Performance Analysis of Convolutional and Gray Coding Techniques in Wireless Communications

Simon W. Pallam, George A. Audu, Saidu Y. Musa, Ibrahim M. Visa

Abstract—Multipath fading effects are posing immense problems to transmitted signals over channels in wireless communication systems. These problems have to be corrected or controlled in order to have a reliable signal transmission in wirelesss communication channels. In order to overcome the problem of multipath fading effects, a reliable coding technique has to be engaged in the design of a wireless communication system. In this paper, a performance analysis of Convolutional and Gray coding techniques are investigated in terms of number of errors and bit error rates (BERs) using Rayleigh multipath channel. After the simulation of the two techniques, investigation reveals that convolutional coding is more preffered to Gray coding in a wireless communication system due to its low BERs.

Keywords – Convolutional coding, *Gray* coding, BER, AGWN channel, Rayleigh multipath fading channel.

1. Introduction

In a wireless communication system, data is transmitted across a channel from a transmitter to a receiver. The channel is most often affected by the fading effects called noise. This noise affects the signal by introducing errors or distortion into the data being transmitted. In order to retrieve the original signal from the receiver with minimal or no errors, a good channel coding technique is employed. This is done by an error control encoder-decoder (codec) pair whose primary function is to enhance the reliability of a message during the transmission of information through a channel. At the front end of the receiver, the distorted transmitted signal is recovered. The number of bit errors and BERs of the recovered signal depends on the amount of noise and interference in the communication channel [1].

- *Simon Wasinya Pallam, M.Eng. ElectricalEngineering, Lecturer at Electrical & Electronics Engineering department, Modibbo Adama University of Technology, Yola, Nigeria. E-mail: spallam@yahoo.com*
- George Adinoyi Audu, M.Eng. Electrical Engineering, *Lecturer at Electrical Engineering department, Bayero University Kano, Kano, Nigeria. E-mail: georgeaudu@yahoo.com*
- *Saidu Yerima Musa, PhD Power and Machines, Lecturer at Electrical & Electronics Engineering department, Modibbo Adama University of Technology, Yola, Nigeria. E-mail: saiduymusa@yahoo.com*
- *Ibrahim Musa Visa, M.Eng. Electrical Engineering, Lecturer at Electrical & Electronics Engineering department, Modibbo Adama University of Technology, Yola, Nigeria E-mail: visaibrahim@gmail.com*

There are various channel coding techniques used for error controls in wireless communication systems. Few of the available techniques are Convolutional coding and Gray coding. Convolutional codes are used for real time error correction (RTEC) while Gray coding, when combined with Forward Error Correction (FEC) codes, can aid in correction of erroneous reception of bits that spill into adjacent symbols. There are basically two popular error correcting schemes in communication systems: Forward Error Correction (FEC) and Automatic Repeat Request (ARR). These schemes can be used to improve the noisy output of the fading channel [2]. The performance analysis of Convolutional coding will be compared with that of Gray coding in terms of BERs. The technique that achieves lower number of bit errors or bit error rates will be adduced a preferred technique for implementation in the design of a wireless communication system. The two techniques will be simulated and their respective graphical reports presented.

2. Physical Layer of Wireless Communication

A typical wireless communication system consists of the source, encoder, modulator, channel, demulator, decoder, and sink components. These components can be grouped into three blocks, namely: Transmitter, Channel and Receiver. The source, encoder and modulator form the transmitter block while the demodulator, decoder and sink form the receiver block. The source generates a stream of data, processed by a binary converter, encoded by the encoder, modulated by M-ary Quadrature Amplitude Modulator (M-QAM) and transmitted across the channel. The source encoder compresses the source data in order to reduce the bandwidth required to transmit the signal while the channel encoder introduces redundancy aimed at protecting the information being transmitted over a fading channel so that the original information is recovered at the receiver. The complete block diagram of a typical wireless communication system is shown in Fig. 1.

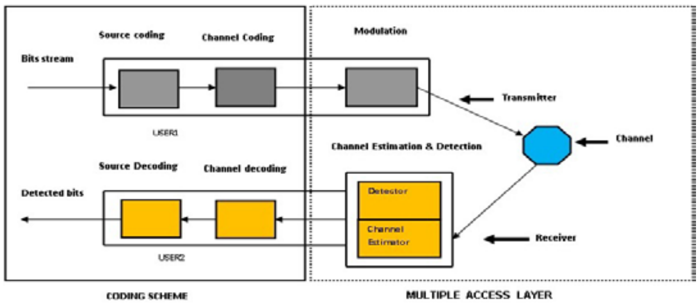


Fig.1 Block diagram of a wireless communication system

2.1 THE TRANSMITTER

The transmitter block of a wireless communication system consists of source and channel encoders. Source coding is a process of compressing the data into a required bandwidth before transmitting it across the channel. The tranmit data across the channel is susceptible to errors due to interference from other users, thermal noise and fading effects. Let

$$x(t)=\Re\{x_b(t)e^{j2\pi ft}\} \qquad (1)$$

where $x_b(t)$ is the baseband signal, f is the carrier frequency and t is the time. The transmit signal reaches the receiver through multiple paths where n^{th} path has an attenuation $\alpha_n(t)$ and delay $\tau_n(t)$ [3]. The process of detecting and correcting these errors by applying coding to the transmitted bits is called channel coding. There are different types of coding techniques used in wireless communication systems. Some of these techniques are convolutional and gray codings.

In 3G and 4G wireless systems, QAM modulation, an attractive technique for achieving a high-rate transmission over wireless links without increasing bandwidth, is recommended. To function satisfactorily, QAM communication systems require high Signal-to-Noise ratio (SNR) to combat the harsh wireless environment [4]. Multiple access scheme can be used to transmit the modulated signal of different users with different spreading sequence across the same channel. The intermediate signal, composed of the superimposed chip streams of different users, entering the channel is called the baseband signal. The chipping or spreading sequence is shown in Fig. 2.

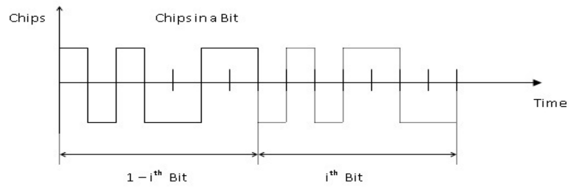


Fig. 2 Modulated signal chips

2.2 THE CHANNEL

Several transmitters send their encoded chip streams in the same frequency band and possibly at all instants across the channel. The transmission is a totally asynchronous process since users can transmit their sequences at any instant. These signals are superimposed in the channel and attenuated in strength. The channel introduces a noise, commonly modeled as Additive White Gaussian Noise (AWGN) with a constant Bit power-to-Noise spectral density ratio *(E_b/N_o)* into the signal. The Symbol power-to-Noise spectral density ratio *(Es/No)* is given as:

$$Es/No(dB) = Eb/No(dB) + log10(k) \qquad (2)$$

where k is bits per sample [5]. *Eb/No* is closely related to the Carrier-to-Noise ratio *(CNR)*, which is the Signal-to-Noise ratio *(SNR)* of the received signal after filtering, but just before detection.

$$CNR = Eb/No\ x\ \ fb/B \qquad (3)$$

Where fb is the channel data rate and B is the channel bandwidth. The equivalent logarithmic expression is given as:

$$CNR(dB) = \ 10log10(\tfrac{Eb}{No}) + 10log10(\tfrac{fb}{B}) \qquad (4)$$

The composite signal from the channel results from multiple signals bouncing off obstacles, suffering varying delays and attenuations and getting superimposed in chip streams. This is called the multipath effect. The multipath channel also introduces fading, called Rayleigh fading, which models scattered signal of wave between the transmitter and receiver, i.e. the attenuation of the different paths varies with time. The fading results from the relative motion of the transmitter and the receiver (Doppler Effect) or the movement of the reflectors in the path of the radio signals, especilally in urban environment. The received signal y from the channel can be represented as:

$$y = hx + N_o \qquad (5)$$

where N_o is the noise contributed by AWGN, x is the transmitted symbols and h is the Rayleigh fading response. For a simple AWGN channel without Rayleigh fading, the received signal is represented by:

$$y = x + N_o \qquad (6)$$

2.3 THE RECEIVER

The front-end of the receiver block is a demodulator and a chip-matched filter, which retrieves the baseband signal from the radio signal coming from the channel, for further processing. This continuous time baseband signal from the channel is converted to a discrete time signal by sampling the output of a filter matched to the chip waveform. Due to the asynchronous nature of the transmissions and the path delay introduced by the channel, the baseband signal corresponding to the bit-streams of different users is received in the receiver at different delayed instants by the multi-user detector. The multi-user detector, however, needs the knowledge of the bit boundaries of all the users for detection. The channel estimation, which is done by the channel parameter estimator, detects the arbitrary delay, magnitude and phase change introduced by the channel for all the users. The source and channel decoding are the receiver counterparts of the source and channel coding components in the transmitter block. The channel decoding extracts the bit stream and corrects some errors based on the property of the codes used. The source decoder decompresses the bits from the channel decoder. This corresponds to the original bit stream transmitted by the system. Thus the transmitted symbol x can be recovered from the received signal y by the process of equalization given by:

$$\hat{y} = \frac{y}{h} = \frac{hx+No}{h} = x + z \qquad (7)$$

$$y_{real} = real(\hat{y}) = real(x+z) \qquad (8)$$

where z is still the AWGN noise except for the scaling factor $1/h$ and y_{real} is the real value of detected signal [6].

3. CONVOLUTIONAL CODING TECHNIQUE

Convolutional codes are highly used for real time error correction (RTEC) in communication systems. The codes can convert the entire bits (data) stream into a single codeword. The two popular error correcting schemes used in communication systems are Forward Error Correction (FEC) and Automatic Repeat Request (ARR) [7]. These schemes can be used to effect corrections on the noisy output of the fading channel. The main decoding strategy for convolutional codes is based widely on Viterbi algorithm. Convolutional codes are usually described by two parameters: the code rate and the constraint length. The code rate, n/k, is the ratio of the total number of bits *(n)* passed into the convolutional encoder to the number of channel symbols *(k)* output by the encoder in a given encoder cycle. The constraint length parameter, *L*, denotes the 'length' of the convolutional encoder [8]. The data bits are fed in small groups of *k-bits* at a time to the shift register. The output of encoded bits are obtained by modulo-2 addition (EX – OR operation) of the input data bits [9]. The input data is shifted into the shift register a single bit at a time producing an n-tuple output a single bit at a time producing an *n-tuple* output for each shift. The convolutional code has a rate, *R=n/k*, associated with the encoder and a transfer function *G(x)* of *nxk* matrix. The Convolutional encoder uses the coding trellis technique to encode the binary data stream at the code rate of *n/k* .This also describes the operation of the corresponding decoder, especially when a Vertibi Algorithm is followed [10], [11].

3.1 SIMULATION OF CONVOLUTIONAL CODING

In this paper, Matlab simulation tool will be used to simulate coding and decoding of a stream of randomly generated data given the constraint lengths and generator polynomials of the convolutional codes. The total number of bits being processed in the simulation is $n=5x10^5$ while the oversampling rate nsamp=4. The convolutional coding scheme with a code rate of 2/3 is used. The signal source generates a binary data stream as a column vector x. A stem plot of the randomly generated signals is presented in Figs. 4.

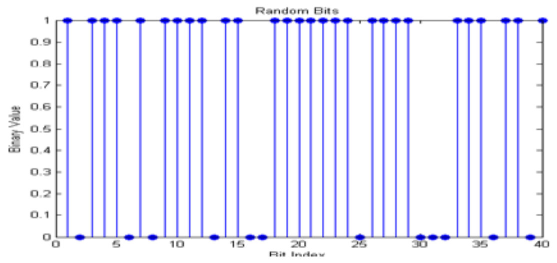


Fig. 4 Plot of Random signals

The Encoder uses the convolutional coding trellis technique to encode the binary data stream at the code rate of 0.667. The encoded bits in x are converted to *k-bit* symbols using Gray mapping scheme before mapping to specific arrangement of points in the 16-QAM. The 16-QAM is a type of M-ary QAM where *M=16*. In 16-QAM modulation scheme, 4 bit information per symbol can be sent across the channel [12].

$$k = log_2 M = log_2 16 = 4 \qquad (9)$$

The mapped bits which are being converted to integers are stem plotted as shown in Fig. 5.

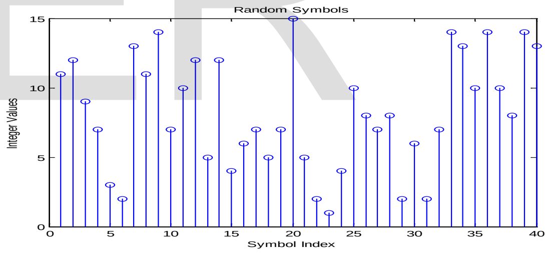


Fig. 5 Plot of Random symbols

The 16-QAM modulator modulates the signal at filter order of 40 and rolloff factor of 0.25. By using the Matlab command *rrcfilter=rcosine(1,nsamp,'fir/sqrt',rolloff,delay),* a square root raised cosine filter is applied to the unsampled transmitted signal. The impulse response and the eye diagram of filtered signal are created as shown in Figs. 6 and 7 respectively.

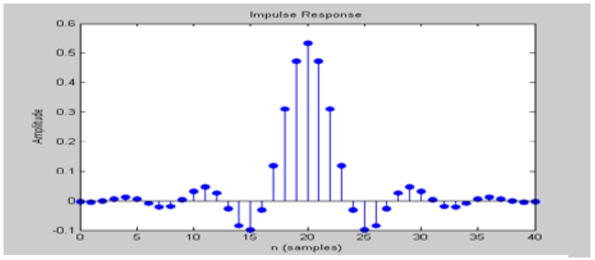


Fig. 6 Impulse response of filtered signal

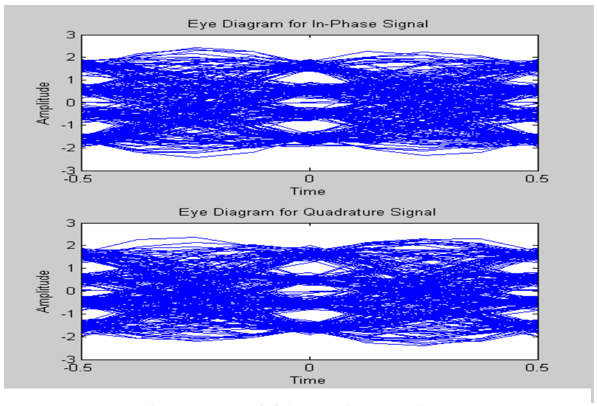


Fig. 7 Eye diagram of filtered signal

The command *ynoisy=awgn(ytx,snr,'measured')* is used to send the filtered signal over the AWGN channel.The signal from the channel is received at the receiver input using the square root raised cosine filter. A scatter plot of received signal before and after filtering is shown in Fig. 8.

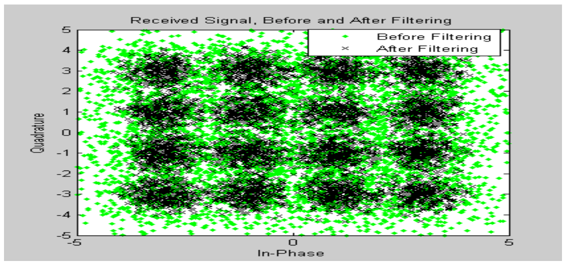


Fig. 8 Scatterplot of received signal

The 16-QAM demodulator is used to demodulate the received signal. At this point, the bit-to-symbol mapping done earlier on is undone while mapping decimal to binary. After decoding the convolutional codes, the BER is computed by comparing the transmitted binary stream 'x' and the received signal 'z'. The computed BER = 4.6x10^{-5} and the total number of errors = 29,250 were recorded during the run of the simulation. A table of simulation results is presented in table 1.

Table 1: Simulation results for convolutionally coded signal

X Bits	Encoded Bits	Xsym Mapped	Y (modulated)	Ytx (Transmitte)	Ynoisy (AWGN channel)	Yrx (Before filtering)	Yrx (After filtering)	Zsym (Demapped)	Decoded bits	Z bits	Parameters and Result
1	0	0	-1.0000 - 3.0000i	0.4877 + 0.9525i	1.5178 - 1.1009i	-2.9617 + 1.4712i	-0.8468 - 3.0987i	0	0	1	M=16
0	0	6	3.0000 - 3.0000i	0.3011 + 0.1335i	1.5552 - 1.7919i	-1.8352 - 0.2137i	3.2059 - 3.0158i	6	0	0	k=4
0	1	12	-1.0000 - 1.0000i	-0.0444 - 0.7536i	1.3599 - 1.1460i	-0.1652 - 1.3076i	-0.8539 - 1.1097i	12	1	0	n=5
0	1	6	-1.0000 - 3.0000i	-0.4467 - 1.3321i	0.3366 - 1.6322i	1.3327 - 3.0077i	-0.9283 - 1.9333i	6	1	0	n=3e5
0	1	6	-1.0000 - 3.0000i	-0.8032 - 2.0432i	1.8207 - 1.3807i	2.8139 - 3.6236i	-0.7678 - 2.6200i	6	1	0	nsamp=4
0	1	8	-1.0000 - 1.0000i	-1.0094 - 2.1958i	1.6340 - 0.7853i	3.4257 - 3.6193i	-0.5325 - 1.0233i	8	1	0	Coderate=0.667
0	1	4	-1.0000 + 3.0000i	-0.9507 - 1.9948i	1.4795 - 0.1608i	3.2946 - 3.1241i	-0.7881 + 2.8057i	4	1	0	Filtorder=40
1	0	9	1.0000 + 3.0000i	-0.5790 - 1.4896i	1.1865 + 1.0275i	2.5052 - 2.3556i	0.4623 + 2.9041i	9	0	1	delay=3
1	0	4	3.0000 - 1.0000i	0.0428 - 0.7566i	1.7985 + 1.9165i	1.2284 - 1.5312i	2.0938 - 1.0395i	4	0	1	Rolloff=0.25
0	0	2	-3.0000 + 1.0000i	0.7478 + 0.0911i	1.1410 + 1.3634i	-0.3087 - 0.7930i	-3.3466 + 1.4529i	2	0	0	EbNo=10
1	1	0	1.0000 + 3.0000i	1.3339 + 0.8927i	0.3321 + 0.8115i	-1.8392 - 0.1709i	1.5444 + 3.0189i	0	1	1	Snr=8.2391
1	1	8	3.0000 + 1.0000i	1.6200 + 1.5168i	0.3342 + 1.1938i	-3.0850 + 0.4032i	2.5051 + 1.3513i	8	1	1	tb=16
0	1	5	1.0000 - 3.0000i	1.5211 + 1.8698i	-0.2171 - 0.0465i	-3.8032 + 1.0388i	0.9143 - 2.8327i	5	1	0	decdelay=32
0	0	7	3.0000 - 3.0000i	1.1218 + 1.9309i	0.2426 - 0.8142i	-3.8407 + 1.8078i	3.2341 - 2.6321i	7	0	0	number_of_errors=31
1	0	9	-3.0000 + 3.0000i	0.6366 + 1.7627i	-0.1633 - 0.0733i	-3.2020 + 2.6801i	-2.9301 + 2.7322i	9	0	1	BER=6.2004e-005
0	0	7	-3.0000 + 1.0000i	0.3309 + 1.4647i	0.0607 + 0.1095i	-2.0373 + 3.5151i	-2.4134 + 1.3602i	7	0	0	
1	1	12	3.0000 - 3.0000i	0.3434 + 1.1425i	0.7149 + 1.1120i	-0.6812 + 4.1167i	2.8891 - 3.7553i	12	1	1	
1	1	0	3.0000 + 3.0000i	0.5931 + 0.8684i	0.9627 + 0.0637i	0.6168 + 4.2990i	2.9331 + 3.5940i	0	1	1	trellis t=
1	0	2	-3.0000 - 1.0000i	0.8030 + 0.6803i	2.3439 + 0.2740i	1.5849 + 3.9462i	-3.8470 - 1.5940i	2	0	1	numInputSymbols: 4
1	1	4	3.0000 + 1.0000i	0.6902 + 0.5431i	1.5898 - 0.0840i	2.0690 + 3.0383i	3.3146 + 1.0682i	4	1	1	numOutputSymbols: 8
1	1	2	3.0000 + 1.0000i	0.1679 + 0.4357i	1.5804 - 0.7025i	2.0065 + 1.6624i	3.4123 + 1.4663i	2	1	1	numStates: 128
1	1	0	3.0000 + 3.0000i	-0.6064 + 0.3648i	1.2970 - 1.4243i	1.4220 - 0.0059i	2.9894 + 2.9497i	0	1	1	
0	1	0	-3.0000 - 3.0000i	-1.3148 + 0.3708i	1.0386 - 1.5369i	0.4336 - 1.7371i	-3.3146 - 3.6107i	0	1	0	
0	0	13	-3.0000 + 1.0000i	-1.6718 + 0.4554i	0.3576 - 0.6569i	-0.7286 - 3.2686i	-2.6919 + 0.6833i	13	0	0	
1	1	4	3.0000 - 1.0000i	-1.5678 + 0.5888i	-1.3002 + 0.5726i	-1.7611 - 4.3460i	2.7204 - 0.9301i	4	1	1	
1	1	0	3.0000 + 1.0000i	-1.1282 + 0.7004i	-2.2208 - 1.2418i	-2.3220 - 4.7787i	3.1841 + 0.8371i	0	1	1	
0	0	15	-3.0000 - 1.0000i	-0.6889 + 0.6929i	-0.8474 - 0.1245i	-2.1672 - 4.4820i	-3.2547 - 1.1699i	15	0	0	
0	0	4	-1.0000 - 1.0000i	-0.3665 + 0.5171i	-1.1337 + 0.0250i	-1.2711 - 3.5044i	-1.2677 - 0.9518i	4	0	0	
--	--	--	--	--	--	--	--	--	--	--	
--	--	--	--	--	--	--	--	--	--	--	

4. GRAY CODING TECHNIQUE

Gray coding is a form of coding technique used in many application such as encoders of absolute-position sensing and angle measurement systems. It is useful in systems where analog information is being converted to digital information or vice versa. Gray codes are often referred to as reflected codes, i.e. the adjacent symbols differ by only one bit from the other. The coding can aid in correction of erroneous reception of bits that spill into adjacent symbols. Digital modulation techniques such as M-ary phase shift keying (M-PSK) and M-ary quadrature amplitude modulation (M-QAM) use Gray coding scheme to represent symbols that are modulated. To represent a binary code $d_1d_2....d_{n-1}\,d_n$ in its corresponding Gray code, the process starts with the less significant bit (LSB) or d_n, moving towards the most significant bit (MSB) or d_1. If d_{n-1} is 1, d_n is replaced by $1-d_n$; otherwise it is left unchanged. The conversion process proceeds to d_{n-1} and continues up to d_1, which is kept the same since d_0 is assumed to be 0. The resulting code

$g_1g_2 \dots g_{n-1}g_n$ is the Gray code. Similarly, converting Gray code $g_1g_2 \dots g_{n-1}g_n$ to its corresponding binary code, the process starts from the nth digit as follows:

$$\sum_n = \sum_{i=1}^{n-1} gi(mod2) \qquad (10)$$

If $\sum_n$ is 1, then g_n is replaced by $1-g_n$; otherwise it is left unchanged. The next computation is:

$$\sum_{n-1} = \sum_{i=1}^{n-2} gi(mod2) \qquad (11)$$

The resulting binary code corresponds to the initial binary code $d_1d_2 \dots d_{n-1}d_n$ [13]. The following are simplified steps for converting the binary code $d_1d_2d_3d_4$ to Gray code $g_1g_2g_3g_4$:

1. The binary MSB d_1, equals the Gray MSB g_1.
2. Gray bit g_2 equals an Exclusive-OR of binary bits d_1 and d_2.
3. Gray bit g_3 equals an Exclusive-OR of binary bits d_2 and d_3.
4. Gray bit g_4 equals an Exclusive-OR of binary bits d_3 and d_4.

Similarly, the following steps apply to the conversion of Gray code $g_1g_2g_3g_4$ to binary code $d_1d_2d_3d_4$.

1. The Gray MSB g_1, equals the binary MSB d_1.
2. If the Gray bit g_2 is 0, the binary bit d_2 equals the previous bit d_1. If bit g_2 is 1, then bit d_2 equals the OR of bit d_1.
3. If Gray bit g_3 is 0, d_3 equals bit d_2. If bit g_3 is 1, then bit d_3 equals the OR of bit d_2.
4. If Gray bit g_4 is 0, d_4 equals bit d_3. If bit g_4 is 1, then bit d_4 equals the OR of bit d_3.

Table 2 shows the binary to Gray code conversion.

Table 2: table of binary and gray codes conversion

DECMAL VALUES	BINARY CODES	GRAY CODES	GRAY VALUES
0	000	000	0
1	001	001	1
2	010	011	3
3	011	010	2
4	100	110	6
5	101	111	7
6	110	101	5
7	111	100	4
8	1000	1100	12
9	1001	1101	13
10	1010	1111	15

4.1 SIMULATION OF GRAY CODED SIGNALS

This simulation includes coding and decoding of a stream of data $n=3x10^4$ with over sampling rate nsamp=1 and the size of signal constellation M=16. The signal source generates a binary data stream 'x' as a column vector. The encoded bits in 'x' are converted to k-bit symbols using Gray coding conversion scheme before mapping to specific arrangement of points in the 16-QAM (Quadrature amplitude modulator) constellation. The mapped bits are further converted to integers. Stem plots of the first 40 bits of random signals and the first 10 of the random symbols are presented in Figs. 9 and 10 respectively.

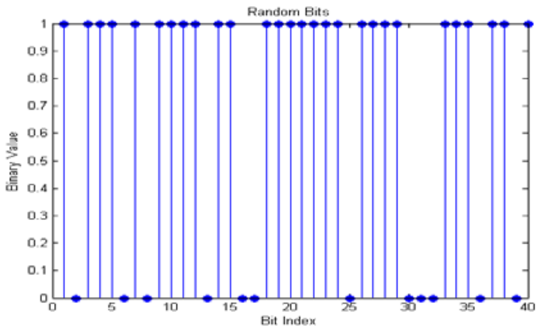


Fig. 9. Stem Plot random signals

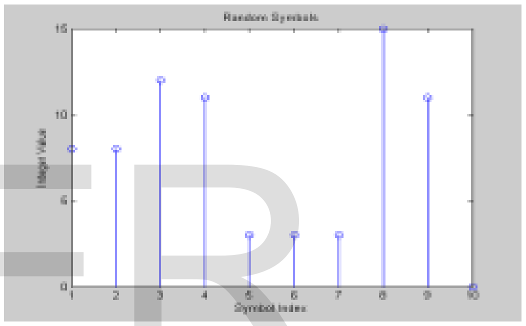


Fig. 10 Stem plot of random symbols

The 16-QAM modulated signal is sent over the AWGN channel with spectral noise density EbNo=10dB, using the command *ynoisy=awgn(ytx,snr,'measured')*. A scatterplot of a noisy signal at the channel output is shown in Fig. 11.

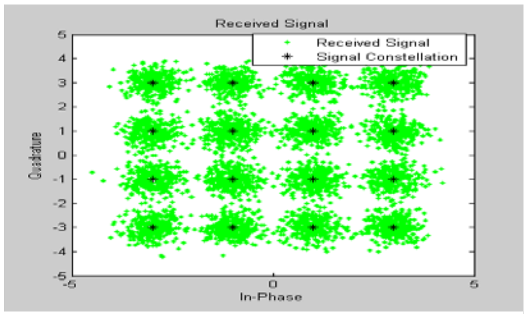
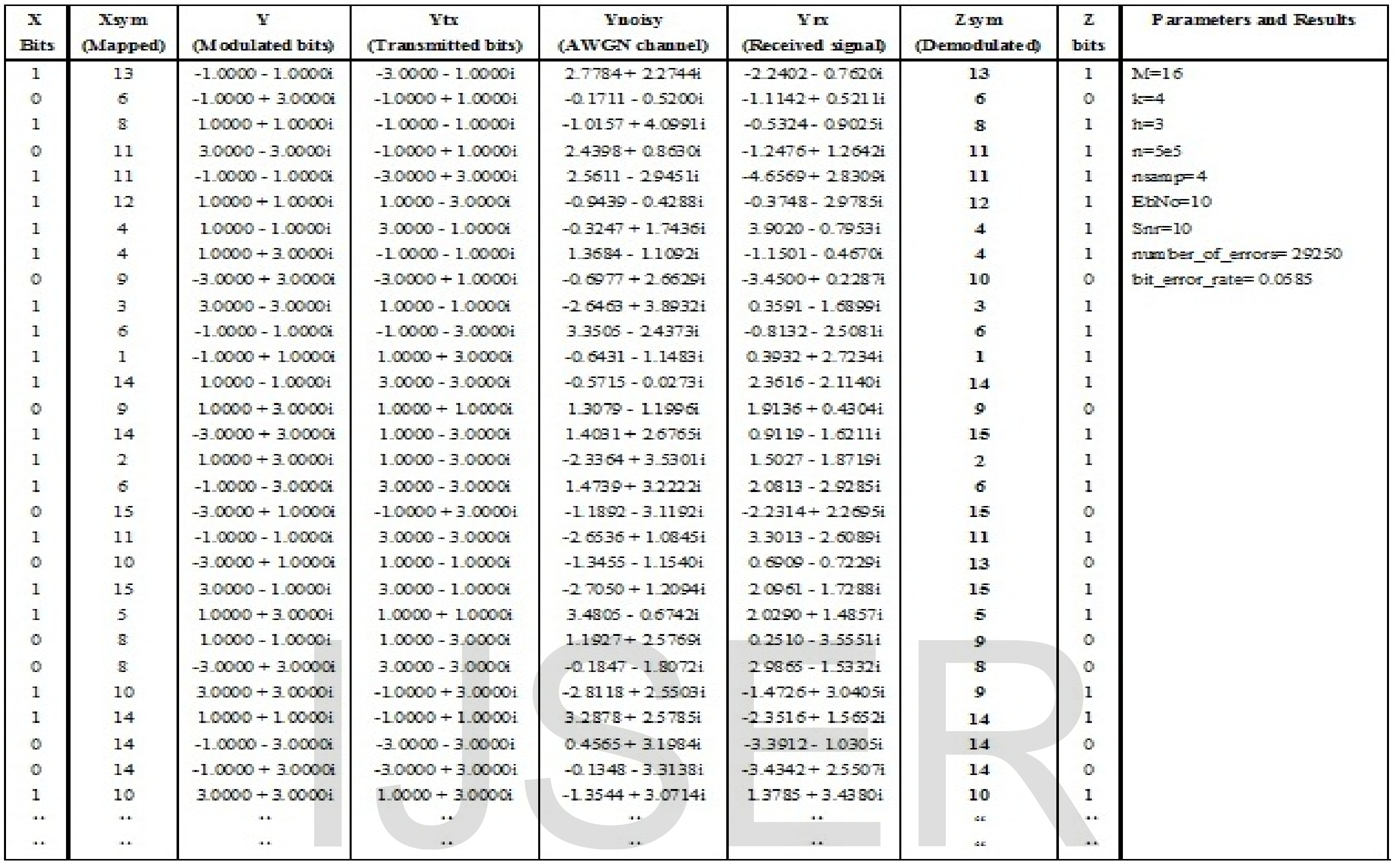


Fig. 11 Scatterplot of received signal

The 16-QAM demodulator is used to demodulate the received signal at the receiver block. At this point, the bit-to-symbol mapping earlier done is demapped, while decimal to binary bits are mapped. Finally, after the received signal is converted from matrix to a vector form z, the BER is computed by comparing transmitted binary stream *x* and the received signal *z*.

From the simulation result the computed BER=0.0585 and the number of errors=23 were recorded during the run of the simulation.

Table 3: Simulation results for Gray coded signal



X Bits	Xsym (Mapped)	Y (Modulated bits)	Ytx (Transmitted bits)	Ynoisy (AWGN channel)	Yrx (Received signal)	Zsym (Demodulated)	Z bits	Parameters and Results
1	13	-1.0000 - 1.0000i	-3.0000 - 1.0000i	2.7784 + 2.2744i	-2.2402 - 0.7620i	13	1	M=16
0	6	-1.0000 + 3.0000i	-1.0000 + 1.0000i	-0.1711 - 0.5200i	-1.1142 + 0.5211i	6	0	k=4
1	8	1.0000 + 1.0000i	-1.0000 - 1.0000i	-1.0157 + 4.0991i	-0.5324 - 0.9025i	8	1	h=3
0	11	3.0000 - 3.0000i	-1.0000 + 1.0000i	2.4398 + 0.8630i	-1.2476 + 1.2642i	11	1	n=5e5
1	11	-1.0000 - 1.0000i	-3.0000 + 3.0000i	2.5611 - 2.9451i	-4.6569 + 2.8309i	11	1	nsamp=4
1	12	1.0000 + 1.0000i	1.0000 - 3.0000i	-0.9439 - 0.4288i	-0.3748 - 2.9785i	12	1	EbNo=10
1	4	1.0000 - 1.0000i	3.0000 - 1.0000i	-0.3247 + 1.7436i	3.9020 - 0.7953i	4	1	Snr=10
1	4	1.0000 + 3.0000i	-1.0000 - 1.0000i	1.3684 - 1.1092i	-1.1501 - 0.4670i	4	1	number_of_errors= 29250
0	9	-3.0000 + 3.0000i	-3.0000 + 1.0000i	-0.6977 + 2.6629i	-3.4500 + 0.2287i	10	0	bit_error_rate= 0.0585
1	3	3.0000 - 3.0000i	1.0000 - 1.0000i	-2.6463 + 3.8932i	0.3591 - 1.6899i	3	1	
1	6	-1.0000 - 1.0000i	-1.0000 - 3.0000i	3.3506 - 2.4373i	-0.8132 - 2.5081i	6	1	
1	1	-1.0000 + 1.0000i	1.0000 + 3.0000i	-0.6431 - 1.1483i	0.3932 + 2.7234i	1	1	
1	14	1.0000 - 1.0000i	3.0000 - 3.0000i	-0.5715 - 0.0273i	2.3616 - 2.1140i	14	1	
0	9	1.0000 + 3.0000i	1.0000 + 1.0000i	1.3079 - 1.1996i	1.9136 + 0.4304i	9	0	
1	14	-3.0000 + 3.0000i	1.0000 - 3.0000i	1.4031 + 2.6765i	0.9119 - 1.6211i	15	1	
1	2	1.0000 + 3.0000i	1.0000 - 3.0000i	-2.3364 + 3.5301i	1.5027 - 1.8719i	2	1	
1	6	-1.0000 - 3.0000i	3.0000 - 3.0000i	1.4739 + 3.2222i	2.0813 - 2.9285i	6	1	
0	15	-3.0000 + 1.0000i	-1.0000 + 3.0000i	-1.1892 - 3.1192i	-2.2314 + 2.2695i	15	0	
1	11	-1.0000 - 1.0000i	3.0000 - 3.0000i	-2.6536 + 1.0845i	3.3013 - 2.6089i	11	1	
0	10	-3.0000 + 1.0000i	1.0000 - 1.0000i	-1.3455 - 1.1540i	0.6909 - 0.7229i	13	0	
1	15	3.0000 - 1.0000i	3.0000 - 1.0000i	-2.7050 + 1.2094i	2.0961 - 1.7288i	15	1	
1	5	1.0000 + 3.0000i	1.0000 + 1.0000i	3.4806 - 0.6742i	2.0290 + 1.4857i	5	1	
0	8	1.0000 - 1.0000i	1.0000 - 3.0000i	1.1927 + 2.5769i	0.2510 - 3.5551i	9	0	
0	8	-3.0000 + 3.0000i	3.0000 - 3.0000i	-0.1847 - 1.8072i	2.9866 - 1.5332i	8	0	
1	10	3.0000 + 3.0000i	-1.0000 + 3.0000i	-2.8118 + 2.5503i	-1.4726 + 3.0405i	9	1	
1	14	1.0000 + 1.0000i	-1.0000 + 1.0000i	3.2878 + 2.5785i	-2.3516 + 1.5652i	14	1	
0	14	-1.0000 - 3.0000i	-3.0000 - 3.0000i	0.4565 + 3.1984i	-3.3912 - 1.0305i	14	0	
0	14	-1.0000 + 3.0000i	-3.0000 + 3.0000i	-0.1348 - 3.3138i	-3.4342 + 2.5507i	14	0	
1	10	3.0000 + 3.0000i	1.0000 + 3.0000i	-1.3544 + 3.0714i	1.3785 + 3.4380i	10	1	
..	..	..	..	..	..	..	..	
..	..	..	..	..	..	..	..	

5. CONCLUSION

In this paper, simulations of convolutional coding and Gray coding were carried out and the performance results presented. From the performance analysis, the convolutional coding technique is more efficient than Gray coding in their applications in wireless communication system. The convolutional coding has a lower number of bit errors and BERs in a multipath fading channel than its counterpart. This work offers an insight into the development of more efficient technique for coding signals in a wireless communication system. The convolutional encoder provides an effective error control mechanism with low BERs. An effective encoding technique such as this is an important element in the development of wireless communication system.

REFERENCES

[1] Sishir Kalita, Parismata Gogoi, and Kandarpa Kumar Sarma, "*Convolutional Coding Using Booth Algorithm for Application in Wireless Communication*" Internal Journal of Electronic Signals and Systems.

[2] Mamta Arora and Hardeep Kaur, "Performance Analysis of Communication System With Convolutional Coding Over Fading Channel", *International Journal for Scienntific and Engineering Research, Volume 4, Issue 5, May-2013.*

[3] Kazi Mohitul Islam, Habib Mohammad Nazir Ahmad, Chowdhury Akram Hossain and A K M Arifuzzman, "performance comparison between Traditional and Gray-mapped 16-QAM Scheme with OFDM in both AWGN and Rayleigh Fading Channel" IJCIT, Volume 01, issue 02, 2011.

[4] Ki Seol Kim, Kwangmin Hyun, Chan Wahn Yu, Youn Ok Park, Dongwean Yoon, and Sang Kyu Park, "General Log-Likelihood Ratio Expression and its Implementation Algorithm for Gray-Coded QAM Signals", ETRI Journal, Volume 28, Number 3, June 2006.

[5] Chris Heagard and Stephen B. Wicker, "Turbo coding", Kluwer p.3. ISBN 978-0-7923-8378-9, 1999

[6] http://www.gaussianwaves.com

[7] Nabeel Arshad and Abdul Basit, *"Implementation and analysis of covulational codes using Matlab", International Journal of Multidisciplinary Sciences and Engineering, Vol. 3, No. 8, August 2012.*

[8] S. Haykin: "*Digital Communication – Fundamental And Application", Second ed., Pearson Publication 2005.*

[9] J. G.Proakis, "Digital Communication", McGraw-Hill Series, 2001.

[10] A. J. Viterbi, *"Error Bounds For Convolutional Codes And Asymptotically Optimum Decoding Algorithm", IEEE Transactions on Information Theory, Vol. IT-13, pp.260-269, April, 1967.*

[11] T. S. Rapport: *"Wireless Communication: Principle and Practice", Pearson Educational International, 2nd ed., 2002.*

[12] John R. Barry, Edward A. Lee, David G. Messerschmitt "Digital Communication, "Kluwer academic publishers , 3rd ed. ch-5, pp-164-184,2003.

[13] Online Electrical Engineering study site http://www.electrical4u.com/gray-code-binary-to-gray-code-and-that-to-binary-conversion

APPENDIX

A. PROGRAM FOR CONVOLUTIONAL CODING

```
% Convolutional coding technique
M = 16; % Size of signal constellation
k = log2(M); % Number of bits per symbol
n = 5e5; % Number of bits being processed
nsamp = 4; % Oversampling rate
x = randint(n,1); % Random bit stream as column vector
stem(x(1:40),'filled'); title('Random Bits'); xlabel('Bit Index');
ylabel('Binary Value'); %Stem plot the first 40 data bits.
% Define a convolutional coding trellis for encoding the data
t = poly2trellis([5 4],[23 35 0; 0 5 13]); % Trellis
code = convenc(x,t); coderate = 2/3;% Encode.
% Define a vector for mapping bits to symbols.
mapping = [0 1 3 2 4 5 7 6 12 13 15 14 8 9 11 10].';
% Do ordinary binary-to-decimal mapping.
xsym = bi2de(reshape(code,k,length(code)/k).','left-msb');
% Map codes using Gray mapping scheme
xsym=mapping(xsym+1);
% Stem Plot the first 40 symbols.
figure;  stem(xsym(1:40)); title('Random Symbols');
xlabel('Symbol Index'); ylabel('Integer Value');
y = qammod(xsym,M); % Modulate using 16-QAM Modulator.
filtorder = 40; % Define Filter order for filtering
delay=filtorder/(nsamp*2); % Group delay ( input samples)
rolloff = 0.25; % Rolloff factor of filter
% Create a square root raised cosine filter.
rrcfilter = rcosine(1,nsamp,'fir/sqrt',rolloff,delay);
figure; impz(rrcfilter,1); % Plot impulse response.
% Upsample and apply square root raised cosine filter.
ytx = rcosflt(y,1,nsamp,'filter',rrcfilter);
% Create eye diagram for part of filtered signal.
eyediagram(ytx(1:2000),nsamp*2);
% Sending signal over an AWGN Channel
EbNo = 10; % Ratio of bit power-to-noise spectral density
snr = EbNo + 10*log10(k*coderate)-10*log10(nsamp);
ynoisy = awgn(ytx,snr,'measured');
% Filter received signal using square root raised cosine filter.
yrx = rcosflt(ynoisy,1,nsamp,'Fs/filter',rrcfilter);
yrx = downsample(yrx,nsamp); % Downsample.
yrx = yrx(2*delay+1:end-2*delay); % Account for delay.
% Scatter Plotting received signal before and after filtering.
h=scatterplot(sqrt(nsamp)*ynoisy(1:nsamp*5e3),nsamp,0,'g.');
hold on;  scatterplot(yrx(1:5e3),1,0,'kx',h);
title('Received Signal, Before and After Filtering');
legend('Before Filtering','After Filtering');
axis([-5 5 -5 5]); % Set axis ranges.
zsym=qamdemod(yrx,M); % Demodulate using 16-QAM.
% Symbol-to-Bit Mapping
[dummy demapping] = sort(mapping);
% Initially, demapping has values between 1 and M.
% Subtract 1 to obtain values between 0 and M-1.
demapping = demapping - 1;
zsym = demapping(zsym+1); % Gray to binary mapping.
z = de2bi(zsym,'left-msb'); % Decimal-to-binary mapping.
% Convert z from matrix to vector.
z=reshape(z.',prod(size(z)),1);  % Convert z in matrix to vector.
tb = 16; % Traceback length for convolutional decoding
z = vitdec(z,t,tb,'cont','hard'); % Decode.
% Compare x and z to compute the number of errors and BER
decdelay = 2*tb; % Decoder delay, in bits
[number_of_errors,bit_error_rate] = ...
biterr(x(1:end-decdelay),z(decdelay+1:end))
```

B. MATLAB PROGRAM FOR GRAY CODING

```
% Gray coding technique
M = 16; % Size of signal constellation
k = log2(M); % Number of bits per symbol
n = 3e4; % Number of bits to process
nsamp = 1; % Oversampling rate
x = randint(n,1); % Create random column vector bit stream
stem(x(1:40),'filled'); title('Random Bits'); xlabel('Bit Index');
ylabel('Binary Value'); % Plot the first 40 bits in a stem plot.
% Bit-to-Symbol Mapping
mapping = [0 1 3 2 4 5 7 6 12 13 15 14 8 9 11 10].';
% Ordinary binary-to-decimal mapping.
xsym = bi2de(reshape(x,k,length(x)/k).','left-msb');
xsym = mapping(xsym+1); % Binary to Gray mapping.
% Stem Plot the first 10 symbols
figure; stem(xsym(1:10)); title('Random Symbols');
xlabel('Symbol Index'); ylabel('Integer Value');
y = qammod(xsym,M); % Modulation using 16-QAM.
ytx = y; % Transmit Signal
EbNo = 10; % spectral noise density in dB
snr = EbNo + 10*log10(k) - 10*log10(nsamp);
% Send signal over AWGN Channel
ynoisy = awgn(ytx,snr,'measured');
yrx = ynoisy; % Received Signal
% Scatter plot received and transmitted signals.
h = scatterplot(yrx(1:nsamp*5e3),nsamp,0,'g.'); hold on;
scatterplot(ytx(1:5e3),1,0,'k*',h); title('Received Signal');
legend('Received Signal','Signal Constellation');
axis([-5 5 -5 5]); hold off  % Set axis ranges.
zsym = qamdemod(yrx,M); % Demodulation using 16-QAM.
%% Symbol-to-Bit Demapping
[dummy demapping] = sort(mapping);
% Initially, demapping  has values between 1 and M.
% Subtract 1 to obtain values between 0 and M-1.
demapping = demapping - 1;
zsym = demapping(zsym+1); % Gray to binary mapping.
z = de2bi(zsym,'left-msb'); % Decimal-to-binary mapping.
```

```
z = reshape(z.',prod(size(z)),1); % Convert z in matrix to
vector.
% Compare x and z to compute the number of errors and BER.
[number_of_errors,bit_error_rate] = biterr(x,z)
```