



Hard Decision Decoding of Binary Digital Signals

Project Submitted to Electrical Engineering Department-
College of Engineering – Salahaddin University

Prepared by:
Nura Nafaih Shoukry
Zainab Kamal Mustafa

Supervised by:
Maha George Zia
Assistant Professor

1. Abstract:

There are three basic elements to every communication system, namely, transmitter, channel, and receiver. The transmitter is located at one point in space, the receiver is located at some other point separate from the transmitter, and the channel is the physical medium that connects them together as an integrated communication system.

The purpose of the transmitter is to convert the message signal produced by the source of information into a form suitable for transmission over the channel. However, as the transmitted signal propagates along the channel, it is distorted due to channel imperfections. Moreover, noise and interfering signals (originating from other sources) are added to the channel output, with the result that the received signal is a corrupted version of the transmitted signal.

The receiver has the task of operating on the received signal to reconstruct a recognizable form of the original message signal for an end user or information sink.

In our project, a convolutional encoder-decoder relates to BPSK signal or BFSK to mitigate the distortion in the received signal and hence to improve the receiver system performance by calculating its bit error rate performance. We build and simulate the system using MATLAB program.

2.3 Convolutional Encoder Structure

Convolutional codes have memory that uses previous bits to encode or decode following bits. The code rate $R=k/n$, where k are the input bits and n are the output coded bits. Fig. 6 shows a convolutional encoder of $R = 1/2$ and generators $g_1=[1\ 0\ 1]$, and $g_2=[1\ 1\ 1]$. It consists of two connected delay elements (a shift register) and two modulo-2 adders (EXORs). The output of a delay element time t is equal to its input at time $t - 1$. The time t is integer [3,4].

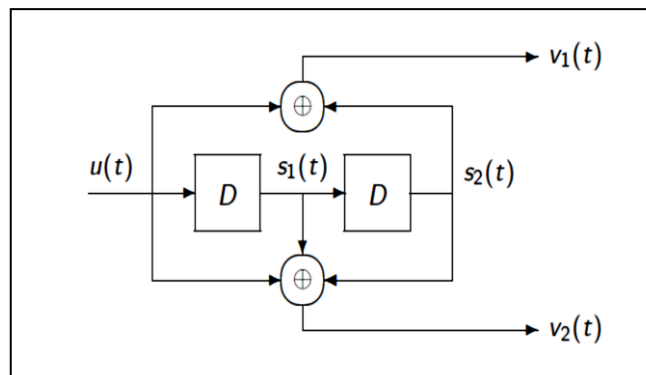


Fig. 6 shows a convolutional encoder of $R = 1/2$, generators $g_1=[1\ 0\ 1]$, and $g_2=[1\ 1\ 1]$

Table 1: Transition table of encoder, $R=1/2$, generators $g_1=[1\ 0\ 1]$, and $g_2=[1\ 1\ 1]$.

Input	Present states		Output		Next states	
$u(t)$	$s_1(t-1)$	$s_2(t-1)$	$v_1(t)$	$v_2(t)$	$s_1(t)$	$s_2(t)$
0	0	0	0	0	0	0
1	0	0	1	1	1	0
0	0	1	1	1	0	0
1	0	1	0	0	1	0
0	1	0	0	1	0	1
1	1	0	1	0	1	1
0	1	1	1	0	0	1
1	1	1	0	1	1	1

The state diagram is shown in Fig.7.

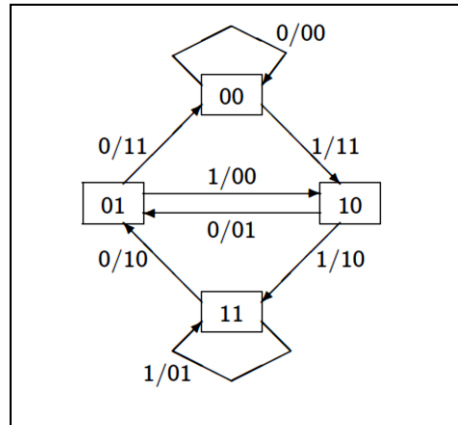


Fig.7. State diagram of the encoder of $R=1/2$, generators $g_1=[1\ 0\ 1]$, and $g_2=[1\ 1\ 1]$

2.4 Decoder of Convolutional Codes [3,4,6]:

The convolutional decoder can be represented by trellis diagram. Fig. 8 shows the trellis diagram of the encoder of $R=1/2$, generators $g_1=[1\ 0\ 1]$, and $g_2=[1\ 1\ 1]$, and using state transitions with state diagram. The horizontal axis is the time-axis. States are denoted by $s_1(t)s_2(t)$, and along the branches are again the input and outputs $u(t)=v_1(t)v_2(t)$.

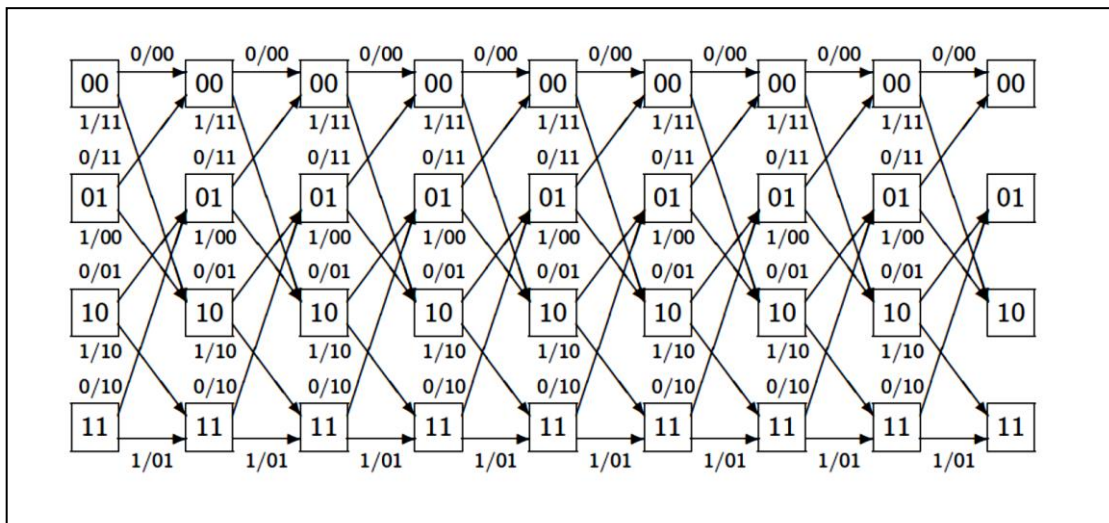


Fig.8.Trellis diagram of the encoder of R=1/2, generators $g_1=[1\ 0\ 1]$, and $g_2=[1\ 1\ 1]$

3. Problem statement:

- 1) To design and simulate a hard decision decoder of binary signals (PSK, and FSK) by using a convolutional encoder-decoder to mitigate the distortion in the received signal.
- 2) To improve the receiver system performance by calculating the bit error rate.
- 3) We build and simulate the system using MATLAB program.

4-Feasibility study:

Hard decision decoding using Viterbi algorithm and Trellis diagram is very efficient and accurate method, but its complexity is increased because this method uses Maximum likelihood decoding that find the code branch in the code trellis that was most likely to transmitted. Therefore, maximum likelihood decoding is based on calculating the hamming distances for each branch forming encode word and this task becomes very difficult and takes time as the number of the code words are increased. No hardware components are used because we use MATLAB in our project so we don't need to spend money, only write the program and simulate it.

5-Project Plan: it is based on the following flow chart shown in Fig.9. This flow chart describes the method of hard decision decoding of binary signals using Viterbi algorithm.

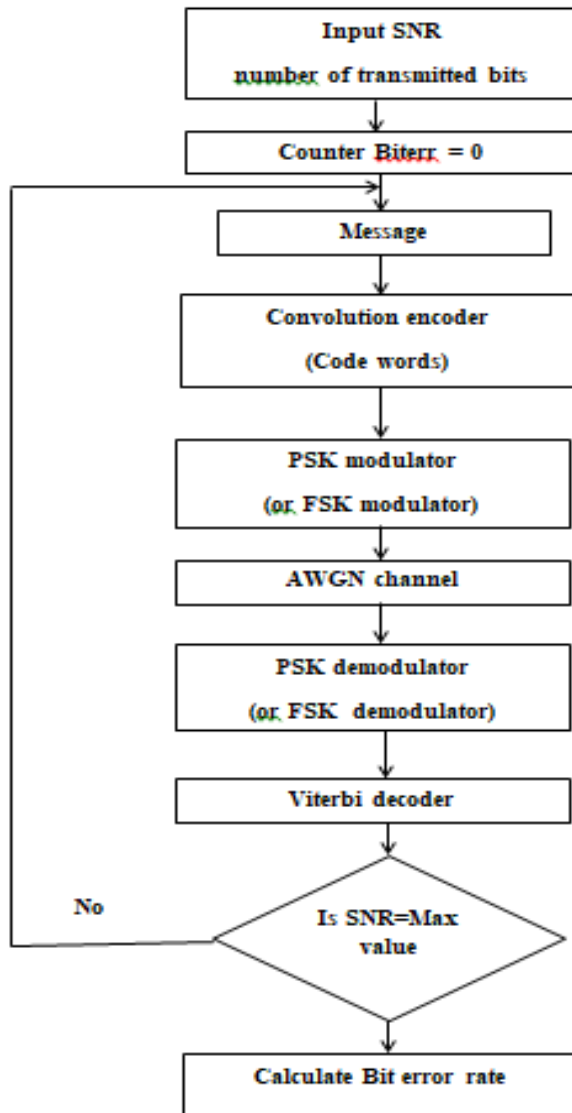


Fig. 9 Flow Chart of hard decision decoding of binary digital signals using convolutional coding.

6- Project Objectives:

In our project, binary digital signals (BPSK, and BFSK) are transmitted via additive white Gaussian noise channel (AWGN). The effect of AWGN is permanent and try to distort the transmitted symbols due to the thermal effect of the system. Therefore, a convolutional encoder-decoder is connected in our system to mitigate the distortion in the received signal and hence to improve the receiver system performance by calculating the bit error rate. We build and simulate the system using MATLAB program.

7- Gantt Chart:

The Gantt chart of our project is a management tool that illustrates our work completed over a period started on 19th Feb.2023 in relation to the time planned for the work which ends in May 2023. Table 2 shows the Gantt chart that includes the start and end dates of tasks, and dependencies between tasks.

Table 2: Gantt Chart of our project

Task Name	26/2/2023	5/3/2023	12/3/2023	19/3/2023	26/3/2023	2/4/2023	9/4/2023	16/4/2023	23/4/2023
System theory									
Study program function									
Literature review									
Development of methodology									
Program design									
Program simulation									
Implementation									

8. Literature review:

- Simon Haskin's published a technical book focused on the theory and practical applications of digital communication systems. It describes signaling over Additive White Gaussian Noise (AWGN) Channels, the impairment is the unavoidable presence of channel noise, which is modeled as additive white Gaussian noise (AWGN). This model is well-suited for the study of phase-shift keying (PSK), quadrature-amplitude modulation (QAM), and frequency-shift keying (FSK) as different ways of accommodating the transmission and reception of binary data [1].
- The fundamental concepts and techniques used in digital communication, including modulation, channel coding, and error control coding were covered in [2]. It also includes baseband digital transmission, as well as convolutional coding.
- F.M.J. Willems, explained in part 1 an introduction to convolutional codes. The book introduces convolutional codes which are a class of error-correcting codes that are used to transmit digital information over noisy communication channels [3].
- The article in [4] show how to construct new convolutional codes from old ones by applying the well-known techniques: puncturing, extending, expanding, direct sum. By applying these methods, several new families of convolutional codes can be constructed. As an example of code expansion, families of convolutional codes derived from classical cyclic codes and character codes were constructed.
- S. Sharma published a comprehensive textbook that covers the fundamental concepts of analog and digital communication systems. The digital communication system covers the binary ASK, PSK and FSK signals. The book provides understanding of the information theory that include source and channel coding [5].
- The paper in [6] is a tutorial introduction to the Viterbi algorithm (VA), its structure, and its analysis, where it has a significant impact on understanding in the theory of convolutional codes.

9. Methodology:

Fig. 10 shows a binary digital signal system with convolutional encoder-decoder (that employ Viterbi algorithm).

In our project, the input message (msg) is random and represented by binary symbols that will be fed to BPSK(or BFSK) modulator.

The modulated signal is passing through a convolutional encoder with generator polynomial (3, [6 7]) of rate $R=1/2$. Then the code words are now passing through Additive white Gaussian noise (AWGN) channel, which is considered as a must due to the thermal effects of the system.

The signal after the channel is distorted and is passed to a demodulator. The demodulated signal is applied to a trellis decoder that uses a Viterbi algorithm to mitigate the noise and to improve receiver system performance. Finally, the bit error rate (BER) performance of both PSK and FSK systems are measured and plotted using MATLAB program.

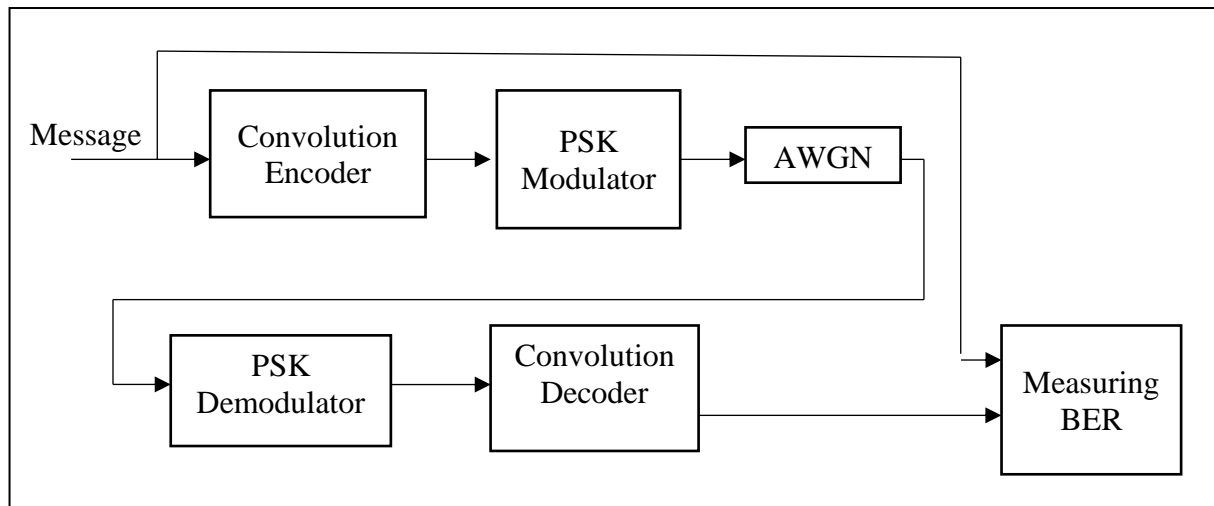


Fig. 10 Binary digital signal system using convolutional encoder-decoder.

10-Implementation

We connect the binary digital signal system using convolutional encoder-decoder as shown in Fig,10.

This system is simulated using MATLAB program with the used program functions listed in Appendix (A).

The bit error rate (BER) is measured versus signal-to-noise ratio (snr) for binary phase shift keying (BPSK) and binary frequency shift keying (BFSK) using convolutional encoder-decoder.

11- Deployment & testing

Fig. 11 shows the plot of BER vs. snr for coded BPSK and coded BFSK. The total number of transmitted bits is 100000 bits as 0 and 1. The snr values varies from 0 to 12. The generator polynomial for convolutional encoder is (3, [6 7]) of rate $R=1/2$. The decoder uses Trellis algorithm.

The simulation results for measuring the BER vs. snr for BPSK and BFSK are tabulated in Table (3) using convolutional encoder-decoder.

The simulation results show that the BER performance for coded BPSK is better than BER performance of coded BFSK.

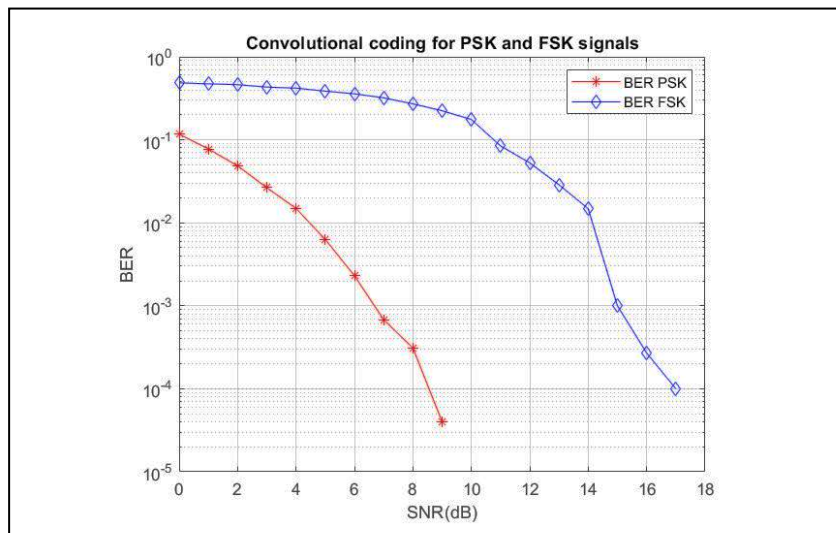


Fig. 11 BER vs. snr for coded BPSK and coded BFSK using (3, [6 7]) convolutional encoder-decoder.

Table (3) Simulation results of BER vs. snr for coded BPSK and coded BFSK

snr	BER for coded BPSK	BER for coded FPSK
0	0.1167	0.4858
1	0.0776	0.4723
2	0.0485	0.4616
3	0.0265	0.4317
4	0.0149	0.4193
5	0.0063	0.3858
6	2.3e-3	0.3579
7	6.8e-4	0.3204
8	3.1e-4	0.2711
9	4e-5	0.2239
10	0	0.1756
11	0	0.0844
12	0	0.0527
13	0	0.0287
14	0	0.0148
15	0	1e-3
16	0	0.00027
17	0	1e-4

References:

1. Simon Haykin ,“*Digital Communications System*”, Copyright © 2014 John Wiley & Sons, Inc.
2. Ali Grami ,” *Introduction to Digital Communications*”, ELSEVIER, 2016.
3. Frans M.J. Willems,” *Introduction to Convolutional Codes, Part 1*”, Eindhoven University of Technology, September 29, 2009.
4. Giuliano Gadioli La Guardia,” *Convolutional Codes: Techniques of Construction*”, Computational and Applied Mathematics 35(2), March 2013.
5. Sanjay Sharma,” *Communication systems (analog and digital)*”, S.k. Kataria & Sons, 2013.
6. G. David Forney, JR.,” *The Viterbi Algorithm*,” IEEE Proceedings, vol. 61, no. 3, pp. 268-278, March 1973.

Appendix (A): MATLAB functions used in simulation program.

Function symbol	Function Description
Randint(m,n)	Generates random binary numbers of 0 and 1 as a matrix (m, n) or column vector (n=1) or row vector (m=1).
poly2trellis	Convert convolutional code polynomials to trellis description. trellis = poly2trellis(ConstraintLength, CodeGenerator) returns the trellis structure description corresponding to the conversion for a rate K / N feedforward encoder. K is the number of input bit streams to the encoder, and N is the number of output connections. ConstraintLength specifies the delay for the input bit streams to the encoder. CodeGenerator specifies the output connections for the input bit streams to the encoder.
pskmod	Modulate signal using M-PSK method $y = \text{pskmod}(x,M)$ modulates the input signal x using the M-Ary phase shift keying (M-PSK) method. M specifies the modulation order. In binary, $M=2$.
fskmod	fskmod Frequency shift keying modulation $Y = \text{fskmod}(X,M,\text{FREQ_SEP},\text{NSAMP})$ outputs the complex envelope of the modulation of the message signal X using frequency shift keying modulation. M is the alphabet size and must be an integer power of two. The message signal consists of integers between 0 and $M-1$. FREQ_SEP is the desired separation between successive frequencies, in Hz. NSAMP denotes the number of samples per symbol and must be an integer greater than 1.
awgn	awgn Add white Gaussian noise to a signal. $Y = \text{awgn}(X,\text{SNR})$ adds white Gaussian noise to X . The SNR is in dB.
pskdemod	pskdemod Phase shift keying demodulation. $Z = \text{pskdemod}(Y,M)$ demodulates the complex envelope Y of a signal modulated using the phase shift key method. M is the alphabet size and must be an integer power of 2.
fskdemod	fskdemod Frequency shift keying demodulation $Z = \text{fskdemod}(Y,M,\text{FREQ_SEP},\text{NSAMP})$ noncoherently demodulates the complex envelope Y of a signal using the frequency shift keying method. M is the alphabet size and must be an integer power of 2. FREQ_SEP is the frequency separation, and must be positive. NSAMP is the required samples per symbol and must be an integer greater than 1.
vitdec	vitdec Convolutionally decode binary data using the Viterbi algorithm.

	<p>vitdec(CODE,TRELLIS,TBLEN,OPMODE,DECTYPE). CODE is assumed to be the output of a convolutional encoder specified by the MATLAB structure TRELLIS TBLEN is a positive integer scalar that specifies the traceback depth OPMODE denotes the operation mode of the decoder. Choices are: 'trunc' : The encoder is assumed to have started at the all-zeros state. The decoder traces back from the state with the best metric. 'term' : The encoder is assumed to have both started and ended at the all-zeros state. The decoder traces back from the all-zeros state. 'cont' : The encoder is assumed to have started at the all-zeros state. The decoder traces back from the state with the best metric. A delay equal to TBLEN symbols is incurred. DECTYPE denotes how the bits are represented in CODE.</p>
biterr	<p>biterr Compute number of bit errors and bit error rate. [NUMBER,RATIO] = biterr(X,Y) compares the binary representation of the elements in the two matrices X and Y. X and Y must be logical, or integer valued. The number of differences in the binary representation is output in NUMBER. The ratio of NUMBER to the total number of bits used in the binary representation is output in RATIO.</p>
semilogy	<p>semilogy Semi-log scale plot. semilogy(...) is the same as PLOT(...), except a logarithmic (base 10) scale is used for the Y-axis.</p>
ylabel	<p>ylabel Y-axis label. ylabel('text') adds text beside the Y-axis on the current axis.</p>
xlabel	<p>xlabel X-axis label. xlabel('text') adds text beside the X-axis on the current axis.</p>
Title	<p>title Graph title. title('text') adds text at the top of the current axis.</p>
legend	<p>legend Display legend. legend(string1,string2,string3, ...) puts a legend on the current plot using the specified strings as labels.</p>
grid	<p>Grid: Grid lines. grid ON adds major grid lines to the current axes. grid OFF removes major and minor grid lines from the current axes.</p>