NONBINARY CONVOLUTIONAL CODING FOR MULTIMEDIA DATA TRANSMISSION

Merin Savaş

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical and Electronic Engineering

Eastern Mediterranean University
September 2009, Gazimağusa

Approval of the Institute of Graduate Studies and Research

____

Prof. Dr. Elvan Yılmaz
Director (a)

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Sciences in Electrical and Electronic Engineering.

____

Assoc. Prof. Dr. Aykut Hocanın
Chair, Department of Electrical and Electronic Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Sciences in Electrical and Electronic Engineering.

____

Assoc. Prof. Dr. Aykut Hocanın
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Hüseyin Bilgekul ________________________________
2. Assoc. Prof. Dr. Hasan Demirel ________________________________
3. Assoc. Prof. Dr. Aykut Hocanın ________________________________

ABSTRACT

NONBINARY CONVOLUTIONAL CODING FOR MULTIMEDIA DATA TRANSMISSION

Keywords: Error Control coding, Convolutional codes, Source metric

In this thesis, the performance of nonbinary convolutional coding technique is investigated and new nonbinary codes with better performance are proposed. Nonbinary convolutional coding technique is a coding technique which is similar to the binary convolutional codes with the same decoding strategy but they are designed for general nonbinary sources. The nonbinary convolutional coding technique is described and simulated under various channel conditions. Synthetic nonbinary source sequences are produced by using Markov processes.

The channels used in the experimental simulations include additive white Gaussian noise (AWGN) and flat fading channel models. Coded image and video sequences are transmitted over the channels by using binary phase shift keying (BPSK) modulation technique. The Viterbi decoding algorithm is used for decoding the encoded sequences. Viterbi decoding employs hard and soft decision metrics. The soft metric is updated to include the source statistics. This enables the decoder to use the source redundancy for improved decoding performance. Results show that nonbinary convolutional coding which uses the source statistic is effective in reducing bit error rate (BER). New nonbinary convolutional coding is optimized to

increase the code distance (d_{free}). The optimized codes are shown to perform better at low BER.

ÖZET

Bu tezde, ikili olmayan evrişimsel kodlama tekniği incelenmiş ve daha yüksek başarımlı kodlar önerilmiştir. İkili olmayan evrişimsel kodlama tekniğinde, geleneksel evrişimsel kodlama tekniği ile aynı yöntem kullanılarak kod çözülmektedir. İkili olmayan kodlama, genel ikili olmayan veri kaynakları için tasarlanmakta ve bu tür kaynaklar için daha yüksek başarım sağlamaktadır. Bu tezde ikili olmayan evrişimsel kodlar incelenmiş ve farklı iletişim kanallarındaki çoklu-ortam veri iletim başarımı benzetimlerle gösterilmiştir. Sentetik ikili olmayan veri kaynakları Markov süreçleri kullanılarak üretilmiş ve hata düzeltme başarımı geliştirilmiş kodlar benzetimlerle gösterilmiştir.

Yapılan deneylerde, kanal modelleri olarak Toplanır Beyaz Gauss Gürültü (TBGG) ve düz sönümlemeli kanal modelleri kullanılmıştır. Kodlanan resim ve video verileri kanal üzerinden İkili Evre Kaydırmaları Anahtarlama (İEKA) modülasyon tekniği kullanılarak iletilmiştir. Kodlanan dizileri çözmek için Viterbi kod çözme algoritması kullanılmıştır. Viterbi algoritması kod çözme işleminde ikili veya yumuşak ölçüt kullanmaktadır. Önerilen kodlarda, yumuşak ölçüt, kaynak istatistiklerini de kullanarak güncellenmiştir. Bunun sonucu olarak kod çözücü kaynak artıklığını da kullanarak başarımını artırmıştır. Sonuçlar ikili olmayan evrimşimsel kodlamada kaynak istatistiğinin kullanımının bit hata oranını düşürmede etkili olduğunu göstermektedir.

Dedicated to my husband Cafer Elgin

ACKNOWLEDGEMENTS

I would like to express my profound appreciation to Assoc. Prof. Dr. Aykut Hocanın, my supervisor, for his invaluable support, encouragement and patience during the development of this thesis. It has been a pleasure to work under his supervision.

Special thanks to my husband Cafer Elgin for him love and sharing hard times with me and also I would like to thank Mr. Ünal Fındık, Mr. Çağrı Özçınar and Mr. Gholamreza Anbarjafari (Shahab) for their help and support. I would never forget their encouragement during this study.

Special thanks also go to my family, for their support and belief in me throughout all these years and this thesis. They supported me in every step of my education and they have made lots of sacrifices for me to have a better future.

TABLE OF CONTENTS

CHAPTER 1 1

INTRODUCTION 1

CHAPTER 2 3

ERROR CONTROL CODING TECHNIQUES 3

2.1 Automatic Repeat Request 3

2.2 Forward Error Correction 4

2.2.1 Linear Block Codes: 4

2.2.2 Turbo Codes 5

2.2.3 Convolutional Codes 5

2.2.3.1 Encoding for Convolutional Codes 6

2.2.3.2 Convolutional Codes for Decoding 7

2.2.3.3 Trellis Diagram 8

2.2.3.4 Viterbi Algorithm 9

2.2.3.5 Convolutional Code Performance Measures 11

CHAPTER 3 13

NONBINARY CONVOLUTIONAL CODES 13

3.1 Nonbinary Convolutional Coding 13

3.1.1 The Design Criterion 14

3.2 Markov Chain 16

3.3 Entropy 18

3.4 Wireless Communication Channel 18

3.4.1 Additive White Gaussian Noise Channel 19

3.4.2 The Flat Fading Channel ... 20
CHAPTER 4 ... 22
SIMULATION RESULTS ... 22
4.1 Simulation Setup ... 22
4.2 Performance study for simulated data ... 24
4.3 Simulation Results for Different Code Distances ... 27
4.4 Performance study using images ... 29
4.5 Simulation results for video sequence ... 37
4.6 Entropy Calculation Results ... 38
CONCLUSIONS AND FUTURE WORK ... 40
REFERENCES ... 42

LIST OF FIGURES

Figure 2.2: An example state diagram 7

Figure 2.3: Encoder for a rate *R=1/3* convolutional code. 8

Figure 2.4: State diagram for encoder in figure 2.3 9

Figure 2.5: Trellis diagram for the encoder in figure 2.3... 9

Figure 2.6: State diagram for d_{free} =2 12

Figure 3.1: Rate *R*=1/3 NCC encoder structure. [6] 14

Figure 3.2: Block Diagram of a General Communication System 19

Figure 3.3: AWGN Channel 19

Figure 4.1: Binary data performance of convolutional coding (*K*=3, AWGN, *m*=2, *g*=[7,5]) transition probabilities:[0.3 0.7;0.7 0.3]).................................... 24

Figure 4.2: Nonbinary convolutional coding performance for data (d_{free} =1, AWGN, *R*=2/4, *K*=5, *m*=4) 25

Figure 4.3: Nonbinary convolutional coding performance for data (d_{free} =2, AWGN, *R*=2/6, *K*=5, *m*=4) 26

Figure 4.4: Performance of nonbinary convolutional coding with different code distances. 27

(d_{free} =1 and d_{free} =2, AWGN, *R*=2/4, *K*=5, *m*=4) 27

Figure 4.5: Performance of nonbinary convolutional coding with different code distances. 28

(d_{free} =1 and d_{free} =2, flat fading, *R*=2/4, *K*=5, *m*=4) 28

Figure 4.6: Performance of nonbinary convolutional coding with different code distances. 29

(d_{free} =1 and d_{free} =2, AWGN, 392 x 294 pixels image, *R*=2/4, *K*=5, *m*=4).............. 29

Figure 4.7: (a) Transmitted Image, (b) Reference image with source statistics similar to transmitted image, and (c) Reference image with source statistics different from transmitted image.. 30

Figure 4.8: Nonbinary convolutional coding performance for image (*K*=5, AWGN, d_{free} =1, gray scale 392 x 294 pixels, *m*=4, *K*=5)..................................... 31

Figure 4.9: Nonbinary convolutional coding performance for image with using different image probabilities (d_{free} =1, AWGN, gray scale 392 x 294 pixels and gray scale 392 x 294 pixels, *R*=2/4, *K*=5, *m*=4)....................... 32

Figure 4.10: Nonbinary convolutional coding performance for image with using different image probabilities (d_{free} =1, AWGN, gray scale 372 x 270 pixels and gray scale 392 x 294 pixels, *R*=2/4, *K*=5, *m*=4)................................. 33

Figure 4.11: Binary image performance of convolutional coding (*K*=3, AWGN, *m*=2, *g*=[7,5])... 34

Figure 4.12: Nonbinary convolutional coding performance for image (flat fading, d_{free} =2, gray scale 372 x 270 pixels, *R*=2/4, *K*=5, *m*=4) 35

Figure 4.13: Nonbinary convolutional coding performance for image (AWGN, d_{free} =2, gray scale 816 x 612 pixels, *R*=2/6, *K*=5, *m*=4) 36

Figure 4.14: Nonbinary convolutional coding performance for video sequence (AWGN, d_{free} =2, 10 consecutive frames, gray scale 120 x 177 pixels/frame, *R*=2/4, *K*=5, *m*=4).. 37

LIST OF TABLES

Table 4.1: Entropy and BER performances of soft decision with source metric
(R=2/4, m=2, K=3, TP 1-4) 39

LIST OF SYMBOLS/ABBREVIATIONS

ARQ	Automatic Repeat Request
AWGN	Additive White Gaussian Noise
BPSK	Binary Phase Shift Keying
BER	Bit Error Rate
FEC	Forward Error Correction
NACK	Negative acknowledgement
NCC	Nonbinary Convolutional Coding
SNR	Signal to Noise Ratio
PSD	Power Spectral Density
SN	Sequence Number
B	Bandwidth
d_{free}	Minimum free distance
S	Entropy
R	Rate

CHAPTER 1

INTRODUCTION

Even though data communication methodologies have been developing considerably, errors still occur during data transmission. Error detection/correction is a very important task in any transmission protocol. It provides the way to protect data from errors, and maintain data integrity. There are many types of error correcting codes such as: Linear Block Codes, Cyclic Codes, Convolutional Codes, as well as the retransmission strategies such as the Automatic Repeat Request (ARQ), etc.

Forward error correction (FEC) is an error correction technique that improves the capacity of a channel by adding some carefully designed redundant information to the data being transmitted through the channel. The process of adding this redundant information is known as channel coding. Convolutional coding and block coding are two major forms of channel coding. Convolutional codes operate on serial data, one or more bits at a time. Block codes operate on relatively large (typically, up to a couple of hundred bytes) message blocks. There are a variety of useful convolutional and block codes, and a variety of algorithms for decoding the received coded information sequences to recover the original data. Convolutional encoding with Viterbi decoding is a FEC technique that is particularly suited to a channel in which the transmitted signal is corrupted mainly by additive white Gaussian noise (AWGN). Viterbi decoding is one of two types of decoding algorithms used with convolutional encoding. The other main type is the sequential decoding. Sequential decoding has the advantage that it can perform very well with long-constraint-length convolutional codes, but it has a variable decoding time. Viterbi decoding is optimal and has the advantage that it has a fixed decoding time. It is well suited to hardware decoder implementation.

In the thesis, beyond binary convolutional coding, nonbinary convolutional coding is investigated. Besides hard and soft decision convolutional coding, soft decision convolutional coding with source metric is investigated. Rate of encoder is changed from 1/2 to 1/3 and change in performance is observed. On the other hand, minimum free distance is changed and its effect is observed. Behaviors of independent and dependent data sources are investigated in terms of entropy and relationship between entropy of the data source and performance of soft decision decoding with source metric is observed. In addition to synthetic data sources, images and video sequences are also applied to the simulations and performances are discussed.

The thesis is organized as follows; In Chapter 2, an overview of convolutional coding is given. General information about nonbinary convolutional codes is introduced in Chapter 3. Moreover, Markov Chain, which is used to model the source, and the wireless channels such as AWGN and flat fading channels are described. Simulation results are presented in Chapter 4. Finally, Chapter 5 summarizes the thesis and identifies areas for future research.

In this thesis as novelty; source metric is integrated into soft decision convolutional coding to improve performance. A new coding scheme is proposed to maintain increased code distance. A new 1/3 rate code is proposed and the performance is observed over fading channel.

CHAPTER 2

ERROR CONTROL CODING TECHNIQUES

A digital communication system is a means of transporting information from transmitter to receiver while channel imposes errors on the transmitted data. Error control codes are used for preventing errors in these transmissions. Different codes are selected to perform in various applications with different requirements. Some typical coding strategies are given below:

2.1 Automatic Repeat Request

ARQ is a simple and commonly used method in error correction. In ARQ systems, firstly errors are detected at the receiver part. Then, errors are discarded and retransmitted if any packet is detected in error. There are three types of ARQ; Stop and Wait, Go-Back-*N* and Selective Repeat Request ARQ. Stop and Wait ARQ is the simplest ARQ procedure. The principle of this type of ARQ system is to complete the transmission of each packet correctly before moving to the transmission of next one. So, it spends time for acknowledgement. It is inherently inefficient and requires new strategy which is Go-Back-*N*. Here, several successive packets can be sent without waiting for the next packet to be requested. After that, Go-Back-*N* becomes quite ineffective for communication systems with high data rates. Selective Repeat Request is used which is similar to Go-Back-*N* except that in Selective Repeat Request only the error frame is retransmitted. [24]

The ARQ method needs duplex arrangement as part from the conventional transmitter to receiver signal, the request signal is to travel from receiver to transmitter. To request and transmit the corrupted data upon the requirement from the receiver has been used very successfully for non-real-time data transmission. For solving this problem Forward Error Correction (FEC) method is introduced. This method needs simplex arrangement as the signal has to travel only from the transmitter to the receiver. Retransmission of data is not necessary in this method. In this method, the channel encoder systematically adds digits to the transmitted message digits which is known as redundancy bits. Although these additional digits convey no new information, they make it possible for the channel decoder to detect, and correct errors in the information bearing digits. The overall probability of error is reduced due to error detection and/or correction. Forward error correction can also be used together with ARQ to improve the performance of ARQ system. With this hybrid system, since the received error containing messages are corrected, the number of re-requests will be reduced decreasing time delay of ARQ system. Forward error correction method will be explained below.

2.2 Forward Error Correction

There are three types of forward error correcting codes. They are linear block codes, turbo codes and convolutional codes.

2.2.1 Linear Block Codes:

Binary information sequence is divided into a fixed length message blocks. These message blocks consist of k information bits and there are total 2^k distinct messages. The encoder, according to certain rules, transforms each input message to the codeword and there are 2^k code words. This set of 2^k code words is called a block code. Code word consists of message part and the redundant checking part. Redundant checking part consists of *n*-*k* parity check digits, which are linear, and some of the information digits and the message part is formed by *k* information bits. The encoded message is;

$$v = u \cdot G \tag{2.1}$$

where *G* is the generator matrix and *u* is the message.

Minimum distance determines random error detection and random error correcting capabilities of the code. Minimum distance of a block code is minimum Hamming distance between all distinct pairs of code words.

2.2.2 Turbo Codes

Turbo codes are a combination of two or more error control codes in serial or parallel. The information bits are interleaved between the two encoders. These are then multiplexed with the uncoded information bits. A priori information is used in decoding stage.

2.2.3 Convolutional Codes

A convolutional code is a type of error-correcting code in which each *m*-bit information symbol to be encoded is transformed into an *n*-bit symbol, where *m*/*n* is the code *rate* ($n \geq m$) and the transformation is a function of the last *k* information symbols, where *k* is the constraint length of the code. [2]

2.2.3.1 Encoding for Convolutional Codes

Convolutional Codes are widely used to encode digital data before transmission through noisy channels. The encoder has memory and the encoder outputs at any given time unit depend not only on the inputs at that time unit but also on some number of previous inputs. An encoder with *k* input bits and *n* output bits have a rate *k/n*. Information bits are divided into blocks with length *k* and these blocks are then mapped into the code words with length *n.* This operation is done independent of the length *k*.

Generator sequences are one way to characterize the encoder structure of convolutional codes. Generator sequences are obtained by applying impulses into the system. After obtaining the generator sequences, these sequences and input sequences are convolved to produce the encoded sequences. All operations are modulo-*2*. Generator equation is:

$$v_l^{(j)} = \sum_{i=0}^{m} u_{l-i} g_i^j = u_l g_0^j + u_{l-1} g_1^j + ... + u_{l-m} g_m^j,\ j = 0,1,... \tag{2.2}$$

In this way, input sequences are encoded. Encoded information sequences are then multiplexed into a single sequence called a codeword for transmission over the channel. The codeword is given by:

$$v = \left(v_0^{(0)} v_0^{(1)}, v_1^{(0)} v_1^{(1)}, v_2^{(0)} v_2^{(1)}, ... \right) \tag{2.3}$$

Also, encoding can be written in a matrix form. Matrix form for encoding is:

$$v = uG \tag{2.4}$$

Another encoding system of convolutional coding is by using the state diagram. It contains memory elements and these contents determine a mapping between the next set of input bits and output bits. Also, this state diagram is time-invariant. There are 2^k branches leaving each state in the state diagram and the states are shown as $S_0, S_1, ..., S_{2^{v-1}}$. State branches are shown as *X*/*YY*, where *X* is the input bit and *YY* is the output bits. Note that, the memory contents are the reverse of the binary representation of the state number. Example of state diagram is in figure 2.2.

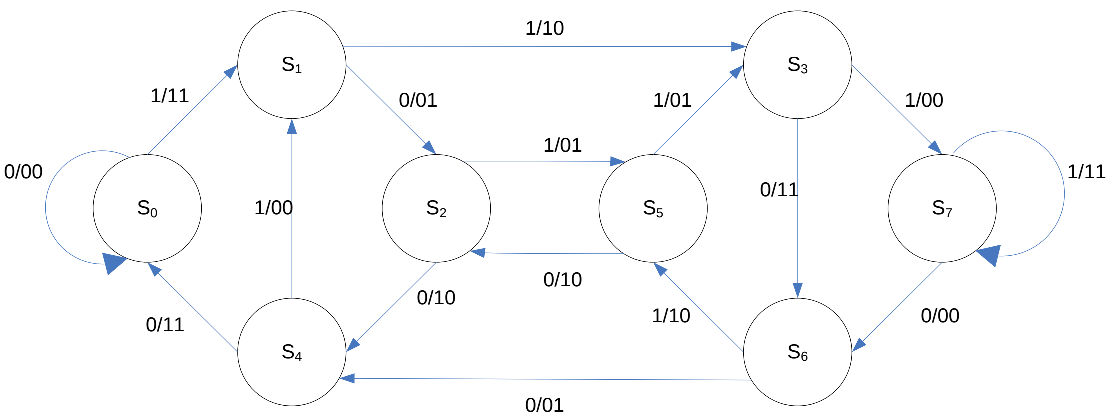

Figure 2.2: An example state diagram

Assume that information sequence $u = (1011000)$ will be encoded with rate *R*=1/2. Loop starts from S_0 and finishes S_0. Input bits will be checked looking at the state diagram. First bit of information sequence is 1. Looking at the state diagram with starting from S_0, output bits will be 11 and state will be S_1. Next input bit is 0, so S_2 will be next state and the output bits will be 01. These operations continue until the end of information sequence. Results will be encoded of the sequence.

2.2.3.2 Convolutional Codes for Decoding

Viterbi algorithm is quick operation for decoding convolutional codes. It shows state diagram of the encoder in time and each time unit is represented with separate state diagram. The resulting structure of Viterbi is a trellis diagram.

2.2.3.3 Trellis Diagram

The trellis diagram is an extension of a convolutional code's state diagram that explicitly shows the passage of time. Example of trellis diagram will be shown in figure 2.5 for figure 2.3. Two adjacent states are connected by branches and trellis diagram branches are labeled with the output bits. These output bits are associated to the state transitions. Consider a general (n,k) binary convolutional encoder with total memory *M* and maximal memory order *m*. The associated trellis diagram has 2^M nodes at each stage or time increment *t*. There are 2^k branches leaving each node and one branch for each possible combination of input values. Also, there are 2^k branches entering to each node. Given an input sequence of kL (*k* is number of input and *L* is length of each input), the trellis diagram must have $L+m$ stages. The first stage is starting and last stage is stopping. In addition to this, there are 2^{kL} distinct paths through the general trellis. Each one of these paths corresponds to a convolutional code word of length $n(L+m)$.

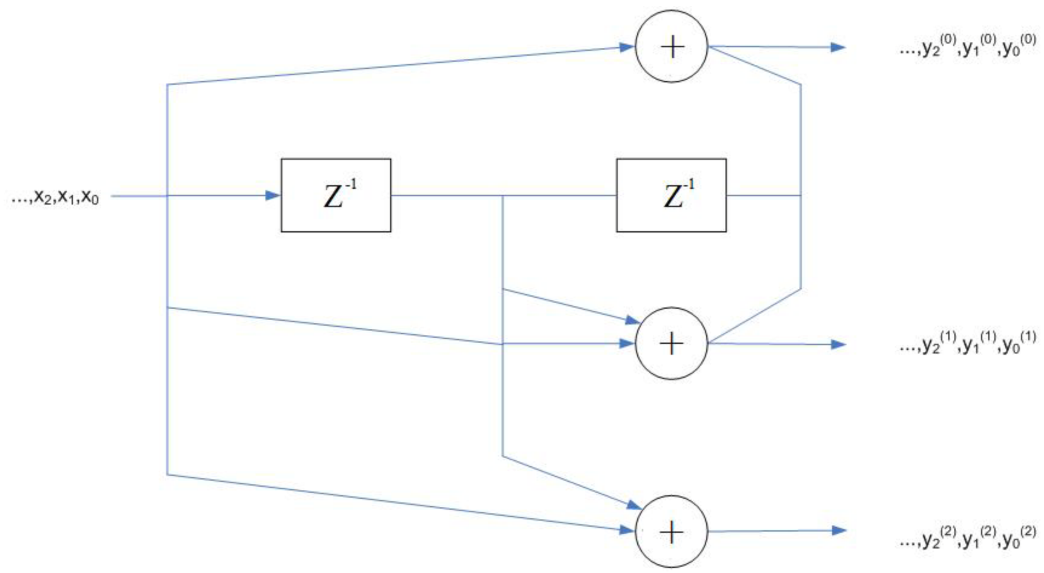

Figure 2.3: Encoder for a rate *R=1/3* convolutional code.

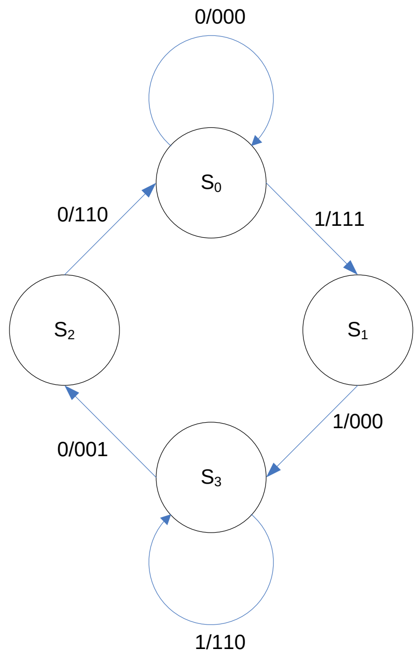

Figure 2.4: State diagram for encoder in figure 2.3

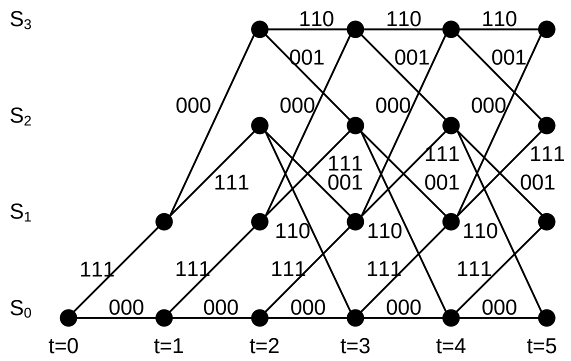

Figure 2.5: Trellis diagram for the encoder in figure 2.3

2.2.3.4 Viterbi Algorithm

Information sequence is encoded and then transmitted and it is corrupted by noise. Received sequence is decoded with Viterbi in convolutional codes. The maximum likelihood decoder selects for estimate the transmitted sequence. This type of decoder maximizes the probability of $p(r/y')$ which are, *r* is received and *y'* is the estimated sequence.

Assume that, this information sequence ***x*** is composed of *L* *k*-bit blocks and this encoded code is labeled as ***y***. At the end of encoding the output sequence consists of *L* *n*-bit blocks. The number of blocks is enlarged by *m* (long shift register) blocks. This sequence is transmitted and corrupted by noise. The received sequence is named as ***r***. This sequence estimated *y'* by decoder and the decoder generates a maximum likelihood.

Formulas of *x, y, r, y'* are:

$$x = \left(x_0^{(0)}, x_0^{(1)}, ..., x_0^{(k-1)}, x_1^{(0)}, x_1^{(1)}, ..., x_1^{(k-1)}, ..., x_{L-1}^{(k-1)} \right) \quad (2.5)$$

$$y = \left(y_0^{(0)}, y_0^{(1)}, ..., y_0^{(n-1)}, y_1^{(0)}, y_1^{(1)}, ..., y_1^{(n-1)}, ..., y_{L+m-1}^{(n-1)} \right) \quad (2.6)$$

$$r = \left(r_0^{(0)}, r_0^{(1)}, ..., r_0^{(n-1)}, r_1^{(0)}, r_1^{(1)}, ..., r_1^{(n-1)}, ..., r_{L+m-1}^{(n-1)} \right) \quad (2.7)$$

$$y' = \left(y'^{(0)}_0, y'^{(1)}_0, ..., y'^{(n-1)}_0, y'^{(0)}_1, y'^{(1)}_1, ..., y'^{(n-1)}_1, ..., y'^{(n-1)}_{L+m-1} \right) \quad (2.8)$$

path metric equation is:

$$M\left(r/y'\right) = \sum_{i=0}^{L+m-1} \left(\sum_{j=0}^{n-1} M\left(r_i^{(j)} \big/ y'^{(j)}_i \right) \right) \quad (2.9)$$

*k*th branch metric: The *k*th branch metric is the sum of the bit metrics for the *k*th block of *r* given *y'*.

$$M\left(r/y'\right) = \sum_{j=0}^{n-1} M\left(r_k^{(j)} \big/ y'^{(j)}_k \right) \quad (2.10)$$

*k*th partial branch metric: The *k*th partial branch metric is sum of the branch metrics for the first *k* branches.

$$M^k\left(r/y'\right) = \sum_{i=0}^{k-1} \left(\sum_{j=0}^{n-1} M\left(r_i^{(j)} \big/ y'^{(j)}_i \right) \right) \quad (2.11)$$

Trellis diagram is used for computation of path metrics in Viterbi. Each node in the trellis is assigned a number and these numbers are the partial path metric of the path. This path starts from state S_0 at time $t = 0$ and finishes at that node and best partial path metric is selected between all entering paths.

2.2.3.5 Convolutional Code Performance Measures

Performance of convolutional code depends on which decoding algorithm was used and the distance properties of the code itself. There are several techniques for performance measure of convolutional codes. These techniques for performance measures are: column distance function, minimum distance and minimum free distance. But the most important distance measure for a convolutional code is minimum free distance d_{free}. Minimum free distance d_{free}, is the minimum distance between any two length codeword in the code. Mathematical definition of minimum free distance d_{free} is:

$$d_{free} = \min\left\{d\left(v', v''\right) : u' \neq u''\right\} \tag{2.12}$$

Where v' and v'' are the codeword corresponding to the information sequences u' and u'', respectively. The performances of coding methods are investigated using different code distance values. As described above, distance between any two codeword in the code is d_{free}. Increasing d_{free} increases distance between labels enabling decoder to make better decisions while decoding. In the simulations, d_{free} is increased from 1 to 2 and performances are compared with each other. Figure below shows transition between labels while d_{free} =2.

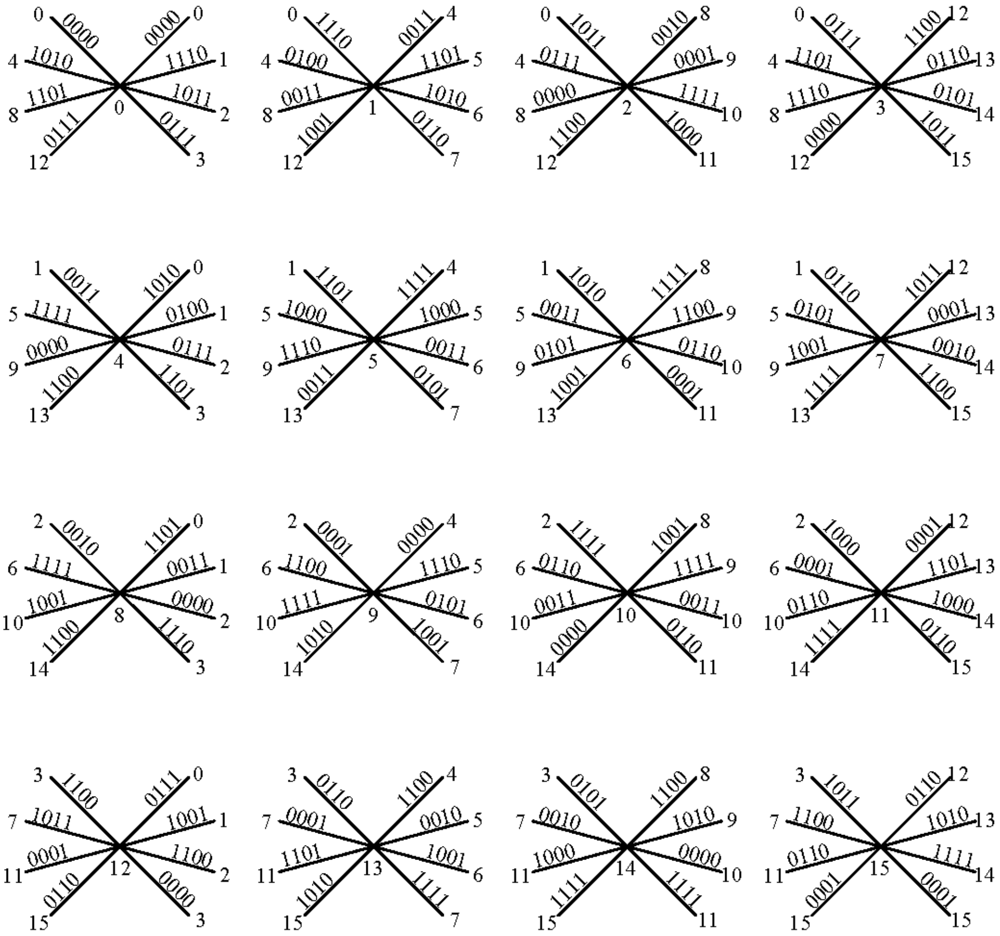

Figure 2.6: State diagram for d_{free} =2

Decoding algorithm also affect the performance of the convolutional codes. Viterbi and Behl, Cocke, Jelmek and Raviv algorithms are used as decoding algorithms. Viterbi algorithm is the most optimal and is simpler to implement amongst the decoding algorithm techniques.

CHAPTER 3

NONBINARY CONVOLUTIONAL CODES

3.1 Nonbinary Convolutional Coding

Nonbinary convolutional codes (NCC) are similar to convolutional codes except that they can be designed for general nonbinary sources. The residual redundancy in the source code output must be preserved for forward error correction. This requires that channel coder input alphabet must have a one-to-one match with the source output.

Let x_n, which is chosen from the alphabet A={0, 1, 2, 3, …. , G-1} be the input, and y_n, chosen from B={0, 1, 2, 3, …., H-1}, be the output of NCC.

H=G^2, output is:

$$y_n = Gx_{n-1}+x_n \tag{3.1}$$

for rate R=1/2 . This can be shown by the following argument:

For the output alphabet $\log_2 H$ and for the input alphabet $\log_2 G$ bits are required.

$$\log_2 H = \log_2 G^2 = 2\log_2 G \tag{3.2}$$

and the rate,

$$R=\frac{\log_2 G}{\log_2 H} = \frac{1}{2} \tag{3.3}$$

In figure 3.1, a block diagram of a NCC with rate R=1/3 is shown.

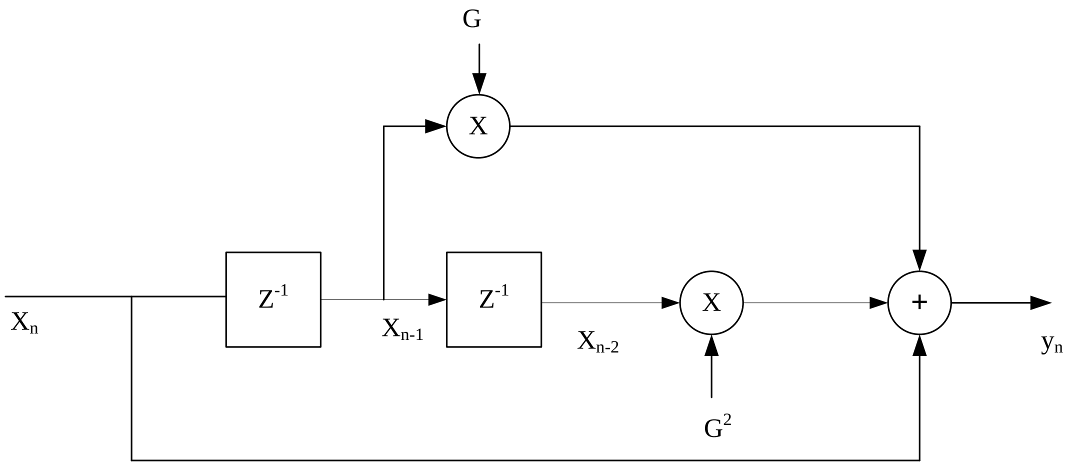

Figure 3.1: Rate *R*=1/3 NCC encoder structure. [6]

The NCC decoder uses the Viterbi algorithm. [6]

3.1.1 The Design Criterion

For a discrete memoryless channel (DMC), let $y = (y_1, y_2, \ldots, y_N)$ and $r = (r_1, r_2, \ldots, r_N)$ denote the transmitted and the received sequences, respectively, where symbols y and r are from the same alphabet. The probability of error is given by

$$P(E) = \sum_r P(E/r)P(r) \tag{3.4}$$

where $P(r)$ is independent of the decoding rule. To minimize the error, the optimum receiver maximizes

$$P(y/r) = \frac{P(r/y)P(y)}{P(r)} \tag{3.5}$$

For fixed length codes, $P(r)$ is irrelevant to receiver's operation and $\log P(y_i / y_{i-1})$ $P(r/y)P(y)$ must be maximized. Since the channel is without memory, each received symbol depends only on the corresponding transmitted symbol.

$$P(r / y) = \prod_i P(r_i / y_i) \tag{3.6}$$

When the information source is assumed to be an M^{th}-order Markov sequence, one can write

$$P(y) = \prod_i P(y_i / y_{i-1}, y_{i-2}, \ldots\ldots, y_{i-M}) \tag{3.7}$$

Combining Equation (3.6) and (3.7) we get

$$P(r / y) = \prod_i P(r_i / y_i) P(y_i / y_{i-1}, y_{i-2}, \ldots\ldots, y_{i-M}) \tag{3.8}$$

Taking the algorithm of both sides gives

$$\log P(r / y) = \sum \log P(r_i / y_i) P(y_i / y_{i-1}, y_{i-2}, \ldots, y_{i-M}) \tag{3.9}$$

The sum is similar to the path metric used in the decoding of convolutional codes. Error correction using convolutional codes is made possible by restricting the possible codeword to code work transition based on the coder structure. The receiver compares the received data stream to the a priori information about the code structure. In the case where there is residual structure in the source coder output, the redundancy can be used for error correction. This residual structure is reflected in the form of conditional properties and can be used in the metric of a convolutional decoder. If we assume a first- order Markov model, the metric becomes

$$\log P(r_i / y_i) = \sum_i P(r_i / y_i) P(y_i / y_{i-1}) \tag{3.10}$$

Examining the path metric, we see that it contains $\log P(r_i / y_i)$ which depends strictly on the channel and $\log P(r_i / y_{i-1})$ which depends on the source statistics. This metric is called the maximum a posteriori (MAP) metric. It is necessary to know the channel and source statistics to be able to implement the path metric. As noted previously, the source is a first-order Markov chain. The source symbols are produced by the state-to-state transitions with given conditional probabilities.

Convolution coding, that took advantage of the residual redundancy was used. In evaluating the branch metric

$$L = \log P(r_i / y_i) + \log P(y_i / y_{i-1}) \tag{3.11}$$

It is assumed that the channel statistics are known. It is shown in the simulation results that the system is quite robust to mismatch between the assumed and the actual statistics. In the simulations, the effect of mismatch was investigated by varying $P(r_i / y_i)$ in the computation of the MAP metric for a given channel error rate. For the matched case, it is assumed that $P(r_i / y_i)$ is exactly known and this information is used in evaluating the MAP metric. The performance of the system is not affected adversely by the mismatch for a range of channel error rates but it is highly dependent on the amount of residual redundancy at the source. The simulation results indeed show that the performance increases for decreasing source entropy. In order to compute the entropy, the probability of source symbols is estimated. The conditional entropy $H(y_i / y_{i-1})$ is calculated using the conditional probabilities which is estimated by using the source symbols. [6]

3.2 Markov Chain

Having the Markov property means that future states depend only on the present states, and are independent of past states. At each step the system may change its state from the current state to another state (or remain in the same state) according to a probability distribution. The changes of state are called transitions, and the probabilities associated with various state-changes are called transition probabilities. [12]

A Markov chain is a sequence of random variables X_1, X_2 ,X_3,…. with the Markov property, namely that, given the present state, the future and the past states is independent. Formally,

$$Pr\ (X_{n+1} = x \mid X_1 = x_1, X_2 = x_2 \ldots, X_n = x_n) = Pr(X_{n+1} = x \mid X_n = x_n) \tag{3.12}$$

Decoding process in Viterbi algorithm depends on channel metric calculation. Here, source information is used to increase the performance of decoder by combining source information with channel metric;

$$L = \log P(r_i / y_i) + \log P(y_i / y_{i-1}) \tag{3.13}$$

where $\log P(r_i / y_i)$ is channel and $\log P(y_i / y_{i-1})$ is source metric.

Source information is in fact metrics that depend on the transition probabilities of the source data, and if transition probabilities between certain symbols are reasonably high, the Viterbi decoder with source metric performs even better. So, Markov chain is used to produce data with manually set transition probabilities and these transition probabilities are then used in decoder for calculation of branch metrics. The data produced with equal transition probabilities will leave no advantage for decoder with source metric because equal transition probabilities mean uniformly distributed data so there is no useful information about source data for the decoder. But if the transition probabilities between certain symbols are considerably high, decoder with source metric will make good use of this information and will show considerably better performance than the other methods. Images are like Markov data because generally in an image, transition probabilities between a symbol and itself are higher than the other transition probabilities.

3.3 Entropy

The information in an image can be modeled as a probabilistic process, where first, a statistical model of the image is generated. The information content (entropy) can be estimated based on this model.

This information per source (pixels or symbol), which is also referred as entropy is calculated by:

$$S = -\sum P_i \ln P_i \tag{3.14}$$

where S is the entropy, P_i refers to the source symbol/pixel probabilities.

If data, generated by using Markov process, has high transition probabilities between specific symbols, the similarity between the source symbols becomes high. As the similarity between symbols of a source increases, entropy decreases. For a source which has low entropy, the effect of soft decision nonbinary convolutional coding with source metric will be better since entropy is completely related to data statistics. Suppose the transition probabilities between each symbol are equal. This means that the symbols are uniformly distributed and there is no similarity between them. In this case entropy will be high and source statistics will have no effect during decoding process. This can be observed from simulation results as well.

3.4 Wireless Communication Channel

Mobile communication has become such an important part of our lives that we cannot imagine a life without it. In view of today's trend in the field of communication, digital communication techniques dominate analogue methods. As shown in figure 3.2, mobile communication systems consist of three main elements which are the transmitter, communication channel, and the receiver. [25]

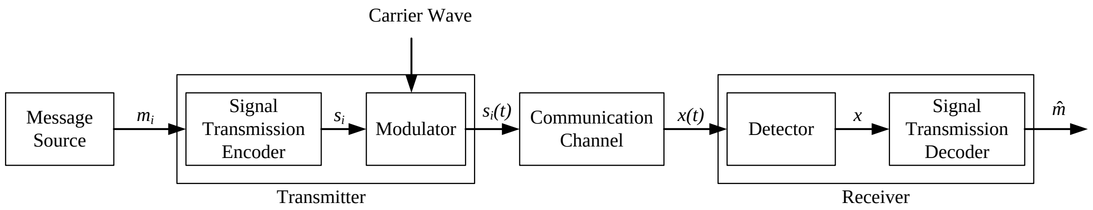

Figure 3.2: Block Diagram of a General Communication System

Communication signals are transmitted through very different kinds of channels. For the purpose of designing and optimizing receiver structures for digital communication systems it is mandatory to construct mathematical models that represent the typical characteristics of these channels. [3] In this thesis, AWGN channel and fading channel models will be used.

3.4.1 Additive White Gaussian Noise Channel

One of the simplest channel models for a communication system is the Additive White Gaussian Noise (AWGN) model. The basic model of AWGN is used to in digital communications as shown below:

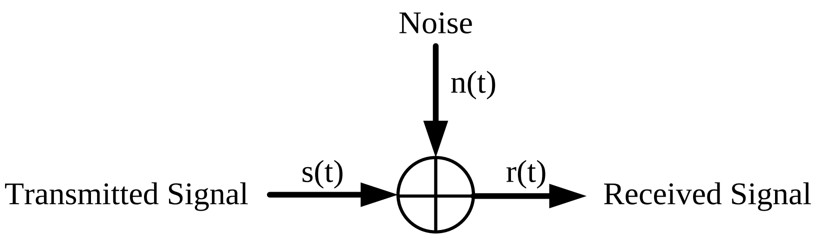

Figure 3.3: AWGN Channel

$$r(t) = As(t) + n(t) \tag{3.15}$$

As it seen from the figure, received signal $r(t)$ is the sum of the transmitted signal $s(t)$ and white Gaussian Noise $n(t)$, whose frequency spectrum is continuous and uniform over a specified frequency band. A is the overall path loss, assumed to be time invariant.

In AWGN system, there is no fading, frequency selective, interface, nonlinearity or dispersion; hence, this model is overly simplistic for wireless communications systems.

An important parameter of measuring the performance of digital modulation systems is the signal-to-noise ratio (SNR). This parameter determines the probability of information error, or bit error rate (BER). The input SNR into the demodulator of AWGN channel is defined as inverse proportion of noise PSD N_0 and bandwidth B, defined below.

$$\mathrm{SNR}=\frac{\left\langle \mathrm{A}^2, S^2(t)\right\rangle}{2\sigma_n^2}=\frac{A^2}{2N_0B} \tag{3.16}$$

3.4.2 The Flat Fading Channel

This simplest fading channel model assumes that the duration of a signal is much greater than the delay spread caused by multipath propagation. If this is true, then all frequency components in the transmitted signal are affected by the same random attenuation and phase shift, and the channel is frequency-flat. If in addition the channel varies very slowly with respect to the elementary-signal duration, then the fading level remains approximately constant during the transmission of one signal (if this does not occur, the fading process is called fast). The additional assumption of slow fading reduces this process to a sequence of random variables, each modeling an attenuation that remains constant during each elementary-signal interval. In conclusion, if x denotes the transmitted elementary signal, then the signal received at the output of a channel affected by slow, flat fading, and additive white Gaussian noise, and demodulated coherently, can be expressed in the form

$$Y=Rx+z \quad (3.17)$$

where z is a complex Gaussian noise and R is a Gaussian random variable, having a Rayleigh PDF. It should be immediately apparent that, with this simple model of fading channel, the only difference with respect to an AWGN channel, described by the input/output relationship

$$y=x+z \quad (3.18)$$

Resides in the fact that R, instead of being a constant attenuation, is now a random variable whose value affects the amplitude, and hence the power, of the received signal. [4]

CHAPTER 4

SIMULATION RESULTS

4.1 Simulation Setup

In this thesis, Monte Carlo Simulation is used in the experiments and is applied on synthetic data, images and video sequences. Markov Chain model is used to generate synthetic data to enable manually adjustable transition probabilities between each data. 10000 binary and nonbinary (4 levels) data are generated using Markov Chain model and used as sources for Convolutional Coding.

For the synthetic data experiments, the data sets are coded, sent through the channel, and decoded 100 times to perform Monte Carlo Simulation. Transition probabilities between data are calculated to be used for source metric in decoding algorithm. The effect of soft metric in the algorithm is observed comparing with the conventional decoding algorithms (hard decision and soft decision viterbi decoding). Different transition probability configurations are used in the simulations to observe the effect of the source itself on Viterbi decoding algorithm. BPSK modulation is used to send data through AWGN and flat fading channels, and then received data is decoded using modified Viterbi decoding algorithm with source metric to observe the performance of the proposed method. The proposed method adds source metric to channel metric while decoding received data to improve performance of Viterbi decoding algorithm.

In the experiments in which images are used, RGB image is first converted to grayscale image and then quantized to obtain a 4 level gray scale image. Then, the intensity values are mapped to values 0, 1, 2, and 3 in the ascending order forming nonbinary data for an image. The source statistics are calculated for the nonbinary data in the same manner used for synthetic data. The nonbinary data is then modulated using BPSK and transmitted through AWGN and flat fading channel. The received data is decoded using Viterbi decoding algorithm. This process is carried out only once for an image. Three types of experiments are set up for image transmission. First, transmitted image is decoded using its own source statistics. For the second type of experiment, the transmitted image is decoded using a different image's source metric which is similar to the transmitted image in terms of source statistics. And for the third experiment, the transmitted image is decoded using a completely different image's in terms of source statistics.

Video sequence transmission is also set up as one of experiments. 10 consecutive frames of a video sequence are converted into 4 level nonbinary sources and then transmitted through AWGN and flat fading channel. The received video sequence is then decoded using Viterbi decoding algorithm with source metric. The video sequence experiment is performed to observe the performance of soft decision nonbinary convolutional coding with source metric on correlated data sources.

Another experiment performed in this thesis is using an increased minimum code distance. Minimum code distance is the minimum hamming distance between the labels of state transition probabilities of the decoder. The effect of increased minimum code distance is observed.

4.2 Performance study for simulated data

The figures 4.1, 4.2, and 4.3 describe the performance of convolutional coding. Data, from synthetic sources, transmitted through the AWGN channel.

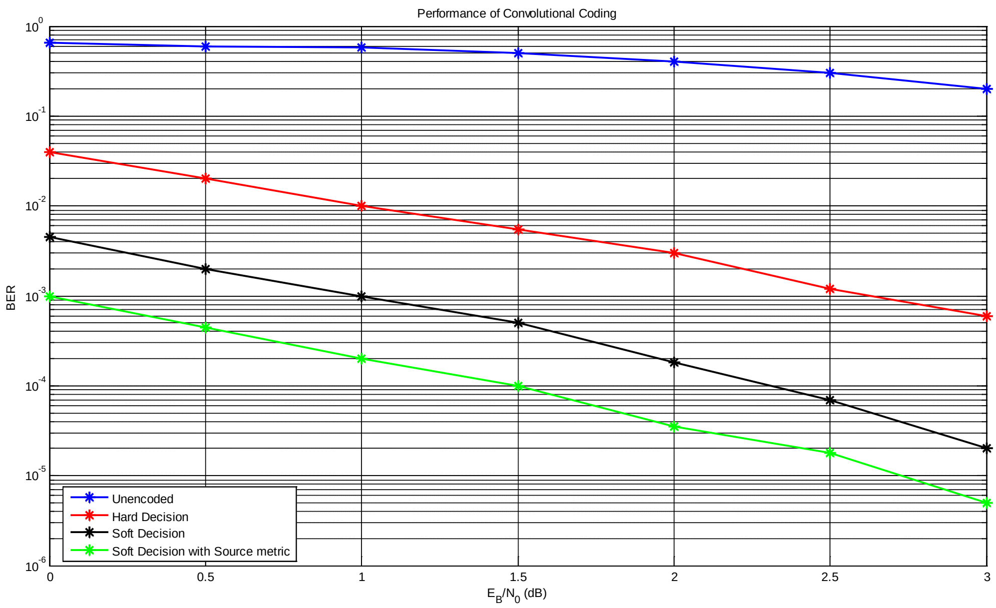

Figure 4.1: Binary data performance of convolutional coding (K=3, AWGN, m=2, g=[7,5]) transition probabilities:[0.3 0.7;0.7 0.3])

BER performance of convolutional coding of binary random data is shown in the above figure. At a BER=10^{-3}, the convolutional coding with soft decision requires E_b/N_0 of 0.9dB whereas convolutional coding with soft decision with source metric requires about 0dB and the convolutional coding with hard decision requires 2.7dB respectively. Here, we see the advantage of using the source statistics in decoding the transmitted bits. The next figure show the performance of nonbinary convolutional coding.

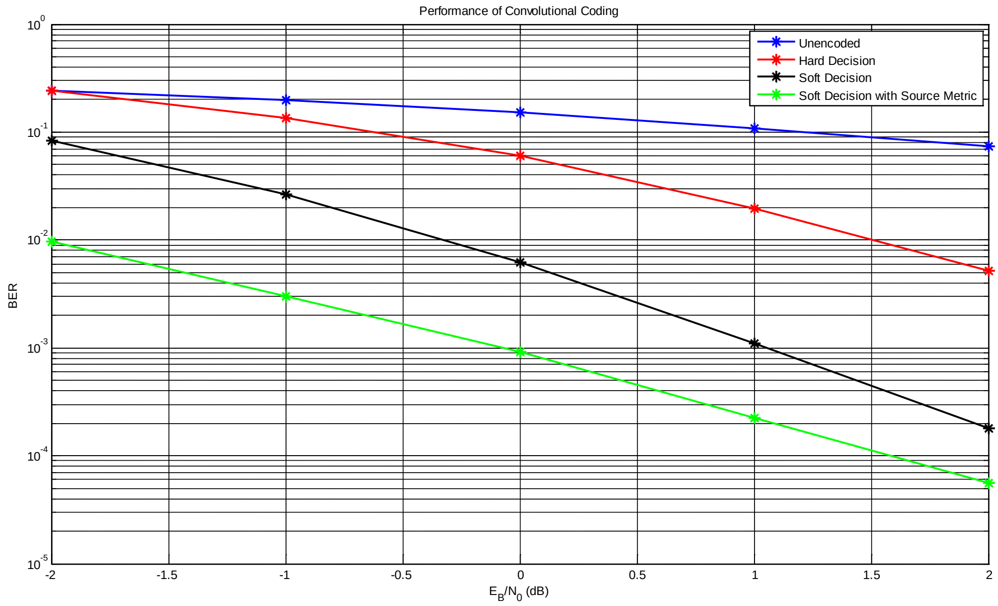

Figure 4.2: Nonbinary convolutional coding performance for data (d_{free} =1, AWGN, *R*=2/4, *K*=5, *m*=4)

The source produces nonbinary random data. The soft decision with source metric performance is higher than that of the other convolutional coding since it depends on source statistics. At a BER=10^{-2}, the convolutional coding with soft decision requires E_b/N_0 of -0.4dB whereas convolutional coding with soft decision with source metric requires -2dB. Finally, the convolutional coding with hard decision requires 1.8dB. It is observed that nonbinary convolutional coding is superior to binary convolutional coding for a nonbinary source with residual redundancy. Next figure show performance comparison of convolutional coding with different code distance and rate.

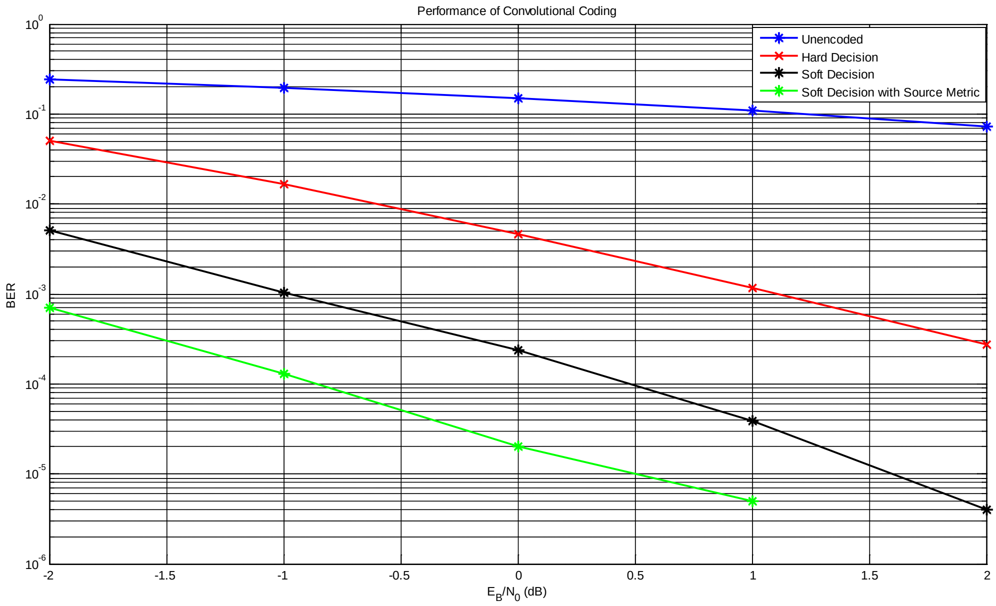

Figure 4.3: Nonbinary convolutional coding performance for data (d_{free} =2, AWGN, R=2/6, K=5, m=4)

In the figure above, rate 2/6 is used for encoding the nonbinary message. The previous figures show the results for code with rate 2/4 which adds 2 controlled redundancy bits to each message bit. In the current experiment, 4 controlled redundancy bits are added to for every 2 of the message bits and effect is observed on the figure above. It can be seen that increasing the coding rate greatly increased the performance of all the coding methods with different metrics. In figure 4.4, when E_b/N_0=0, BER is 7x10^{-4} for soft decision with source metric for d_{free} =2. However in figure 4.3, when E_b/N_0=0, BER is 2x10^{-4} for d_{free} =2 which shows the increase in performance.

4.3 Simulation Results for Different Code Distances

The second set of figures 4.4 to 4.6 describe the performance of soft decision convolutional coding with source metric using different code distances (d_{free}) in AWGN and fading channels. These figures describe the performance of the proposed code with higher d_{free} which are designed by using the procedure described in section 2.2.3.5.

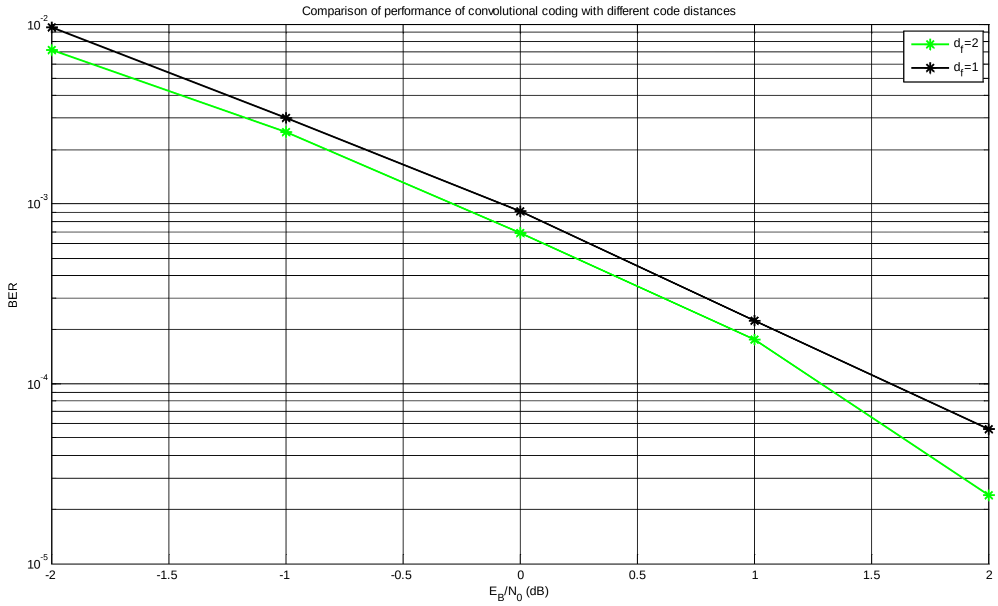

Figure 4.4: Performance of nonbinary convolutional coding with different code distances.

(d_{free} =1 and d_{free} =2, AWGN, R=2/4, K=5, m=4)

Figure 4.4 shows soft decision NCC with source metric performances having different code distances. The messages consist of 10000 bits and the experiment is repeated 100 times for each E_b/N_0 value. The minimum free distance (d_{free}) is increased from 1 to 2 and it can be seen from figure 4.4, increasing d_{free} to 2 slightly increased the performance of the coding method.

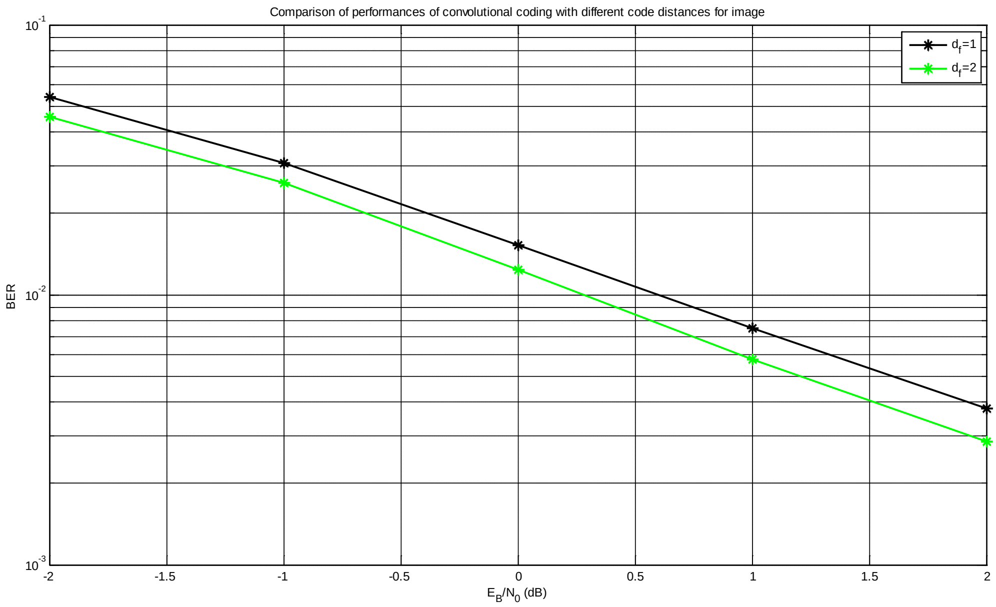

Figure 4.5: Performance of nonbinary convolutional coding with different code distances.

(d_{free} =1 and d_{free} =2, flat fading, R=2/4, K=5, m=4)

Above figure illustrates NCC with soft decision with source metric performances with different code distances in fading channel. d_{free} is again increased from 1 to 2 and results are compared. Comparing figure 4.4 and 4.5, fading channel greatly decreased performances of both methods but still increasing d_{free} to 2 results slightly better performance in flat fading channel compared to when d_{free} is 1.

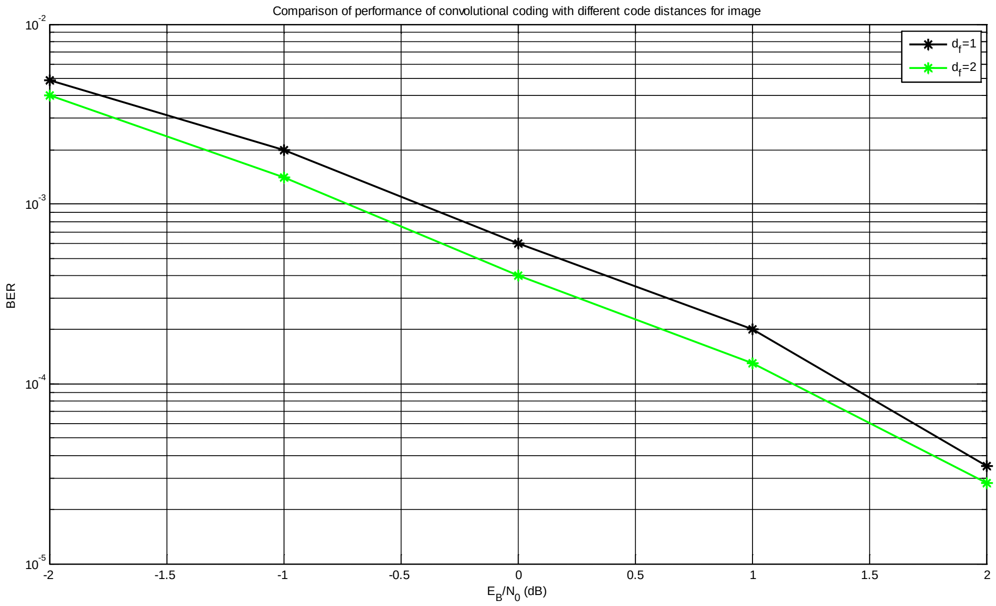

Figure 4.6: Performance of nonbinary convolutional coding with different code distances.

(d_{free} =1 and d_{free} =2, AWGN, 392 x 294 pixels image, *R*=2/4, *K*=5, *m*=4)

In Figure 4.6, the effect of the code distance to NCC is demonstrated for an image. When code distance (d_{free}) value is increased, the method again shows better performance.

4.4 Performance study using images

The third set of figures 4.8-4.13 describes the performance of image transmission. The performances of coding methods are observed on image transmission rather than using random data sets to emphasize power of the proposed methods in multimedia transmission. The following images are converted to 4 level grayscale images and used in the simulations:

(a) (b) (c)

Figure 4.7: (a) Transmitted Image, (b) Reference image with source statistics similar to transmitted image, and (c) Reference image with source statistics different from transmitted image

Depending on the application area, calculation and transmission of source metric would be redundant. To avoid redundant calculations and transmissions, pre-calculated source statistics can be used to decode transmitted images. Three types of scenarios are set up and performances are compared to show instead of using message source metrics, pre-calculated image sources can also be used to improve the performance of convolutional coding.

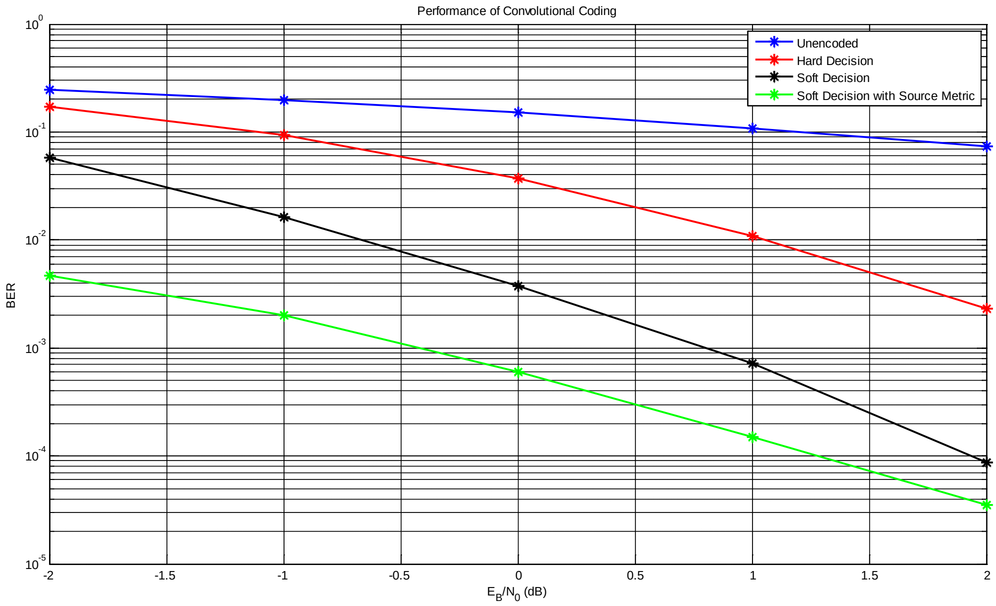

Figure 4.8: Nonbinary convolutional coding performance for image (K=5, AWGN, d_{free} =1, gray scale 392 x 294 pixels, m=4, K=5)

The figure illustrates BER performance of NCC for image which has 392 x 294 pixels. At a BER=4x10^{-3}, the NCC with soft decision requires E_b/N_0 of -0.1dB whereas convolutional coding with soft decision with source metric requires -1.8dB. The convolutional coding with hard decision requires 1.6dB. In the previous experiments, data is generated with first order Markov process to have transition probabilities similar to an image's transition probabilities. In this experiment, image is transmitted to see the performance while transmitting a visual data set.

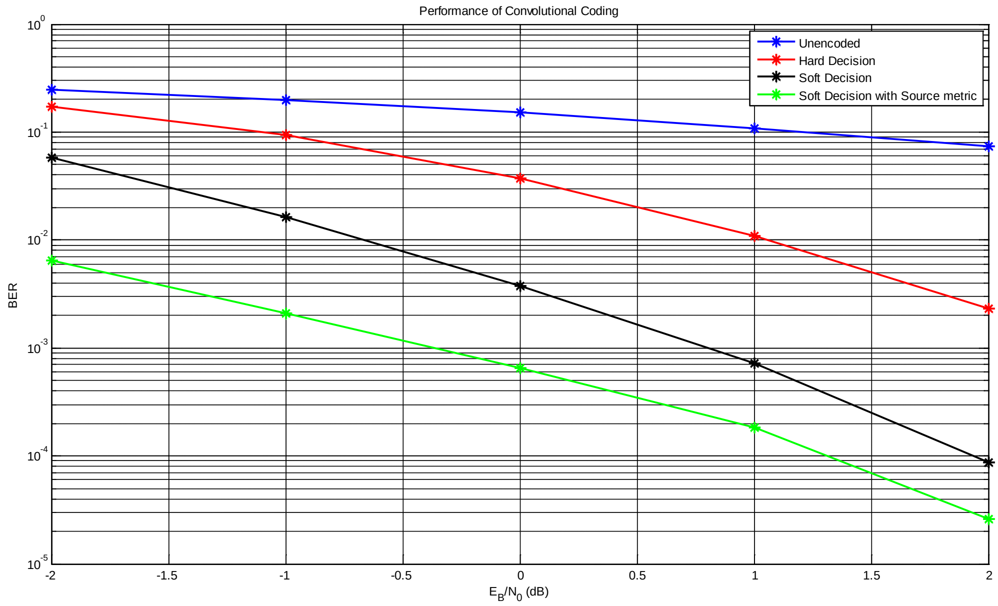

Figure 4.9: Nonbinary convolutional coding performance for image with using different image probabilities (d_{free} =1, AWGN, gray scale 392 x 294 pixels and gray scale 392 x 294 pixels, R=2/4, K=5, m=4)

In this figure, BER performance of NCC for a 392 x 294 image is illustrated. The source decision with source metric performance is over performing the other techniques, whereas the soft decision based convolutional coding is comparable with source metric technique in low SNR value. At BER=5x10^{-2} soft decision convolutional coding with source metric needs -1.8dB, soft decision convolutional coding needs -0.4dB and hard decision convolutional coding requires 1.5dB of E_b/N_0. This figure proves that if source statistics of reference image are close to source statistics of original image, soft decision convolutional coding with source metric performs better results.

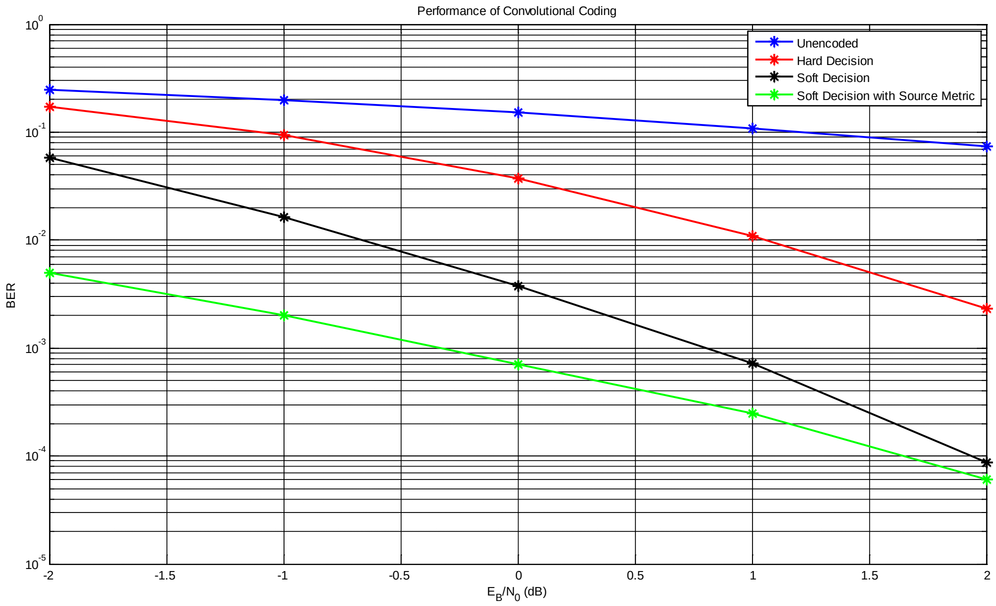

Figure 4.10: Nonbinary convolutional coding performance for image with using different image probabilities (d_{free} =1, AWGN, gray scale 372 x 270 pixels and gray scale 392 x 294 pixels, R=2/4, K=5, m=4)

In the figure above, instead of using original image source statistics for the soft decision with source metric, different image source statistics are used which are generated from a completely different image. Because the reference image is very different from original image, the statistics fail to help decoder decreasing the performance of soft decision convolutional coding with source metric. For a good performance, source statistics of reference image should be close to the original image's source statistics. Next figure illustrates performance comparison between nonbinary and binary convolutional coding methods.

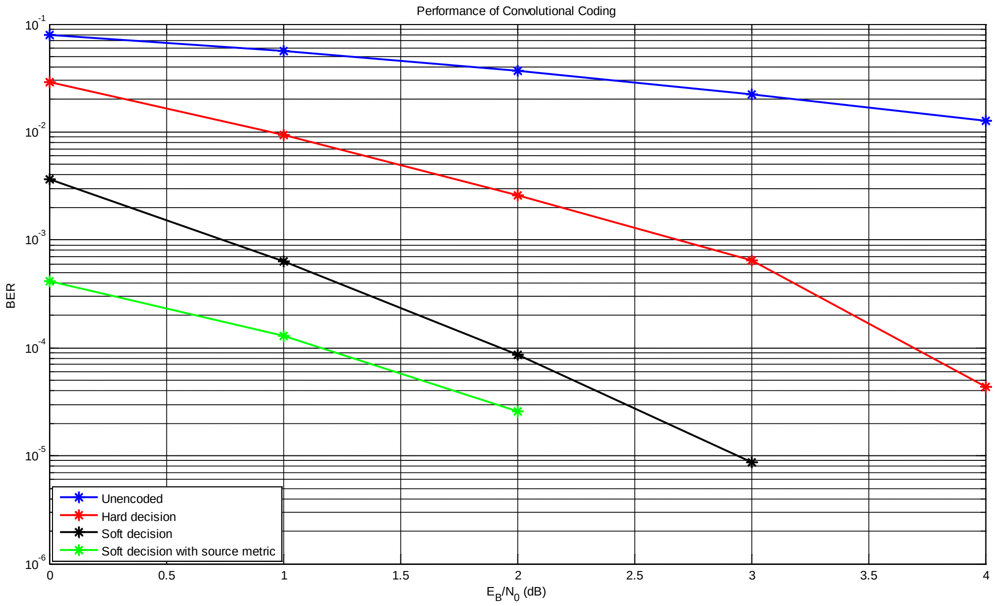

Figure 4.11: Binary image performance of convolutional coding (*K*=3, AWGN, *m*=2, *g*=[7,5])

Figure 4.11 illustrates performances of binary convolutional coding methods. For a performance of BER=10^{-4}, soft decision convolutional coding with source requires E_b/N_0 of 1.2dB, soft decision convolutional coding requires 1.8dB, and hard decision convolutional coding requires 3.7dB. Binary convolutional coding performs good performance, but to use binary convolutional coding data should be binarized first. Some data sets may not be binary depending of the application area, so, if a nonbinary data set is to be transmitted, nonbinary convolutional coding can be used to directly encode nonbinary data sets without converting them into binary. As can be seen from the results, nonbinary convolutional coding has performance as good as binary convolutional coding.

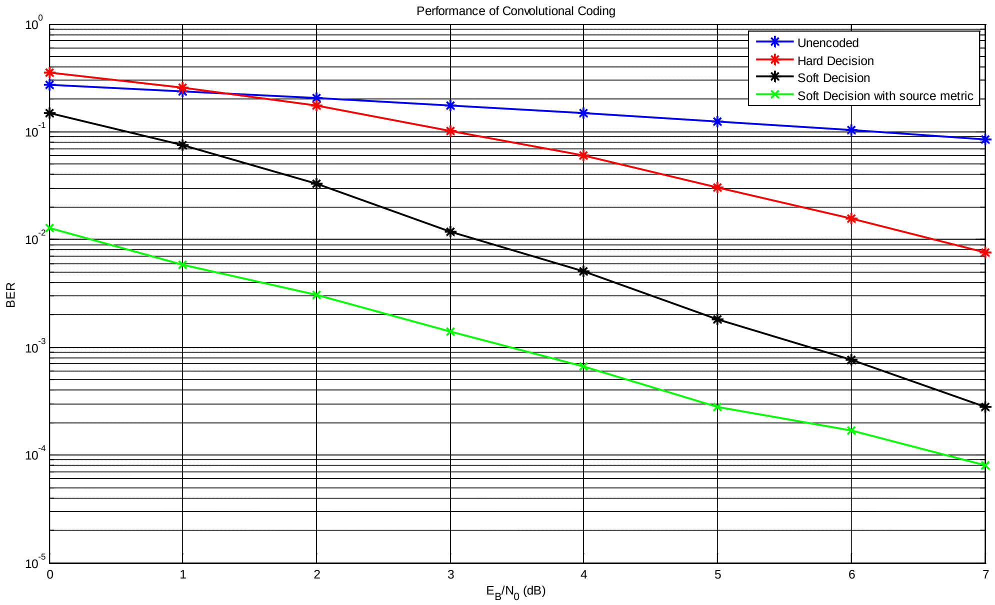

Figure 4.12: Nonbinary convolutional coding performance for image (flat fading, d_{free} =2, gray scale 372 x 270 pixels, R=2/4, K=5, m=4)

In this figure, NCC performance for image is observed in fading channel. At BER of 10^{-3}, soft decision convolutional coding with source metric has 2.7dB greater performance than soft decision convolutional coding. While soft decision convolutional coding with source metric requires 0.3dB at BER=10^{-3}, soft decision convolutional coding requires 3.1dB and hard decision convolutional coding requires 6.6dB.

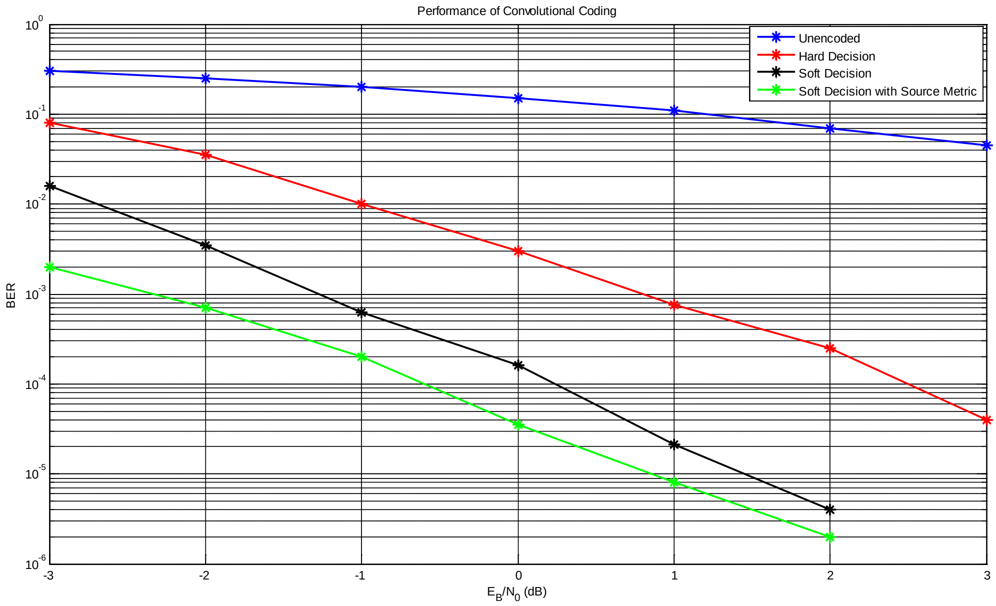

Figure 4.13: Nonbinary convolutional coding performance for image (AWGN, d_{free} =2, gray scale 816 x 612 pixels, *R*=2/6, *K*=5, *m*=4)

Figure 4.13 illustrates performance of convolutional coding methods with rate *R*=2/6 for an image. Comparing performance of methods used, at BER=10^{-3}, soft decision nonbinary convolutional coding with source metric requires E_b/N_0 of -2.3dB, soft decision nonbinary convolutional code requires E_b/N_0 of -1.25dB and hard decision nonbinary convolutional coding requires E_b/N_0 of 0.85dB.

4.5 Simulation results for video sequence

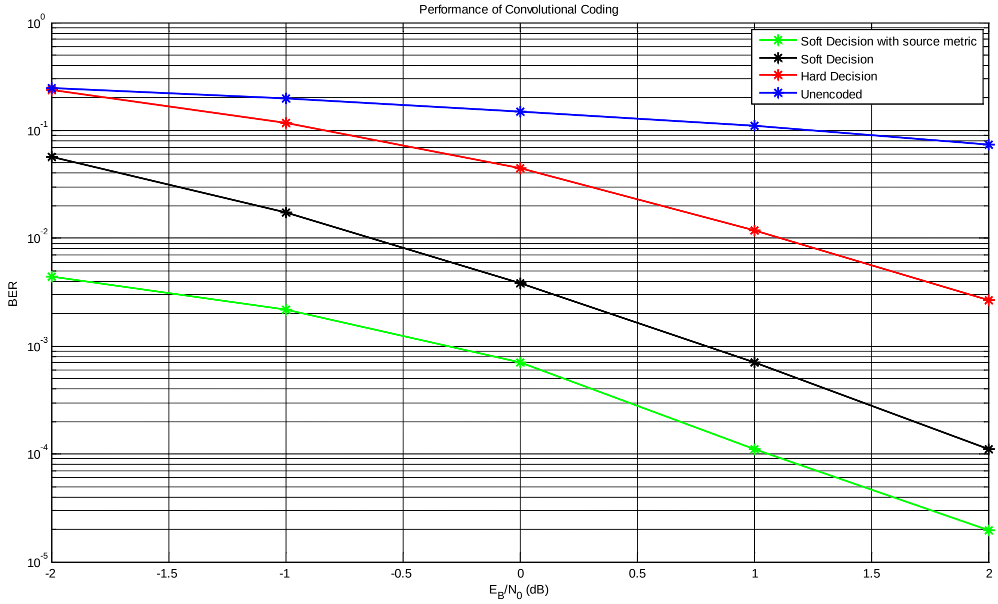

Figure 4.14: Nonbinary convolutional coding performance for video sequence (AWGN, d_{free} =2, 10 consecutive frames, gray scale 120 x 177 pixels/frame, R=2/4, K=5, m=4)

In this figure, 10 consecutive frames of a video sequence are encoded and transmitted over AWGN channel. There is more redundancy in video sequences than a single static image. When the redundancy is increased the NCC method performs better. Comparing the results, at BER=4×10^{-3}, soft decision nonbinary convolutional coding with source metric requires E_b/N_0 of -2dB, soft decision nonbinary convolutional code requires E_b/N_0 of 0dB and hard decision nonbinary convolutional coding requires E_b/N_0 of 1.6dB.

The benefit of coding for a video sequence can be explained due to the fact that there is not much difference between two consecutive frames. And hence the correlation between the consecutive frames can be explained by the NCC algorithm in its updated source metric. The residual redundancy enables the decoder to make better decision in state transition.

4.6 Entropy Calculation Results

The entropy values show that the soft decision with source metric of a given SNR value is higher than the other techniques which represents that the randomness of this technique is higher. As, entropy of the source increases the performance decreases. When entropy increases, the structure in the source which can be utilized by NCC decreases and the decoder makes more errors. When the entropy is low, there is a lot of structure in the source and the decoder makes efficient use of this in making correct decisions.

Markov Random Data	SNR				TP	Entropy
	0	1	2	3		
BER	0	0	0	0	1	0
	0.0045	0.0008	0	0	2	1.9998
	0.0036	0.0006	0	0	3	1.9706
	0.00076	0.00014	0	0	4	1.9616

Transition Probability (TP1)

	0	1	2	3
0	0	1	0	0
1	0	1	0	0
2	0	1	0	0
3	0	1	0	0

Transition Probability (TP2)

	0	1	2	3
0	0.25	0.25	0.25	0.25
1	0.25	0.25	0.25	0.25
2	0.25	0.25	0.25	0.25
3	0.25	0.25	0.25	0.25

Transition Probability (TP3)

	0	1	2	3
0	0.4	0.3	0.2	0.1
1	0.3	0.5	0.2	0.1
2	0.1	0.2	0.6	0.1
3	0.05	0.05	0.2	0.7

Transition Probability (TP4)

	0	1	2	3
0	0.8666	0.1253	0.0071	0.0015
1	0.1608	0.7263	0.0961	0.0121
2	0.003	0.665	0.8636	0.0691
3	0.0051	0.0043	0.1277	0.8678

Table 4.1: Entropy and BER performances of soft decision with source metric

(R=2/4, m=2, K=3, TP 1-4)

Table above shows various symbol transition probability tables for synthetic data and BER results for the soft decision nonbinary convolutional coding with source metric method for these synthetic data at SNR values 0, 1, 2, 3, and 4. Entropy of each data source is calculated to show the relationship between BER performance and data source. Each data source is generated with different transition probability tables to obtain different entropy values. Looking at table above, for data source with transition probabilities TP1, the data is highly correlated resulting a very small entropy value with a very high decoding performance. Data source with transition probabilities TP2 is a uniformly distributed random data source resulting a very low correlation between data resulting a high entropy value. Decoding performance for this data source is low.

CONCLUSIONS AND FUTURE WORK

In this thesis, the binary and nonbinary convolutional coding techniques are examined in communication channels such as AWGN and the flat fading channel. Convolutional decoding with hard, soft and soft with source metric decisions are implemented and the simulation results are discussed. The results show that both binary and nonbinary convolutional codes with soft decisions with source metric perform is better than the other technique. Soft decision with source metric uses source transition probabilities for metric calculation. Binary and nonbinary convolutional coding techniques are applied to images and video sequence transmission and the performances are evaluated. Image transmission is tested after the image source statistics are obtained. When the decoder utilizes these statistics, better decisions lend to lower BER. It is interesting to note that even the source statistics of different images can increase the efficiency of the decoder.

There are several different performance measures that can be used to compare convolutional codes. In this thesis, the commonly used measure, the minimum free distance is used. The effect of increasing the code distance is examined. When the code distance is increased, Hamming distance among the branch labels are increased, and hence the convolutional coding performance is also increased.

Use of residual redundancy with NCC improved the bit error performance. Use of nonbinary convolutional coding on a video sequence yielded good results because of high residual redundancy of video sequences.

A new nonbinary convolutional code, with rate 2/6 (1/3 for binary), is proposed and compared to nonbinary convolutional codes with rate 2/4 (1/2 for binary). The results show that the proposed code is superior to the ones with rate 2/4.

In future work, different trellis depths, quantizers and more precise channel models such as the multipath fading channel can be used.

Field Programmable Gate Array (FPGA) boards and Direct Sequence Spread Spectrum (DSSS) systems are examples of areas that nonbinary convolutional coding technique can be used.

REFERENCES

[1] W. B. Stephen, *Error Control Systems for communication and Storage,* Upper Saddle River: Prentice Hall, 1995

[2] S. Lin and D. J. Costello, JR., *Error Control Coding: Fundamentals and Applications*, Prentice Hall, Englewood Cliffs, N. J., 1983

[3] N. Tekbıyık, "Closed Loop Power Control with Fixed Step Size in DS-CDMA Cellular Systems", M.S. Thesis, Department of Electrical and Electronic Engineering, Eastern Mediterranean University, 2005.

[4] E. Biglieri, *Coding For Wireless Channels*, Springer Science and Business Media, 2005.

[5] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM '89)*, vol. 3, pp. 1680-1686, Dallas, Tex, USA, November 1989.

[6] A. Hocanın and K. Sayood, "Error Correction Capability of Nonbinary Convolutional Codes", *Proceedings of SIU 97 (5th National Signal Processing and Applications Conference)*, Kusadasi, Izmir, 1-3 May 1997, pp. 377-381.

[7] N. Demir and K. Sayood, "Joint source/channel coding for variable length codes," *in Proc. Data Compression Conf*., 1998, pp. 130-148.

[8] J. G. Dunham and R. M. Gray, "Joint source and channel trellis encoding," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 516-519, July 1981.

[9] K. Sayood and J. C. Borkenhagen, “Use of residual redundancy in the design of joint source/channel coders,” *IEEE Trans. Commun.*, vol. VOL. 39, pp. 838-846, June 1991.

[10] K. Sayood, Fulig Liu, and J. D. Gibson, “A constrained joint source/channel coder design,” *IEEE* Joum. Selec. *Areas* in Comm. VOL. 12, pp. 1584- 1593, Dec. 1994.

[11] J. G. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2001.

[12] Members of Wikipedia. The Free Encyclopedia, May 2009. http://en.wikipedia.org/wiki/Markov_chain

[13] P. Sweeney, *Error Control Coding*: From Theory to Practice, 2002.

[14] Tariq Haddad and Abbas Yongaçoglu, "Joint Source/Channel Soft Viterbi Decoding," iscc, pp.467, *The Fourth IEEE Symposium on Computers and Communications*, 1999.

[15] T. K. Moon, Error Correction Coding, Mathematical Methods and Algorithm, Wiley-Interscience, 2005.

[16] Viterbi and J. Omura, *Principles of Digital Communication and Coding*. McGraw-Hill Kogakusha LTD., Tokyo Japan, 1979.

[17] P. Sweeney, *Error control coding: From Theory to Practice*. John Wiley & Sons. Ltd, West Sussex, England, 2002.

[18] G. D. Forney “The Viterbi Algorithm”, *Proceedings of the IEEE*, Vol 61, No.3, pp 268-278, March 1973.

[19] John G. Proakis and Masoud Salehi, *Communication Systems Engineering*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1994.

[20] T. C. Ancheta Jr., "Joint source channel coding", Ph.D. dissertation, Univ. of Notre Dame, IN, Aug. 1977.

[21] K. Sayood J.C.Borkengahen, "Use of residual redundancy in the design of joint source and channel coders", *IEEE Trans. Comm.*, vol 39, pp 838-846, Mar. 1991.

[22] M. Savaş, "Automatic Repeat Request", BSc., Final Year Project, Department of Electrical and Electronic Engineering, Eastern Mediterranean University, 2004.

[23] U. Hocanın, "Closed-Loop Power Control and Code Synchronization in DS-CDMA", M.S. Thesis, Department of Electrical and Electronic Engineering, Eastern Mediterranean University, 2005.

[24] V. Vaishampayan and N. Farvardin, "Joint design of block source codes and modulation signal sets", *IEEE Trans. Inform. Theory*, Vol. 38, pp.1230-1248, July 1992.

[25] C.E. Shanon, "A mathematical theory of communication", *Bell Syst. Tech. J.*, Vol 27, pp.379-423, July 1948.

[26] S. Venbu, S. Verdu, and Y. Steinberg, "The source-channel separation theorem revisited", *IEEE Trans. Inform, Theory,* Vol. IT-41, No.1, pp.44-54, Jan. 1995.

[27] C. Perrine, C.Chatellier, S.Wang and C. Olivier, "A joint source channel coding strategy for video transmission", *IEEE Information and Communication Technologies: From Theory to Applications, 2008,* April 2008.

[28] K. Sayood, H. Out, and N. Demir, "Joint source/channel coding for variable length codes", *IEEE Trans. Comm.*, Vol. 48, no.5, pp. 787-794, May 2000.

[29] M. Jeanne, J. Carlach, P. Siohan, and L. Guivarch, “Source and joint source-channel decoding of variable length codes,” in *Proc. IEEE Int. Conf. Communications*, 2002, pp. 768-772.

[30] C. Demiroglu, M. Hoffman, and K. Sayood, “Joint source channel coding using arithmetic codes,” *IEEE Trans. Commun.*, vol. 49, no. 9, pp.1540-1548, Sep. 2001.

[31] Ajay Dholakia, *Introduction to convolutional codes with applications*, Kluwer Academic Publishers, 1994.

[32] G. D. Forney, *The Viterbi Algorithm*, Proceeding of the IEEE, Vol. 61, no. 3, pp. 268278, Mar 1973.

[33] W. Xu, J. Hagenauer and J. Hollman, “Joint source channel decoding using the residual redundancy in compressed images”, in *Proc. ICC/SUPERCOMM’96*, June 1996, pp. 142-148.

[34] N. Demir, K. Sayood, “Joint Source/Channel Coding for Variable Length Codes,” dcc, pp.0139, *Data Compression Conference*, 1998.