Communication Lab II

(ELC3940)

Experiment No.: 07

Object:

(i). Use the bit sequence of Exp. 1 and pass it through an AWGN Channel. Plot the BER curve by increasing the SNR from 1 to 15dB.

(ii). Now apply the channel coding of Exp. 6 and repeat (i).

(iii). Listen the speech signal received in (i) and (ii) for SNR=15 dB and grade it on a scale of 5 with respect to the original signal.

G. No: GL3136

S. No: A3EL-02

F. No: 19ELB056

Name: Maha Zakir Khan

Date of performing experiment: 22 | 03 | 2022

Date of report submission: 29 | 03 | 2022

Software Used:

MATLAB®, Release 2021a (R2021a), a programming platform designed specifically to analyze and design systems and products. The heart of MATLAB is the MATLAB language, a matrix-based language, it provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.

Program:

Using .wav file 'Hello5' to read the speech signal in MATLAB and performed PCM encoding using the function 4-bit PcmEncoder from previous *Experiment-01*

Note: I used a randomly generated binary bit sequence as an alternative to the speech signal encoded values.

```
info = audioinfo('C:\Users\Maha Khan\Downloads\Hello5.wav');
 % Voice signal with dual channel and fs as the sampling frequency
[signal , fs] = audioread('C:\Users\Maha Khan\Downloads\Hello5.wav');
```

Taking PCM sequence as Uni-Polar Scheme

```
 % Function to pcm encode the signal
[msg ,Rb] = PcmEncoder(signal,info);

% Unipolar signaling
A = 1;
msg = msg(1:100000)*A;
N = size(msg);msglen = N(2);
```

Generating AWGN Channel Noise with snr from 1dB to 15dB to be added to the transmitted signal

```
for snr = 1:15

    % Generating random Gaussian noise with given snr as Error
    E = awgn(zeros(1,(msglen/k)*n),snr);

    threshold = A/2; % Setting threshold as the dmin
    for i = 1:(msglen/k)*n
        if E(i) >= threshold
            E(i) = 1;
        else E(i) = 0;
        end
    end

```

Adding Error to the transmitted code signal and decoding on receiver side

```
    for i = 1:(msglen/k)

        % (7,4)linear coding to generate codeword for
        % the given message signal X using custom function hamCoder74
        Cm = hamCoder74(msg(c:c+3));
        C(ce:ce+6) = Cm;

        % Adding Error to the codeword using modulo-addition
        % to generate recieved signal
        r = rem(Cm+E(ce:ce+6),2);
        R(ce:ce+6) = r;

        % Correcting the recieved signal using custom function errCorrect74
        crktCode =  errCorrect74(r);
        Rc(ce:ce+6) = crktCode;

        % Decoding corrected recieved signal from 7 bit codeword to
        % 4 bit message using custom function decode74
        Decodedcrkt = decode74(crktCode);
        D(c:c+3) = Decodedcrkt;


        ce = ce+7;
        c = c+4;
    end

```

Adding Error to uncoded message signal and calculating BER for coded and uncoded signal

```
60      % Received message signal via AWGN channel without channel coding
61 -    R_uncoded = rem(msg+E(1:msglen),2);
62
63      % Bit Error Rate of uncoded received signal
64 -    [~,ber] = biterr(R_uncoded,msg);
65 -    BERuncoded(snr) = ber;
66
67      % BitError Rate of decoded message signal via AWGN channel with
68      % channel coding
69 -    [~,ber] = biterr(D,msg);
70 -    BERcoded(snr) = ber;
71
72 -  end
```

Plot of BER vs SNR for coded and uncoded signal

```
73
74 -    SNR = 1:15;
75 -    snrlin=10.^(SNR./10);
76
77      % BER theoritical of uncoded signal
78 -    BERuncodedTheo=qfunc(sqrt(snrlin/4));
79
80      % BER theoritical of coded signal
81 -    pe = BERuncodedTheo ;
82 -    BERcodedTheo = 1 - nchoosek(4,1)*(pe).*(1-pe).^3 - (1-pe).^4;
83
84 -    semilogy(SNR,BERcoded,'linewidth',2,'marker','>');
85 -    grid on;
86 -    hold on;
87 -    semilogy(SNR,BERuncoded,'linewidth',2,'marker','>');
88 -    hold on;
89 -    semilogy(SNR,BERcodedTheo,'linewidth',2,'linestyle','--');
90 -    hold on;
91 -    semilogy(SNR,BERuncodedTheo,'linewidth',2,'linestyle','--');
92
93 -    xticks([1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]);
94 -    xlabel('SNR in dB ');
95 -    ylabel('BER ');
96 -    title('BER vs SNR Plot of Unipolar Signaling Scheme');
97 -    legend('Coded (Simulated)','Uncoded (Simulated)','Coded (Theoritical)',...
98          'Uncoded (Theoritical)');
99 -    hold off;
```

Comparing sound quality of the coded and uncoded signal

```
    % Decoding Pcm signal to Quantized signal
        ymax = max(signal(:,1));
        ymin = min(signal(:,1));
        codedSig = PcmDecoder(D,ymin,ymax);
        % sound(codedsig,fs);
        uncodedSig = PcmDecoder(R_uncoded,ymin,ymax);
        % sound(uncodedsig,fs);
```

Functions used:

1.) hamCoder74:

```
function [codeWord] = hamCoder74(message)
% Codes the given message signal ingto (7,4) Hamming Code
% Input: 4-bit message  Output: 7-bit codeword

k = 4; % No. of bits in message signal
n = 7; % No. of bits in Codeword



 Parity3 =  [1 1 0;
             0 1 1;
             1 1 1;
             1 0 1];
 Identity4 = eye(k);

 % Generator matrix for generating codeword
 G = [ Parity3, Identity4];

 % Generating codeword for all possible values of message signal
 codeWord = rem(message*G,2);

end
```

2.) errCorrect74:

```
function [correctedcode] = errCorrect74(R)
% To detect and correct error in 7,4 Hamming Code
% Input: 7-bit Erroroneous codeword   Output: Corrected 7-bit Codeword

k = 4; % No. of bits in message signal
n = 7; % No. of bits in Codeword
Parity4 =  [1 0 1 1
            1 1 1 0
            0 1 1 1];
Identity3 = eye(n-k);

    % Check matrix for checking error in the input codeword
    H = [ Identity3, Parity4];
    Syndrome = rem(R*H',2);

  % Generating Syndrome Table
          E = flipud(eye(7,7)); % All possible 1-bit Error matrix
          Rt = rem(zeros(1,7)+E,2);% Adding error to any example codeword
          SyndromeTable = rem(Rt*H',2);

idx = find(ismember(SyndromeTable, Syndrome,'rows'));

    if Syndrome == 0
        correctedcode = R;
    else
    errdetect = E(idx,:);
    correctedcode = rem(errdetect+R,2);
    end

end
```

3.) decode74:

```
function [m] = decode74(Cm)
% Function to decode the (7,4) hamming codeword
% Input: 7-bit Codeword    Output: 4-bit message

k = 4; % No. of bits in message signal
n = 7; % No. of bits in Codeword
 Parity3 =  [1 1 0;
             0 1 1;
             1 1 1;
             1 0 1];
 Identity4 = eye(k);

 % Generator matrix for generating codeword
 G = [ Parity3, Identity4];

 % Matrix containg all possible value of the message signal
 a = dec2bin(0:1:2^k-1)-'0';

 % Generating codeword for all possible values of message signal
 C = rem(a*G,2);

 idx = find(ismember(C, Cm,'rows'));
 m = dec2bin(idx-1,4)-'0';

end
```

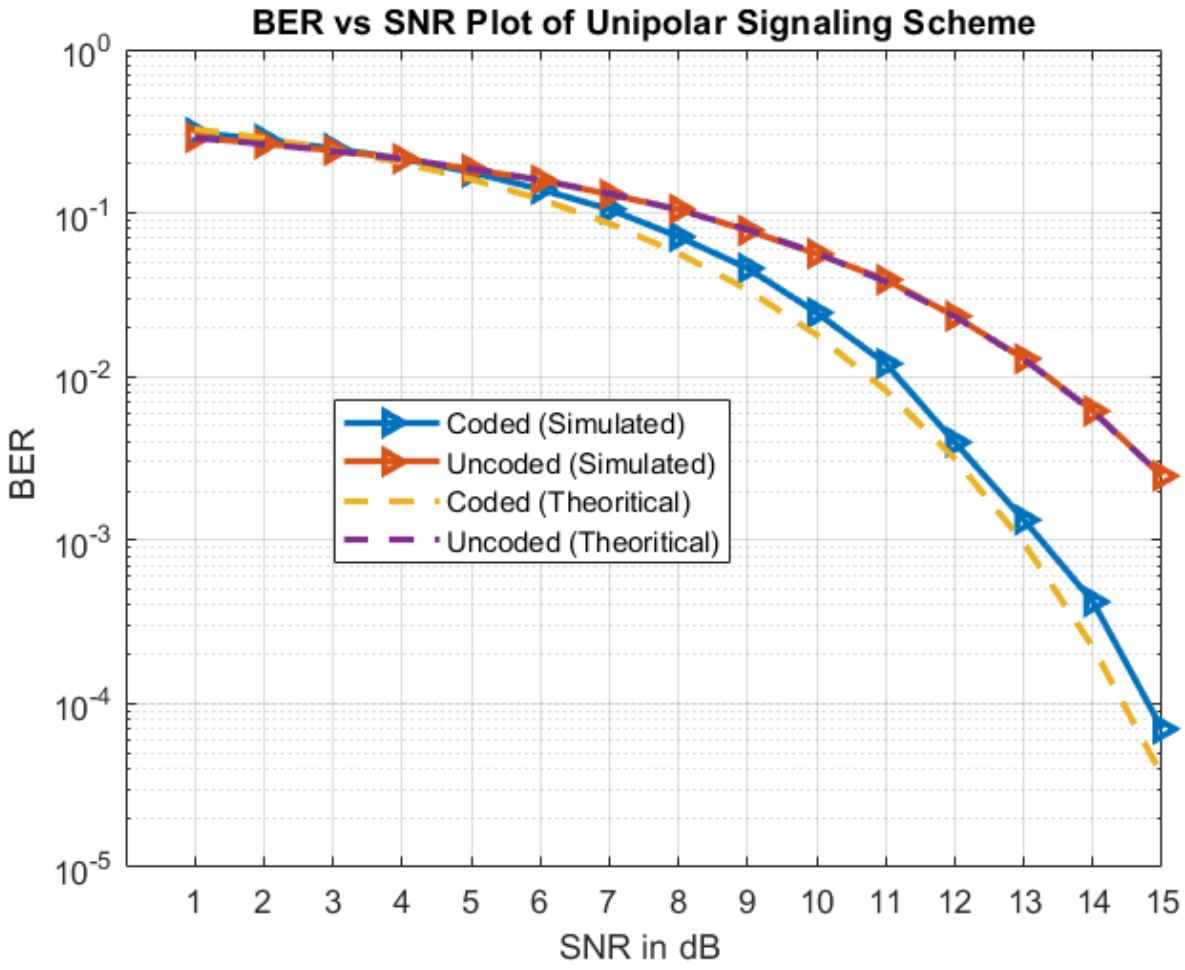
4.) PcmDecoder:

```
function [dsig] = PcmDecoder(pcm,ymin,ymax)
% 4-bit to 16 level Decoder
l = 16;
codebook = linspace(ymin,ymax,l);
decoder =  [0 0 0 0;
            0 0 0 1;
            0 0 1 1;
            0 0 1 0;
            0 1 1 0;
            0 1 1 1;
            0 1 0 1;
            0 1 0 0;
            1 1 0 0;
            1 1 0 1;
            1 1 1 1;
            1 1 1 0;
            1 0 1 0;
            1 0 1 1;
            1 0 0 1;
            1 0 0 0];
rpcm = pcm;
dindex = (linspace(0,0,length(rpcm)/4))';
i = 1;
c = 1;
    while i<length(rpcm)
        for j = 1:length(decoder)
            sub = abs(rpcm(i:i+3)-decoder(j,:));
            if sum(sub) == 0
                dindex(c)= j-1;
            end
        end
        i = i+4;
        c = c+1;
    end
    dsig = (linspace(0,0,length(dindex)))';
    for i = 1:length(dsig)
        dsig(i) = codebook(dindex(i)+1);
    end
end

```

Observation:

Comparison of (7,4) Hamming code with Uncoded Signal:



Discussion:

In this Experiment, I used recorded speech signal for generating uni-polar message bit sequence and introduced AWGN Error with snr varying from 1dB to 15dB. I used only 100,000 of the original

By comparing the plot of the coded (with Channel Coding) and uncoded (without Channel Coding) signal, it was observed that the coded signal BER was lower for higher Snr compared to uncoded signal, this occurred due to the fact that the (7,4) hamming code corrected some of the 1-bit Errors present in the signal resulting in a lower BER value for higher SNR compared to the uncoded signal

For n= 7, k = 4

Hamming distance $d_{min} = n-k = 7-4 = 3$

Error bits that can be detected = $t_d = d_{min} - 1 = 2$

Error bits that can be corrected = $t_c = [(d_{min} -1)/2] = [(3-1)/2] = 1$

The signal quality doesn't change significantly for coded signal on listening and comparing the audio. I'd scale the coded signal as 3.7 and uncoded signal as 3.5 on a scale of 1 to 5 for the 16-level Quantized Decoded Speech Signal