X)$, i=1,2,....2t-1, is the minimal polynomial of α^i [2].

A desirable property linear cyclic block codes in general, and the BCH code in particular, have to possess is the systematic structure of their codewords. This structure is essentially a concatenation of (n-k) parity check bits and k message bits.

Every code polynomial $\mathbf{v}(X)$ in an (n, k) BCH code can be expressed as

$$\begin{aligned}\mathbf{v}(X) &= \mathbf{u}(X)\mathbf{g}(X)\\ &= (u_0 + u_1X + \ldots + u_{k-1}X^{k-1})\mathbf{g}(X) \qquad (4)\end{aligned}$$

where the coefficients of $\mathbf{u}(X)$ are the k information digits to be encoded, $\mathbf{g}(X)$ is generator polynomial of the code and $\mathbf{v}(X)$ is the corresponding code polynomial. Hence, encoding of a BCH code can be achieved by multiplying the message sequence, $\mathbf{u}(X)$ by the generator polynomial, $\mathbf{g}(X)$. However, the codewords obtained using this method do not have the required systematic structure. To put the codewords into systematic structure the following three steps are required [2].

1. Pre-multiply the message sequence $\mathbf{u}(x)$ by X^{n-k}
2. Obtain a remainder $\mathbf{b}(X)$ from dividing X^{n-k} $\mathbf{u}(X)$ by the generator polynomial, $\mathbf{g}(X)$. The remainder $\mathbf{b}(X)$ is the parity-check polynomial.
3. Combine $\mathbf{b}(X)$ and $X^{n-k}\mathbf{u}(X)$ to obtain the desired code polynomial,

$$\mathbf{v}(X)= \mathbf{b}(X) + X^{n-k}\mathbf{u}(X) \cdot \qquad (5)$$

All the above three steps can be implemented using the division circuit shown in Fig. 3.

The encoding operation is carried out as follows:

1. With the gate turned on, the k information digits $u_0, u_1, \ldots u_{k-1}$ [or $\mathbf{u}(X) = u_0 + u_1X + \ldots + u_{k-1}X^{k-1}$ in polynomial form] are shifted into the circuit and simultaneously into the communication channel. Shifting the message $\mathbf{u}(X)$ into the circuit from the front end is equivalent to pre-multiplying $\mathbf{u}(X)$ by X^{n-k}. As soon as the complete message has entered the circuit, the n-k digits in the register form the remainder $\mathbf{b}(X)$ and thus they are the parity check digits.
2. Break the feedback connection by turning off the gate.
3. Shift the parity-check digits out and send them into the channel. These n-k parity-check digits $b_0, b_1, \ldots, b_{n-k-1}$ together with the k information digits, form a complete code vector.

Decoding Schemes

One of the major parts of the decoding scheme for a BCH code is the computation of the syndrome of the received vector. For a t-error-correcting BCH code the syndrome is a 2t-tuple,

$$\mathbf{S} = (S_1, S_2, \ldots, S_{2t}).$$

whose i^{th} component of the syndrome is given by

$$\begin{aligned}S_i &= \mathbf{r}(\alpha^i)\\ &= r_0 + r_1\alpha^i + r_2\alpha^{2i} + \ldots + r_{n-1}\alpha^{(n-1)i} \qquad (6)\end{aligned}$$

for $1 \leq i \leq 2t$, where $\mathbf{r}(X)=r_0+r_1X+r_2X^2+\ldots+r_{n-1}X^{n-1}$ is a received polynomial version of the transmitted code polynomial $\mathbf{v}(X)=v_0+v_1X+v_2X^2+\ldots+v_{n-1}X^{n-1}$.

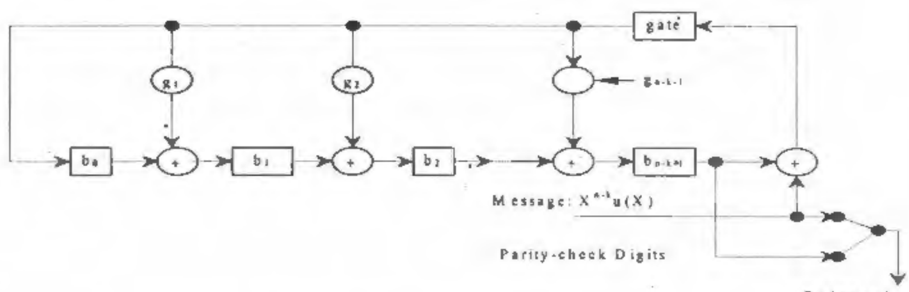

Figure 2 Encoding Circuit for an (n, k) BCH code

The syndrome components can also be computed from the error pattern $\mathbf{e}(X)$ as follows.

$\mathbf{r}(X) = \mathbf{v}(X) + \mathbf{e}(X)$ and $\mathbf{r}(\alpha^i) = S_i = \mathbf{v}(\alpha^i) + \mathbf{e}(\alpha^i)$

Since α, α^2, α^3, ..., α^{2t} are roots of code polynomial $\mathbf{v}(\alpha^i) = 0$, and the i^{th} *component of the syndrome* $S_i = \mathbf{e}(\alpha^i)$ for $1 \le i \le 2t$. If the error pattern has v errors at locations $X^{j1}, X^{j2}, \ldots, X^{jv}$ where $0 \le j_1 < \ldots < j_v < n$.; then the error pattern can be expressed as

$$\mathbf{e}(X) = X^{j1} + X^{j2} + \ldots + X^{jv}. \qquad (7)$$

The syndrome components corresponding to this error pattern are given by

$$S_i = \mathbf{e}(\alpha^i) = (\alpha^{j1})^i + (\alpha^{j2})^i + (\alpha^{j3})^i + \ldots + (\alpha^{jv})^i \qquad (8)$$

In the set of Eq. (8) the $\alpha^{j1}, \alpha^{j2}, \alpha^{j3}, \ldots, \alpha^{jv}$ are unknown variables. Any method of solving this set of equations can be a decoding algorithm for BCH codes [2]. Once $\alpha^{j1}, \alpha^{j2}, \alpha^{j3}, \ldots, \alpha^{jv}$ are found the powers $j_1, j_2, \ldots, j_v$ determine the error locations in $\mathbf{e}(X)$. For convenience, we can define the error location numbers as $\beta_l = \alpha^{j_l}$ for $1 \le l \le v$. In terms of the error location numbers, Eq. (8) can be written in an expanded form as follows

$$\begin{aligned} S_1 &= \beta_1 + \beta_2 + \ldots + \beta_v \\ S_2 &= \beta_1^2 + \beta_2^2 + \ldots + \beta_v^2 \\ &\vdots \\ S_{2t} &= \beta_1^{2t} + \beta_2^{2t} + \ldots + \beta_v^{2t} \end{aligned} \qquad (9)$$

Now let us define the following polynomial:

$$\begin{aligned} \sigma(X) &= (1 + \beta_1 X)(1 + \beta_2 X)\ldots(1 + \beta_v X) \\ &= \sigma_0 + \sigma_1 X + \sigma_2 X^2 + \ldots + \sigma_v X^v \end{aligned} \qquad (10)$$

The roots of $\sigma(X)$ are the inverses of the error location numbers. For this reason, $\sigma(X)$ is often referred to as the *error-location-polynomial*. It is an unknown polynomial whose coefficients must be determined [2]. The coefficients of $\sigma(X)$ can be expressed in terms of the error location numbers, $\beta_l = \alpha^{j_l}$ using Eq. (10).

$$\begin{aligned} &\sigma_0 = 1, \quad \sigma_1 = \beta_1 + \beta_2 + \ldots + \beta_v = S_1 \\ &\sigma_2 = \beta_1\beta_2 + \beta_2\beta_3 + \ldots + \beta_{v-1}\beta_v \\ &\quad \cdots \\ &\sigma_v = \beta_1\beta_2\beta_3 \ldots \beta_v \end{aligned} \qquad (11)$$

The above equations are related to the syndrome components of the received vector by the following Newton's identities.

$$\begin{aligned} &S_1 + \sigma_1 = 0 \\ &S_2 + \sigma_1 S_1 + 2\sigma_2 = 0 \\ &S_3 + \sigma_1 S_2 + \sigma_2 S_1 + 3\sigma_3 = 0 \\ &S_v + \sigma_1 S_{v-1} + \ldots + \sigma_{v-1} S_1 + v\sigma_v = 0 \\ &S_{v+1} + \sigma_1 S_v + \ldots + \sigma_{v-1} S_2 + \sigma_v S_1 = 0 \end{aligned} \qquad (12)$$

If it is possible to evaluate the elementary functions $\sigma_1, \sigma_2, \ldots, \sigma_v$ from Eq. (12), the error location numbers $\beta_1, \beta_2, \ldots, \beta_v$ can be found by determining the roots of error location polynomial $\sigma(X)$. From the above development we can conclude that the error-correcting procedure for BCH codes consists of the following three major steps.

1. Compute the syndrome $S = (S_1, S_2, \ldots, S_{2t})$ from the received polynomial $r(X)$.

2. Determine the error location polynomial, $\sigma(X)$, from the syndrome components.

3. Determine the error location numbers $(\beta_1, \beta_2, \ldots, \beta_v)$ from roots of $\sigma(X)$ and perform the error correction

Peterson [2] devised the decoding algorithm that carries out these three steps. Step-2 is the most complicated part of decoding a BCH code. In what follows, two algorithms that are used to find the error location polynomial and the error location numbers are outlined.

Berlekamp's Iterative Algorithm

This algorithm uses Eqs. (13), (14) and (15), Table 1 and the flow chart of figure-4 to find the error location polynomial, $\sigma(X)$. If the polynomial has degree greater than t, there are more than t errors in the received polynomial $\mathbf{r}(X)$, and generally it is not possible to locate them.

The following quantity

$$d_\mu = S_{\mu+1} + \sigma_1^{(\mu)} S_\mu + \sigma_2^{(\mu)} S_{\mu-1} + \ldots + \sigma_{l_\mu}^{(\mu)} S_{\mu+1-l_\mu} \qquad (13)$$

is called *the μ^{th} step discrepancy*. If $d_\mu = 0$, the coefficients of $\sigma^{(\mu)}(X)$ satisfy the $(\mu + 1)^{th}$ Newton's identity and

$$\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X) \tag{14}$$

If $d_\mu \neq 0$ the coefficients of $\sigma^{(\mu)}(X)$ do not satisfy the $(\mu + 1)^{th}$ Newton's identity and a correction term must be added to $\sigma^{(\mu)}(X)$ to obtain $\sigma^{(\mu+1)}(X)$. To accomplish this, one goes back to the steps prior to the μ^{th} step and finds a polynomial $\sigma^{(\rho)}(X)$ at step ρ, where the ρ^{th} discrepancy $d_\rho \neq 0$ and the term $(\rho - l_\rho)$ has the largest value,where l_ρ is the degree of $\sigma^{(\rho)}(X)$. Then the $\sigma^{(\mu+1)}(X)$ polynomial is given by

$$\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X) + d_\mu d_\rho^{-1} X^{(\mu-\rho)} \sigma^{(\rho)}(X) \tag{15}$$

This is the minimum degree polynomial whose coefficients satisfy the first $(\mu + 1)^{th}$ Newton's identities.

Table-1: Iteration process to obtain Error Location Polynomial

μ	$\sigma^{(\mu)}(X)$	d_μ	l_μ	$\mu - l_\mu$
-1	1	1	0	-1
0	1	S_1	0	0
1				
2t				

Chien's Search and Substitute Algorithm

The last step in the decoding procedure of a BCH code is finding the error-location numbers. Error-location numbers are reciprocals of the roots of the error location polynomial, $\sigma(X)$. The roots of $\sigma(X)$ can be found simply by substituting all possible Galois field elements except zero, i.e., α, α^2, ..., α^{n-1}, α^n where $n=2^m-1$ into $\sigma(X)$. If α^l is found to be a root of $\sigma(X)$ then its inverse α^{n-l} is an error-location number and the received vector has error at location n-l, i.e. the received digit r_{n-l} is an error digit. Chien's algorithm for searching error-location numbers is illustrated in the flow chart of Fig. 4.

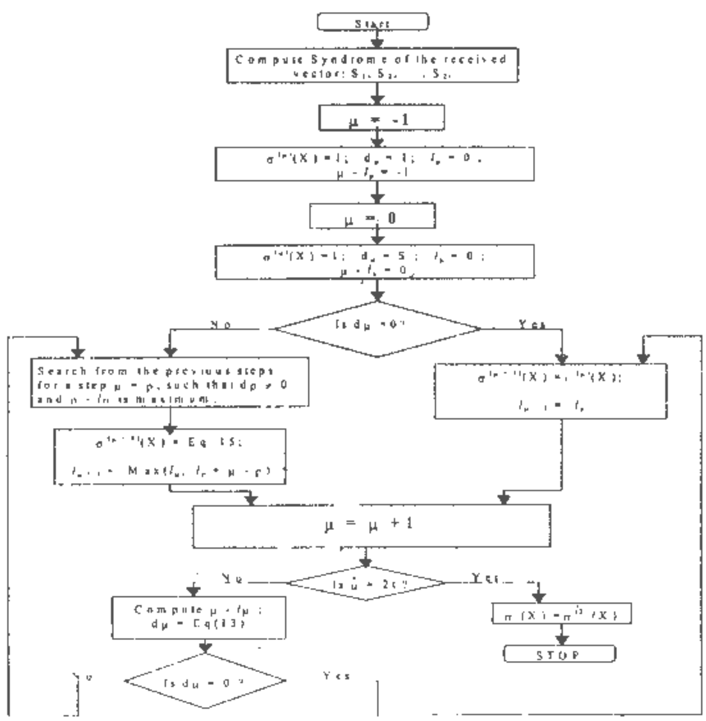

Figure 3 Berlekamp's Iterative Algorithm

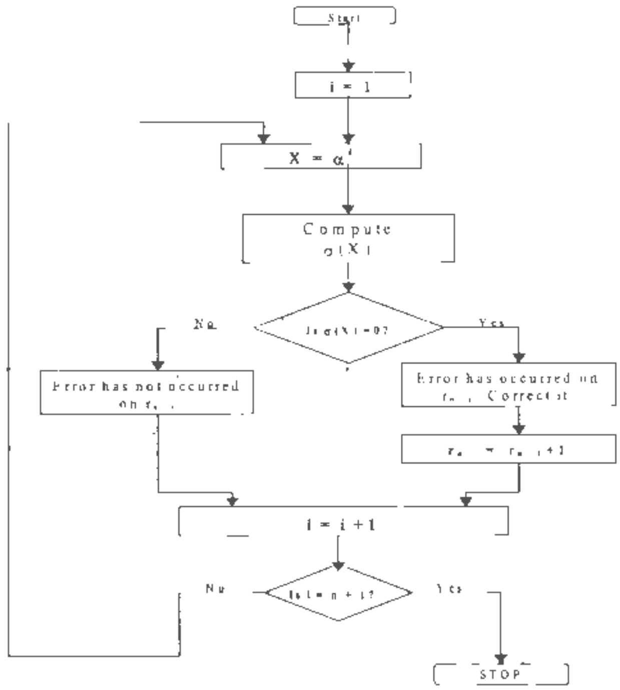

Figure 4 Chien's Search and Substitute Algorithm

To decode r_{n-l} the decoder forms the sum $\sigma(\alpha^l) = 1 + \sigma_1\alpha^l + \sigma_2\alpha^{2l} + . . . + \sigma_v\alpha^{vl}$. If this sum is zero, then α^l is a root of $\sigma(X)$ and r_{n-l} is not correctly received. Adding '1' to r_{n-l} can do the correction. If the sum is not zero r_{n-l} is correctly received and there is no need of correction. In this way the decoder decodes *each bit one by one* starting from the last bit and produces the decoded vector. If the number of errors is less than or equal to t, then a t-error-correcting BCH code will recover the transmitted codeword, otherwise the decoder commits an error.

IMPLEMENTATION

Three BCH codes, *Triple-error-correcting (15,5), Double-error-correcting (15,7)* and *Single-error-correcting (15,11),* are selected to implement the adaptive coding scheme. Each of these codes has code length of 15 bits and information bits of length k = 5, 7 and 11 respectively. *The code length of each of the three codes is chosen to be the same to satisfy the constant bandwidth constraint.*

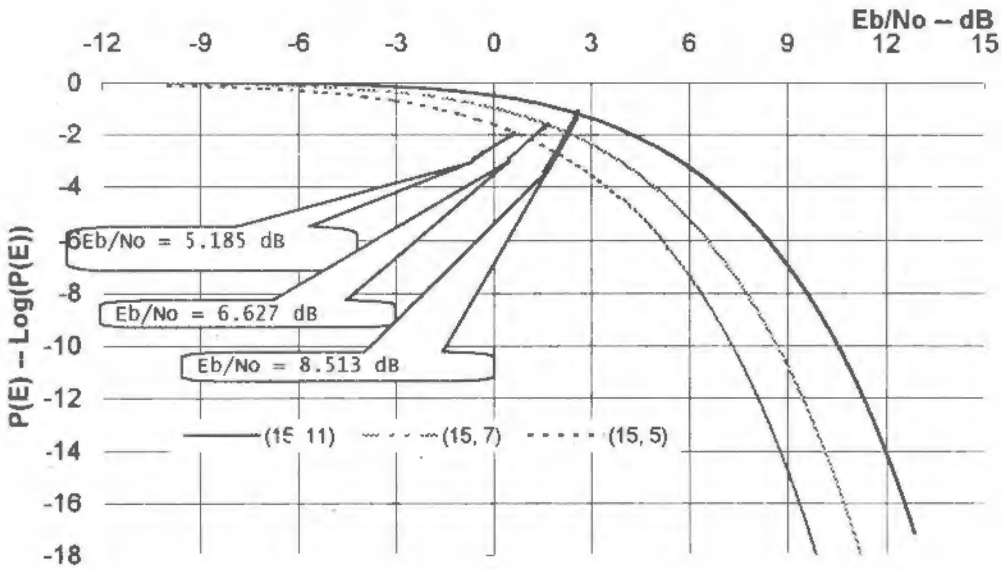

Figure 5 Plot of error Probability of the Codes

The (15,11) BCH Code is good for communication channels with minimum noise level: It is the fastest code when compared with the other two codes. About 73.3% of the total data transmitted over the channel per codeword is information bits. However, it has the weakest error correction capability when compared with the other two. When using the (15,7) code only about 46.7% of the transmitted data represent the information bits. It is slower than the (15,11) code but has better error correction capability. The (15,5) BCH code has strongest error correction capability, when compared with the other two codes. This capability is achieved at the expense of information transfer rate. Out of the 15 bits to be transmitted per block about 67% are overhead bits required for forward error correction (FEC) and only 33% represent the information bits. This would be the preferred code for communication channels that may be affected by strong noise.

The error probability over a transmission channel is a function of the transition probabilities of the channel. The transition probabilities can be calculated from the knowledge of the signal used, the probability distribution of the noise on the channel and the output quantization threshold [4]. Using the relationships among these variables, error probabilities of the three BCH codes are plotted as a function of the signal level in figure-5. Assuming that the upper-bound of the error probability is $P(E) \leq 10^{-6}$ i.e. one error in a million is acceptable for the type of communication under investigation, we note following characteristics of the three codes from the figure.

- Triple-error-correcting (15,5) BCH code is error free for signal to noise ratio ≥ 5.185 dB.
- Double-error-correcting (15,7) BCH code is error free for signal to noise ratio ≥ 6.627 dB, and
- Single-error-correcting (15,11) BCH code is error free for signal to noise ratio ≥ 8.513 dB.

Adaptive Coding Scheme

This work is based on the optimal utilization of a channel that basically demands the (15,5) BCH code for error free transmission. We cannot use the other two codes on this channel without increasing the transmitter power level. The transmitter power level available for the adaptive coding scheme is equal to the level required for the (15,5) BCH code. The bandwidth assigned for the adaptive coding scheme is also equal to the bandwidth that is assigned for the (15,5) BCH code. The information transfer rate that we seek to achieve using the adaptive coding scheme is to be greater than the one that can be achieved using the (15,5) BCH code

Measures of Performance

The performance of an adaptive coding scheme as compared to a fixed code suitable for a particular noisy communication channel can be measured by the download-time-saving and Code Gain attained by the adaptive coding scheme.

Download Time

Suppose that ΔT_f and ΔT_a are the download time required by a reference fixed code and an adaptive coding scheme, respectively, to complete transferring of a given text file. The Download Time Percentage (DTP) is then given by

$$DTP = \frac{\Delta T_a}{\Delta T_f} \times 100 \ \% \qquad (16)$$

The above expression indicates that if the adaptive coding scheme is to be superior to the reference fixed code DTP must be less than 100%. Accordingly, as the value of DTP decreases, the performance with respect to the speed of the adaptive coding scheme will increase.

Percentage Time Saved (PTS can be an alternative measure of speed to DTP and is given by

$$\mathbf{PTS = (1 - DTP)*100\%} \qquad (17)$$

It indicates the time saving attained on file transfer duration because of the introduction of adaptive coding scheme. Larger values of this percentage indicate more effective utilization of the channel bandwidth.

Code Gain

To measure the code gain of an adaptive coding scheme, another fixed rate code, in addition to the reference fixed rate code, needs to be employed for comparison purposes.

Let C_{rf} be the reference fixed rate code and C_{nf} be the new fixed rate code. Assume that the channel remains error free, for C_{rf} when the carrier power level is at X_0 dB and for C_{nf} when the carrier power level is at X_f dB.

Let r_{if} be information transfer rate attained by C_{rf} when the carrier power level is at X_0 dB. Further, let r_{ia} be information transfer rate attained by the adaptive coding scheme used on the same channel when transmitter carrier level is also set at X_0 dB Clearly, $r_{ia} > r_{if}$. The new fixed rate code, C_{nf}, is required to have information transfer rate equal to r_{ia} when its transmitter carrier level is set at X_f dB.

Code length, n, of the codes to be used on the channel under investigation shall remain constant in order to comply with the constant bandwidth constraint. Hence to attain the required information transfer rate r_{ia}, the new fixed rate code, C_{nf}, shall be chosen such that its code length equals that of C_{rf} and the number of overhead bits is smaller than that of C_{rf}, i.e., longer information bits per codeword for C_{nf}. This implies that the new fixed code has weak error correction capability and hence demands larger transmitter carrier level for error free transmission over the channel. X_f is therefore, greater than X_0.

The difference between these levels, $\mathbf{CG=X_f-X_0}$ is referred to as code gain, CG, attained by the adaptive coding scheme. CG greater than 0 dB indicates good performance attained by the adaptive coding scheme over the reference fixed rate code.

Functional Description

The transmitter-receiver system of the adaptive coding scheme implemented in this work is shown in the block diagram of Fig. 7.

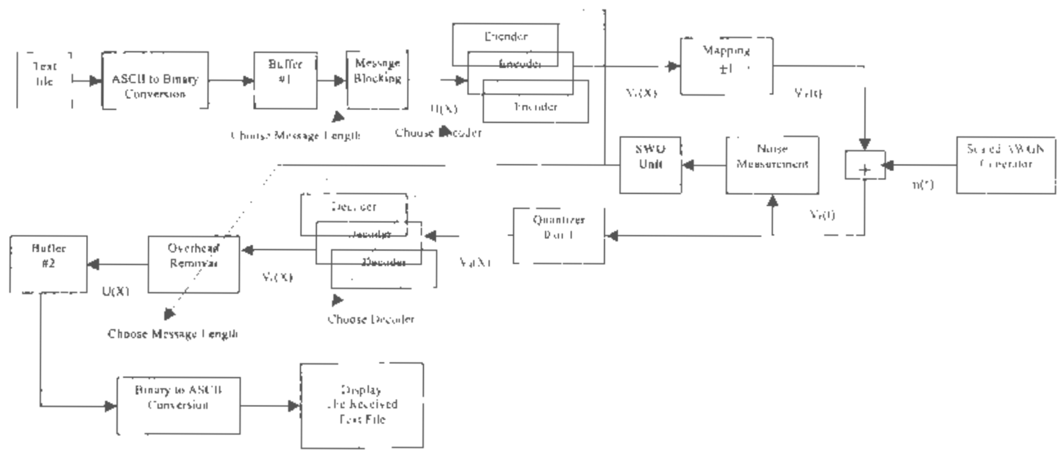

Figure 6 Block Diagram of Adaptive Coding Scheme

In what follows a functional description of each block is presented

Text File: This block represents any text file to be transferred from one site to another using adaptive coding scheme

ASCII to Binary: This block performs the ASCII to Binary conversion of each character of the text file. The output of this block is an equivalent binary data for each character of the text file.

Buffer#1: This is a temporary storage for the stream of binary bits.

Message Blocking: This block is used to segment the binary data received from buffer#1 into groups of information blocks, u(X). The length of the consecutive information blocks is determined by the signal from SWO unit

Systematic Encoder: This block encodes the input message block in systematic form as explained in section-III. One of the three encoders is set online at a time, based on the SWO control signal from the receiving station

Mapping: Mapping the one/zero output of the systematic encoder on to antipodal baseband signaling scheme provides the transmitted channel symbols. This is simply a matter of translating zeros to +1s and ones to -1s. This can be accomplished by performing the operation, **Y=1-2X**, on each encoded output symbol, where X represents the input binary signal and Y represent the output baseband signal. Mapping represents a simplified BPSK modulation.

Communication Channel Simulation: The channel assumed for the analysis is Additive White Gaussian Noise (AWGN) channel. Hence, adding noise to the transmitted channel symbols involves generating Gaussian random number, scaling the numbers according to the desired energy per symbol to noise ratio, Es/No, and adding the scaled Gaussian random numbers to the values of channel symbols.

The noise n(t) that corrupts an AWGN channel has constant power spectral density, N_o, watts/Hz over the entire frequency range, that is $\Phi_{nn}(f)=N_o$ for all f. This type of noise is wideband. For works with narrowband signals, it is mathematically convenient to pass the noise and the desired signal through an ideal band pass filter, having a pass band that includes the spectrum of the signal but is much wider [3].

The spectral density is given by

$$\Phi_{nn}(f)=\begin{cases} N_o & for\ |f \pm f_c| < \frac{B}{2} \\ 0 & otherwise \end{cases} \qquad (18)$$

The autocorrelation function of this noise signal, which is the Fourier transform of power spectral density, is [1]

$$\varphi_{nn}(\tau) = BN_o\ Si(B\pi\tau) \qquad (19)$$

where,

$$Si(\theta) = \frac{Sin(\theta)}{\theta} \qquad (20)$$

The power of white noise, which is also its variance, for the band limited case, can be computed as, [1]

$\sigma^2 = \varphi_{nn}(0) = N_o B$ and

$$N_o = \frac{\sigma^2}{B} \qquad (21)$$

The energy per symbol to noise ratio,

$$\frac{E_s}{N_o} = \frac{E_s}{\sigma^2 / B} = E_s \frac{B}{\sigma^2} \qquad (22)$$

We deal with a digital signal assumed to be corrupted on a bit-by-bit basis. Hence, B equals to 0.5Hz, that is a band occupied by a single bit. If we set the energy per symbol Es, equal to 1 unit of energy (for reference) then

$$\frac{E_s}{N_o} = \frac{1}{2\sigma^2} \qquad (23)$$

From equation (23) we can conclude that, by keeping the signal level constant (Es=1) a signal with a given Signal to Noise Ratio (SNR) can be obtained by varying only the noise level. Therefore, to simulate the desired channel of a given SNR, generate an AWGN noise of variance, $\sigma^2 = \frac{0.5}{Es/No}$ and add to the values of the channel symbols.

Quantization: Before decoding the received symbols, hard-decision quantization is required to convert the baseband signal to binary signal. In this work the received channel symbols are quantized to one-bit precision, i.e., if the symbol value is less than zero it is quantized to one, otherwise it is quantized to zero.

Systematic Decoder: This unit receives the output of the quantizer. It detects errors that may occur due to the channel noise. If there are correctable errors the decoder corrects them otherwise it commits erroneous decoding. The output of the decoder is a decoded codeword. Any of the three decoders become online based on the control signal received from the SWO Unit.

Overhead Removal: This unit receives the output from one of the decoders and removes the overhead portion of the block. Length of the overhead portion is decided by the control signal from SWO unit

Buffer#2: This is a temporary storage for a stream of received binary bits.

Binary to ASCII Conversion: This unit receives the binary digits from buffer#2 converts them to decimal number and then to an equivalent ASCII character.

Text Display: The received character of the transmitted text files are displayed one after the other in the order of their arrival

Noise Level Measurement: While the communication is in progress the adaptive coding scheme performs noise power measurement on the background using sequences of data collected previously for this purpose. It calculates and returns the sample variance of the sequences. The unit processes the last five blocks of the transmitted symbols at the end of reception of every block. This number is chosen based on the accuracy of the results obtained from repeated tests performed on the channel. The accuracy of the output of the measuring unit increases as the number of blocks used for measurement increases. However, if the number of blocks is excessively increased, it will cause measuring delay and consequently delays the overall trans-receive process.

SWO Unit: This unit receives the value of noise variance measured by the noise level-measuring unit. It compares this value with the preset range of noise variance and chooses a code that is appropriate for the next transmission.

Encoding and Decoding Operations

These operations depend on a signal received from the SWO units. As discussed in the above paragraphs the first five blocks are processed using a strong code, in this case the (15,5) BCH code. At the end of reception of the fifth block the SWO unit starts to send command signals to the units under its control.

If the noise variance falls below the maximum level set for the (15,11) BCH code, the SWO instructs the transmitter and the receiver to use

(15,11) on the next block. If the noise variance is above the maximum level set for (15,11) BCH code and below the maximum level set for (15,5) BCH code, then the next information block will be transmitted and received using (15,7) BCH code. If the measnred variance is above the maximum level set for (15,7) BCH code, then the system will configure itself for (15,5) BCH code and it remains online until the noise level decreases.

If the noise power is strong and passes the maximum level set for the (15,5) BCH code, the system will commit erroneous decoding and it is said to be in High Bit Error Rate (HBER) state. The occurrence of such condition is very limited since the channel is tested and the carrier level set for the best margin of (15,5) BCH code before it is used to carry traffic.

RESULTS

Noise Sensitivity of the Codes

Table 3: Measured Download Times

No.	Code Type	Eb/No [dB]	Average Download Time (sec)		
			Sample I	Sample II	Sample III
1	(15,11)	8.75	94	188	281
2	(15, 7)	7.41	147	295	441
3	(15, 5)	4.80	206	412	618

In order to determine the noise free operation zone of the codes several tests were made using a program written for this purpose and the results shown in table-3 are obtained.

Table 2: Measured Noise Response

No.	Code Type	Measured and Corrected Variance	Receive Eb/No[dB]
1	(15, 11)	$\sigma^2 \leq 0.06668$	$\geq$ 8.75
2	(15, 7)	$\sigma^2 \leq 0.09078$	$\geq$ 7.41
3	(15, 5)	$\sigma^2 \leq 0.16557$	$\geq$ 4.80

PERFORMANCES

Download Time

Table-4 shows average download times (in seconds) required to complete the transfer of three sample text files of various sizes by each fixed rate codes over a simulated channel appropriate for each of them.

Note

Sample-I	*Text file of size **2,386** bytes*
Sample-II	*Text file of size **4,774** bytes*
Sample-III	*Text file of size **7,162** bytes*

These text files were transmitted using the adaptive coding scheme and several readings were taken. The results are listed in table-5. Results obtained using sample-III, the largest size text file, can be taken as the overall average results for DTP and PTS calculation of the adaptive coding scheme. The average DTP is about 72%, which is equivalent to 28% average time saving. This implies, a channel that was working using the fixed code (15,5) is now able to save 28% the total download time because of the use of the adaptive coding scheme.

Code Gain (CG)

By employing a higher transfer rate fixed code on the channel used for adaptive coding scheme, keeping constant bandwidth, and trying to achieve the transmission speed achieved by the adaptive coding scheme, coding gain of the adaptive coding scheme can be measured. In this case the fixed code used for this purpose is the (15,7) BCH code. It is chosen because it is the only available intermediate rate code next to (15,5) BCH code of length 15 bits that approximates the transfer rate achieved by the adaptive coding scheme.

The time taken by this code to complete the transmission of the sample-III text file is 441 seconds at the signal to noise ratio of 7.41 dB. The average time taken by the adaptive coding scheme to complete the transmission of the same text file is 438.6 seconds at signal to noise ratio of 4.8 dB. Actually the speed achieved by the adaptive coding scheme is better than the speed achieved by (15,7) BCH code. For the purpose of this measurement these speeds can be assumed comparable and hence the difference between the levels of the two signal to noise ratios can be considered as the coding gain attained by the adaptive coding scheme.

CG = SNR of (15,7) – SNR of the Scheme
= 7.41 – 4.8 [dB] = **2.61 dB**

Table 4: Calculated DTP and PTS

Description	Download Time (Sec)		
	Samples		
	I	II	III
Trial No. 1	125	326	365
Trial No. 2	140	323	347
Trial No. 3	118	279	570
Trial No. 4	161	230	562
Trial No. 5	172	296	349
Average, ΔT_a	**143.2**	**290.8**	**438.6**
ΔT_f	**206**	**412**	**618**
Average, DTP	**69.5%**	**71.6%**	**72%**
Average, PTS	**30.5%**	**28.4%**	**28%**

CONCLUSIONS

From Table 3 and Fig. 6 it is clear that the error free zone values practically approximate the one obtained theoretically. For (15,11) BCH code at BER equals to 10^{-6}, the theoretically obtained SNR is 8.513dB while the one obtained by experiment is 8.75dB. For (15,7) and (15,5) codes also the theoretical value are 6.627dB and 5.185dB and the experimental values are 7.41dB and 4.8dB respectively.

From Table 5 we can see that the recorded download times for a given sample text file at different trials are different. This is because of noise level variations, which force the coding scheme to use one of the three codes for longer time than the others. For instance if the noise level is strong the (15,5) BCH code will be online for most of the time and if noise level is weak the (15,11) or (15,7) BCH codes will be the dominant codes that will be employed on the channel. The variability of the DTP that are observed in Table 5 are the consequences of this phenomenon. Furthermore, since the speed attained by the adaptive coding scheme is greater than the speed attained by (15,7) BCH code, we can surmise that the code gain is better than 2.61 dB.

In general the overall performance of the implemented adaptive coding scheme over a channel suitable only for (15,5) BCH code is better than the performance of a (15,7) BCH code on its own channel.

REFERENCES

[1] Ketema Alemu, *Digital and Analogue Communication System over Noisy Communication Channels*, Addis Ababa University, 1982.

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

[3] John G. Proakis, *Digital Communications*, McGraw-Hill, New York, 1995.

[4] Million Fekadu Demissie, *Implementation of Adaptive Coding Scheme for Optimum Utilization of Data Communication Channel's Bandwidth*, Addis Ababa University, 2001.