A Coding Theory Tutorial

Randy Yates

27–Aug–2009

Digital Signal Labs
signal processing systems

http://www.digitalsignallabs.com

888-708-3698

Typeset using LaTeX 2_ε

Contents

1 Introduction **3**
1.1 Overview . . . 3
1.2 Intended Audience . . . 4

2 Theory **4**
2.1 Elements of a Digital Communication System . . . 4
2.2 Coding Theory Basics and Block Codes . . . 4
2.3 Two Important Families of Codes . . . 5
2.4 Other Important Concepts . . . 6
2.4.1 Constructing a Decision Scheme . . . 6
2.4.2 Quantifying Coding Performance . . . 6

3 Example: A Soft-Decision Decoded Linear Systematic Code **7**
3.1 Theory . . . 7
3.2 Simulation . . . 9

4 For Further Study **9**

5 Terms, Abbreviations, and Notation **9**

6 Revision History **10**

7 References **11**

Appendices **12**

List of Figures

1 Digital Communication System Block Diagram . . . 3
2 Simulation Results . . . 10
3 Coded BPSK Simulation MATLAB M-File . . . 13
4 Uncoded BPSK Simulation MATLAB M-File . . . 14
5 Uncoded BPSK Theoretical Computation MATLAB M-File . . . 14

List of Tables

1 Revision History . . . 11

1 Introduction

1.1 Overview

Any real-world communication system is plagued by noise and thus susceptible to errors in the transmission of information. Theoretical work by pioneers such as Claude Shannon [1] laid the foundation for methods of designing communication systems such that errors which occur in transmission can be reduced to arbitrarily small probability. These methods are collectively known as *coding theory*.

In the terminology of digital communication systems, data is transmitted through a *channel*, and the components which embody coding theory are called the *channel encoder* and *channel decoder*, as shown in Figure 1. Therefore we will be discussing the design and simulation of these blocks in this tutorial.

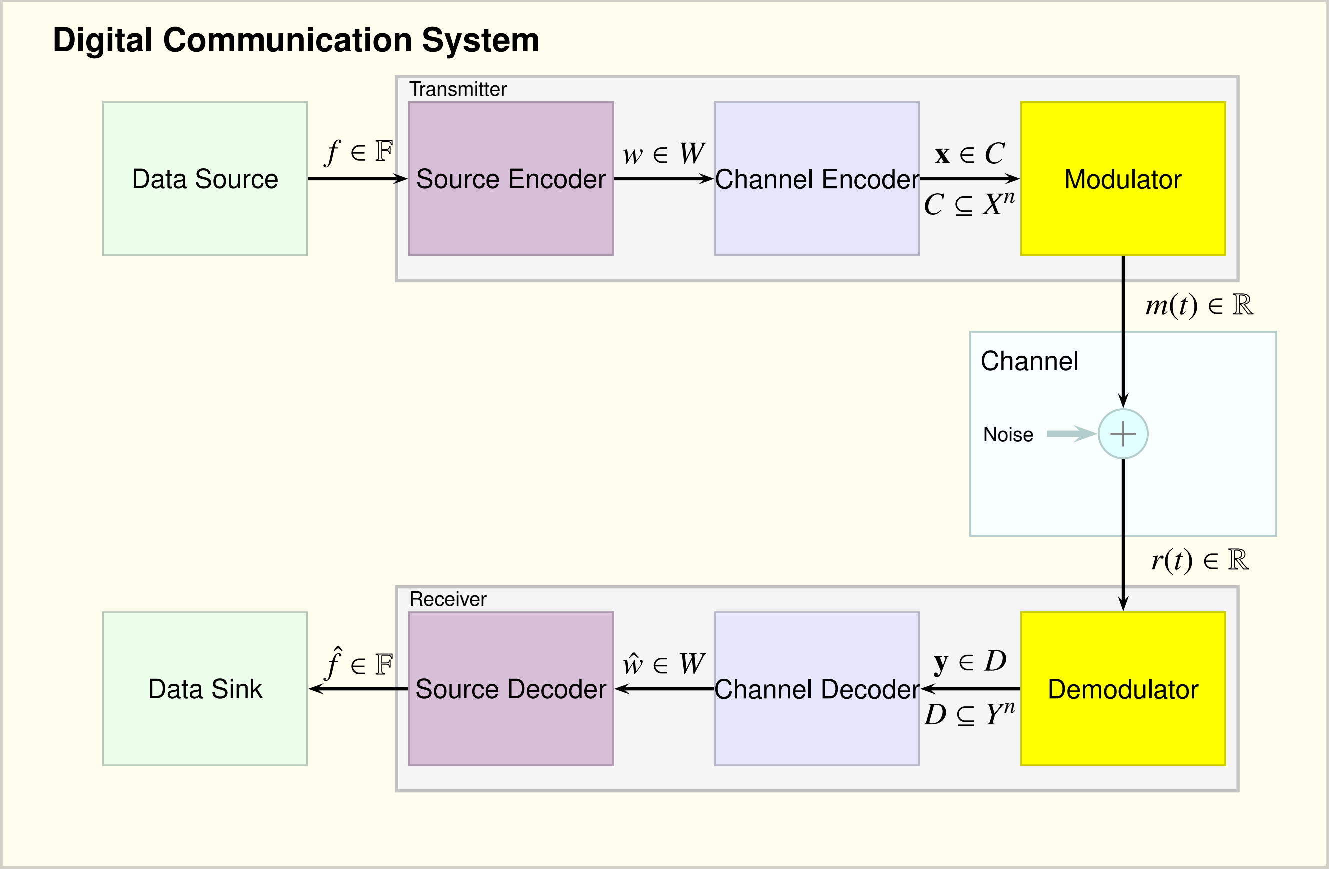

Figure 1: Digital Communication System Block Diagram

In this tutorial we define fundamental concepts of coding theory, show an example of a block code, and provide theoretical and simulated results of this code applied to a BPSK system.

1.2 Intended Audience

This tutorial assumes:

1. The reader has a working knowledge of linear algebra and general vector spaces ([2], [3]).
2. The reader has at least some familiarity with abstract (or modern) algebra ([3], [4]).
3. The reader has a working knowledge of probability theory ([5], [6]).

2 Theory

2.1 Elements of a Digital Communication System

Figure 1 is a block diagram of a digital communication system. The input to the system is a sequence of elements f[1] that we wish to transmit over the channel and reproduce at the output of the receiver as $\hat{f}$ with as little error as possible within a given set of constraints (power, bandwidth, computational complexity, latency, etc.).

The source encoder removes redundancy from the source samples by the use of compression. This is desirable since it reduces the average data rate required for the system, utilizes the channel more efficiently, and improves error correction. Functions such as A/D conversion and encryption are also sometimes performed in this block.

Source coding, while related to coding theory, is a separate topic and will not be covered in this tutorial. For more information, see the texts by Cover and Thomas [7] or Roman [8], or search for the following topics: information theory, entropy, Kraft's theorem, Huffman codes.

The channel encoder/decoder improve the reliability of the transmission of the source-encoded information through the channel. These two blocks jointly embody the topic of this tutorial.

The modulator generates a continous-time output $m(t)$ from the discrete-time channel encoder output suitable for the channel. It can include functions such as pulse-shaping, constellation mapping, line-coding, RF upconversion, etc.

The channel is the physical medium through which the message must be sent. Examples include an ethernet cable, the magnetic media and read/write heads of a disk drive, an RF channel, and the optical fiber and transmitter/receiver of a fiber-optic system.

Each block in the receiver performs the inverse function of the corresponding transmitter block. The final output of the receiver, $\hat{f}$, is an estimate of our original input source message f.

2.2 Coding Theory Basics and Block Codes

In order to understand what a block code is, we must first make some preliminary definitions. Note that the notation used in these definitions is consistent with that in Figure 1.

[1] Note that we omit the sequence index typically used in discrete-time systems, e.g., $f[n]$, with the understanding that all inputs and outputs correspond to a specific index.

Definition (Encoding Scheme) An **encoding scheme** is a bijective mapping $f\colon W \mapsto C \subseteq X^n$.

Definition (Decision Scheme) A **decision scheme** is a surjective mapping $g\colon D \subseteq Y^n \mapsto W$.

Definition (Discrete Memoryless Channel) A **discrete memoryless channel**[2] consists of an input alphabet $X = \{x_1, x_2, ..., x_q\}$ and an output alphabet $Y = \{y_1, y_2, ..., y_t\}$, where $X \subseteq Y$, and a set of **channel probabilities**, or **transition probabilities**, $p(y_j|x_i)$, satisfying

$$\sum_{j=1}^{t} p(y_j|x_i) = 1, \quad i = 1, 2, \ldots, q. \tag{1}$$

Think of a channel as sequentially accepting values from $C \subseteq X^n$ and outputting values from Y^n. Since the channel is memoryless, each symbol is independent from the other and therefore the probability $p(\mathbf{d}|\mathbf{c})$ that $\mathbf{d}$ is received given that $\mathbf{c}$ was sent is

$$p(\mathbf{d}|\mathbf{c}) = \prod_{i=1}^{n} p(d_i|c_i). \tag{2}$$

Definition (Block Code) Let $X = \{x_1, x_2, ..., x_q\}$ be a finite set, called a **code alphabet**, and let X^n be the set of all vectors of length n over X. Any nonempty subset C of X^n is called a **q-ary block code**. q is called the *radius* of the code. Each vector in C is called a **codeword**. If $C \subseteq X^n$ contains M codewords, then it is customary to say that C has **length** n and **size** M, and is an $(n, \log_q M)$-code. The **rate** of a q-ary $(n, \log_q M)$-code is

$$R = \frac{\log_q M}{n}. \tag{3}$$

We now know what a block code is. Why is it useful? Shannon's Noisy Coding Theorem provides the answer to that:

Theorem (Shannon's Noisy Coding Theorem) Consider a discrete memoryless channel with capacity $\mathcal{C}$. For any value $R < C, R \in \mathbb{R}^+$, there exists a sequence C_n of q-ary codes and corresponding decision schemes f_n with the following properties:

1. C_n is an $(n, \log_q |C_n|)$-code; that is, C_n has length n and rate at least $(\log_q |C_n|)/n$;
2. The maximum probability of error of f_n approaches 0 as n approaches infinity,

$$\lim_{n\to\infty} p_e^{max}(n) = 0. \tag{4}$$

2.3 Two Important Families of Codes

Definition (Systematic Code) A q-ary (n, k)-code is called a **systematic** code if there are k positions $i_1, i_2, \ldots, i_k$ with the property that, by restricting the codewords to these positions, we get all of the q^k possible q-ary words of length k. The set $\{i_1, i_2, \ldots, i_k\}$ is called the **information set**.

[2] Note that to make this definition consistent with the block diagram in Figure 1, the "channel" defined here includes the modulator, channel, and demodulator in the block diagram.

In other words, a systematic code is one in which the original message can be picked out of the codeword by looking at the appropriate positions. For example, consider the encoding of the message $\begin{bmatrix}0 & 1 & 2 & 3\end{bmatrix}$ over the alphabet from $\mathbb{Z}_4$ given by $\begin{bmatrix}0 & 3 & 2 & 0 & 3 & 1\end{bmatrix}$, where the information set is $\{4, 6, 3, 2\}$, and where $q = 4$, $n = 6$, and $k = 4$.

Theorem (Finite Fields) All finite fields have size p^n for some prime number p and $n \in N$. Furthermore, there is exactly one field (up to isomorphism) of size $q = p^n$, denoted by $\mathbb{F}_q$ or $GF(q)$.

The set $\mathbb{F}_q^n$ of all n-tuples with components from $\mathbb{F}_q$ is a vector space over $\mathbb{F}_q$ with dimension n.

Definition (Linear Code) A code $L \subseteq \mathbb{F}_q^n$ is a **linear code** if L is a subspace of $\mathbb{F}_q^n$. If L has dimension k over $\mathbb{F}_q^n$, we say that L is an **[n,k]-code**.

2.4 Other Important Concepts

2.4.1 Constructing a Decision Scheme

We've now defined the key components of a block code, including the decision scheme, but we haven't shown how a decision scheme can be constructed.

Definition (Weight) Let $\mathbf{y}$ be a vector from Y^n. The weight of $\mathbf{y}$, denoted $w(\mathbf{y})$, is the number of non-zero elements in the vector.

For example, let $\mathbf{y} = \begin{bmatrix}1 & 0 & 2 & 3 & 0 & 3 & 1\end{bmatrix} \in \mathbb{Z}_4^7$. Then $w(\mathbf{y}) = 5$.

Definition (Hamming Distance) Let $\mathbf{y}_1$ and $\mathbf{y}_2$ be two vectors from Y^n. The Hamming distance between $\mathbf{y}_1$ and $\mathbf{y}_2$, denoted $d(\mathbf{y}_1, \mathbf{y}_2)$, is $w(\mathbf{y}_1 - \mathbf{y}_2)$. Thus the Hamming distance can be thought of as the number of places in which two vectors differ.

Definition (Minimum Distance) Let C be a block code. The minimum distance of the code is the minimum Hamming distance between any two elements in the code.

Now consider a block code which has a minimum distance of 3: any two codewords will differ in at least three places. If a received codeword has a single error, the Hamming distance between it and the true codeword will be one, while the Hamming distance between it and any other codeword will be at least two. Thus we see that the Hamming distance can be used as a decision scheme for mapping received codewords to actual codewords C. Such a decision scheme is called *minimum-distance decoding*. The final mapping from a codeword in C to a message in W is performed by the inverse function $f^{-1} : C \mapsto W$, which is guaranteed to exist since f is bijective.

2.4.2 Quantifying Coding Performance

The performance of an error-correcting code is often quantified in terms of the *coding gain*:

Definition (Coding Gain) For a given modulation type and bit-error rate, coding gain is defined as the ratio of the SNR per bit required for the uncoded signal to the SNR per bit required for the coded signal.

This ratio is often given in units of decibels.

3 Example: A Soft-Decision Decoded Linear Systematic Code

3.1 Theory

Let the source alphabet be $X = \mathbb{Z}_2$ (a field). Consider the following matrix over $\mathbb{Z}_2$

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{5}$$

The rows of this matrix can be thought of as four vectors in $\mathbb{Z}_2^7$. Since these vectors are linearly independent, they span a 4-dimensional subspace of $\mathbb{Z}_2^7$ which we denote C. Hence the $k \times n$ G matrix can be thought of as a linear transformation from all vectors in $\mathbb{Z}_2^4$ to C, taking 1×4 source data vectors $\mathbf{w}$ to 1×7 codeword vectors $\mathbf{c} \in C$:

$$\mathbf{c} = \mathbf{w}G \tag{6}$$

Hence we see that the code represented by G is a **linear code**. The G matrix is referred to as the **generator matrix**.

We may partition the G matrix into a 4×4 identity matrix and a 3×4 G' matrix as follows:

$$G = \begin{bmatrix} I & | & G' \end{bmatrix}, \tag{7}$$

where G' is simply the right three columns of the original G matrix:

$$G' = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}. \tag{8}$$

The codeword then becomes

$$\begin{aligned} \mathbf{c} &= \mathbf{w} \times G \\ &= \mathbf{w} \times \begin{bmatrix} I & | & G' \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{w} & | & \mathbf{w} \times G' \end{bmatrix}, \end{aligned} \tag{9}$$

This shows that the code defined by this matrix is *systematic*, the source message being embedded in the left four symbols of the codeword.

After the codewords are formed from the message source, they are sent through the channel. Since the information may be corrupted on its way through the channel, the received codeword $\mathbf{c}'$ may no longer be in the subspace C. How could we detect this? If we could somehow come up with another linear transformation which had C as a kernel (or nullspace), then we would know that any received codeword that transformed to the zero vector would be correctly received.

Let H be the matrix

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{10}$$

Since the rows of H are linearly independent, H spans a 3-dimensional subspace of $\mathbb{Z}_2^7$. It is not hard to show that each row of H is also orthogonal to each row of G. This shows that the nullspace of H is at least a subset of G. Finally, since, for any linear transformation H,

$$\mathit{nullity}(H) + \mathit{rank}(H) = n \Longrightarrow \mathit{nullity}(H) = n - \mathit{rank}(H) = 4, \tag{11}$$

then the nullspace of H is precisely the space spanned by G. Thus we may conclude that

$$\mathbf{c} \in C \Longleftrightarrow \mathbf{c}H^T = \mathbf{0}. \tag{12}$$

For this reason, H is called the **parity check matrix** for the code C. The 1×3 vector that results from $\mathbf{c}H^T$ is referred to as the **syndrome**.

Suppose that sending the codeword **c** through the channel results in a single error in the ith position. Let $\mathbf{e}_i$ be the error vector with a 1 in the ith position so that

$$\mathbf{c}' = \mathbf{c} \oplus \mathbf{e}_i \tag{13}$$

Then the syndrome that results is

$$\begin{aligned} (\mathbf{c} \oplus \mathbf{e}_i)H^T &= \mathbf{c}H^T \oplus \mathbf{e}_i H^T \\ &= \mathbf{0} \oplus \mathbf{e}_i H^T \\ &= \mathbf{e}_i H^T \end{aligned} \tag{14}$$

But notice that the matrix H has been cleverly designed so that its i-th column, read from the top down, is just the binary representation of the number i. In other words, the vector $\mathbf{c}'H^T$ tells us the location of the error, where location zero means no error.

An example encoding of the message $\begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix}$:

$$\mathbf{c} = \mathbf{m}G = \begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \tag{15}$$

Let us introduce an error in the fourth digit, resulting in $\mathbf{c}' = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$. The syndrome is then

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{16}$$

This code is one of the family of *Hamming codes*.

3.2 Simulation

In this section we will apply the Hamming code of the previous section to a BPSK system using soft-decision decoding, simulate the system with and without coding, and determine the coding gain.

From the previous section, $W = X^k$ and $C \in X^n$, where $X = \mathbb{Z}_2$, $k = 4$, and $n = 7$. In other words, the source symbols from the source encoder are binary and are blocked into vectors of four symbols each. The encoding scheme then maps each element from $\mathbb{Z}_2^4$ to $C \in \mathbb{Z}_2^7$ by multiplying by the generator matrix G as discussed above.[3]

The BPSK modulator transforms the codewords' input alphabet from $\{0, 1\}$ to $\{-1, +1\}$ and sends the words over the channel. Each element of each vector is transmitted sequentially since this is BPSK, and therefore each element of each vector is corrupted by the noise of the channel.

Soft-decision decoding ([9], section 8.1.4) combines the demodulator and the channel decoder by formulating the so-called correlation metrics. The simulation code is shown in Figure 3.

Probability of bit errors were simulated and computed for the uncoded system in order to verify the correctness of the simulation. The well-known result for BPSK is given by the equation

$$P_e = Q\left(\sqrt{\frac{2E_b}{N_o}}\right). \tag{17}$$

The simulation code is shown in Figures 4 and 5, respectively.

The simluation results are shown in Figure 2.

When E_b/N_o is in dB, the coding gain is defined as

$$G_c = \left(\frac{E_b}{N_o}\right)_{\text{uncoded}} - \left(\frac{E_b}{N_o}\right)_{\text{coded}}. \tag{18}$$

At a BER of approximately 2×10^{-4}, $G_c = 8 - 6.4 = 1.6$ dB. At a BER of approximately 1×10^{-2}, $G_c = 4.4 - 3.4 = 1$ dB. Thus we see that the coding gain is a function of the BER.

To summarize, we find that even a short, simple Hamming code provides a coding gain of about 1.5 dB at moderate bit-error rates. This allows us to use approximately 30 percent less transmit power to achieve the same BER as an uncoded system.

4 For Further Study

5 Terms, Abbreviations, and Notation

BPSK Binary PSK.

Mappings:

[3] Note that this multiplication has to occur over the field Z_2 (mod 2).

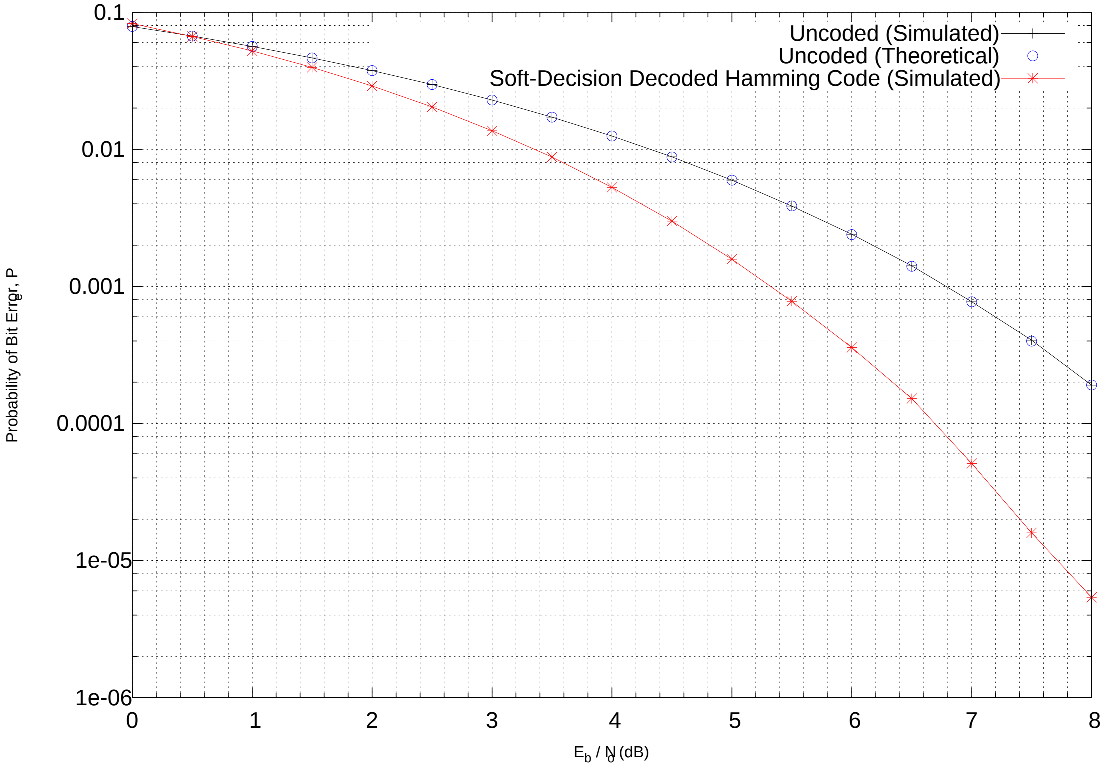

Figure 2: Simulation Results

1-1 A mapping $f\colon G \mapsto H$ is “1-1” (one-to-one) if $f(g_1) = f(g_2) \Rightarrow g_1 = g_2$.

bijective mapping 1-1 and onto.

injective mapping 1-1.

onto A mapping $f\colon G \mapsto H$ is “onto” if $f(G) = H$.

surjective mapping onto.

PSK Phase Shift Keying.

RF Radio Frequency.

SNR Signal-to-Noise Ratio.

6 Revision History

Table 1 lists the revision history for this document.

Rev.	Date/Time	Person	Changes
PA1	19-Aug-2009	Randy Yates	Initial Version
PA2	27-Aug-2009	Randy Yates	Changed font size to 11 points.

Table 1: Revision History

7 References

[1] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the Institute of Radio Engineers*, vol. 37, pp. 10–21, 1949.

[2] Carl D. Meyer, *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, 2000.

[3] M. Artin, *Algebra*. Prentice Hall, 1991.

[4] I. Herstein, *Topics in Algebra*, 2nd ed. Wiley, 1975.

[5] Athanasios Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. WCB/McGraw-Hill, 1991.

[6] Alberto Leon-Garcia, *Probability and Random Processes for Electrical Engineering*. Addison-Wesley, 1989.

[7] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley and Sons, Inc., 1991.

[8] S. Roman, *Coding and Information Theory*. Springer, 1992.

[9] J. G. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 2001.

Appendices

```
function ber = simcoded(ebno, N)
%ber = simcoded(ebno, N)
% FUNCTION: Simulated Soft-Decision Decoding Bit Error Rate of Hamming-Coded Coherent Binary PSK
% AUTHOR: Randy Yates
% INPUT PARAMETERS:
%     ebno = E_b / N_o (may be vector)
%     N    = number of data values to simulate. note that N should be a multple of 4
% OUTPUT PARAMETERS:
%     ber = bit-error rate result of the simulations
% DESCRIPTION:
% CREDITS:
%     From section 8.1.4 of
%     @BOOK{proakiscomm,
%       title = "{Digital Communications}",
%       author = "John~G.~Proakis",
%       publisher = "McGraw-Hill",
%       edition = "fourth",
%       year = "2001"}

fprintf(1,'Begin Coded Simulation\n');

M = length(ebno);

% Adjust ebno for code rate
ebno = ebno .* 4/7;

% force N to be a multiple of 4
K = ceil(N / 4);
N = 4 * K;

% Specify the generator matrix
G = [1 0 0 0 0 1 1;...
     0 1 0 0 1 0 1;...
     0 0 1 0 1 1 0;...
     0 0 0 1 1 1 1 ];

% Generate a 16x4 matrix of all possible row vectors over Z_2
W = zeros(16, 4);
for l = 0:15
  W(l+1, 4) = mod(l,2);
  W(l+1, 3) = mod(floor(l/2),2);
  W(l+1, 2) = mod(floor(l/4),2);
  W(l+1, 1) = mod(floor(l/8),2);
end

% Generate all possible codewords
C = mod(W * G, 2); % 16x7
Cm = 2 * C - 1; % 16x7 codeword matrix for correlating channel outputs with elements in {-1, +1}.

ber = zeros(1, M);

for m = 1:M
  fprintf(1,'  Eb/No = %f (%d of %d)\n', ebno(m), m, M);
  % produce a string of 1's and 0's
  w = round(rand(K, 4));

  % generate the codewords
  c = mod(w * G, 2);

  % generate "modulator" output m \in {-1, +1}
  modulated = 2 * c - 1;

  % compute variance of noise required based on passed ebno
  var = 1 / (2 * ebno(m));

  % simulate passing through channel by adding zero-mean Gaussian noise of specified variance:
  ch = modulated + randn(K,7) * sqrt(var);

  % compute correlation metrics and perform soft-decision decoding
  cm = ch * Cm'; % Kx16 array of metrics for each received codeword

  % Find max - note max() operates column-wise on an array
  cmmax = max(cm')'; % Kx1 array of row-wise max values in cm

  % decode
  cmmaxidx = (cm == repmat(cmmax, 1, 16)); % Kx16, with a single 1 in each row indicating the decoded element
  what = cmmaxidx * W;

  % compute bit error rate:
  ber(m) = sum(sum(what ~= w)) / N;
end

fprintf(1,'End Coded Simulation\n');
```

Figure 3: Coded BPSK Simulation MATLAB M-File

```
%-*- mode:matlab -*-
function [ber, data, rxdata] = simuncoded(ebno, N)
%[ber, data, rxdata] = simuncoded(ebno, N)
% FUNCTION: Simulated Uncoded Bit Error Rate of Coherent Binary PSK
% AUTHOR: Randy Yates
% INPUT PARAMETERS:
%     ebno = E_b / N_o
%     N    = number of data values to simulate
% OUTPUT PARAMETERS:
%     ber = bit-error rate result of the simulations
% DESCRIPTION:
% CREDITS:

fprintf(1,'Begin Uncoded Simulation\n');

M = length(ebno);
ber = zeros(1, M);

for m = 1:M
  fprintf(1,'  Eb/No = %f (%d of %d)\n', ebno(m), m, M);
  % produce a string of 1's and -1's
  data = 2 * (rand(N, 1) > 0.5) - 1;

  % compute variance of noise required based on passed ebno
  var = 1 / (2 * ebno(m));

  % simulate passing through channel by adding zero-mean Gaussian noise of specified variance:
  chdata = data + randn(N,1) * diag(sqrt(var));

  % "slice" the receiver input:
  rxdata = 2 * (chdata > 0) - 1;

  % compute bit error rate:
  ber(m) = sum(rxdata ~= data) / N;
end

fprintf(1,'End Uncoded Simulation\n');
```

Figure 4: Uncoded BPSK Simulation MATLAB M-File

```
function ber = comuncoded(ebno)
%ber = comuncoded(ebno)
% FUNCTION: Theoretical (computed) Uncoded Bit Error Rate of Coherent Binary PSK
% AUTHOR: Randy Yates
% INPUT PARAMETERS:
%     ebno = E_b / N_o
% OUTPUT PARAMETERS:
%     ber = bit-error rate result of the simulations
% DESCRIPTION:
% CREDITS:

ber = q(sqrt(2 * ebno));
```

Figure 5: Uncoded BPSK Theoretical Computation MATLAB M-File