```
% 19ucc023
% Mohit Akhouri
% Observation - Performing the encoding and decoding for a (7,4)
 Hamming
% Code and finding BER for different values of SER

clc;
clear all;
close all;

% This code will perform the encoding and decoding of (7,4) Hamming
 Code
% The Hamming code is given by the generator function G and we will
 use the
% block diagram of encoder and decoder to perform the operations. We
 will
% use two types of decoder - SOFT DECISION decoder and HARD DECISION
% decoder.

% Finally we will find the values of BER for different SER and also
 plot
% the graph between SER and BER for hard and soft decision decoder.

N = 10000; % Size of input sequence

n = 7; % columns of Generating matrix G for Hamming Code
k = 4; % rows of Generating matrix G for Hamming Code

G = zeros(4,7); % Initializing the generating matrix G
G=[1 0 0 0 1 0 1 ; 0 1 0 0 1 1 1 ; 0 0 1 0 1 1 0 ; 0 0 0 1 0 1 1]; %
 Defining the generating matrix G

SER_dB = (0:7); % Defining the array to store SER values in dB from
 0dB to 7dB
SER_array = zeros(1,8); % array to store the SER values converted from
 dB to unitless

% Loop to calculate unitless value of SER
for i=1:size(SER_dB)
    SER_array(i) = 10^(SER_dB(i)/10);
end

size_SER = size(SER_array,2); % to store the number of elements in the
 SER_array

message = zeros(16,4); % to store the Message bits ( 0000 to 1111 )

% Loop to calculate the message in terms of bits and storing them in
% message matrix for further computation
for i=1:16
    str = dec2bin(i-1,4); % To get the binary equivalent of decimal
 number'i-1'
```

```
    temp_array = zeros(1,4); % To store the bits of the binary
 equivalent obtained from above

    % Loop to store the bits in the correct position of temp_array
    for j=1:size(str,2)
        if(str(j)=='0')
            temp_array(j) = 0;
        else
            temp_array(j) = 1;
        end
    end
    message(i,:) = temp_array; % to store the encoded bits in the
 message matrix
end

codeword = mod(message*G,2); % To store the corresponding codewords to
 each bit sequence of message matrix

% Displaying the bit sequence and corresponding codeword
disp('The codewords are as follows :');
disp(sprintf('%-15s \t %-15s','Message','Codeword'));
for i=1:16
    disp(sprintf('%-15s \t
 %-15s',int2str(message(i,:)),int2str(codeword(i,:))));
end

codeword_modf = codeword; % Initializing array to store the modified
 codewords
codeword_modf(codeword_modf==0) = -1; % modify the codewords by
 comparing them to 0 and then assigning -1

INFO_mat = randi([0,1],N,4,size_SER); % to store the information to be
 transmitted through AWGN channel

H = [G(:,k+1:n)',eye(n-k)]; % To store the H matrix obtained by
 transpose of G matrix and multiplying with Identity matrix I

% Loop to calculate the information ( bit sequence ) to be transmitted
for i=1:size_SER
    code_transm(:,:,i) = mod((INFO_mat(:,:,i)*G),2);
end

code_transm_modf = code_transm; % To store the modified transmitted
 codeword
code_transm_modf(code_transm_modf==0) = -1; % Modification of
 transmitted codeword done by comparing to 0 and then assigning -1

% Loop to determine the information ( bit sequence ) received
for i=1:size_SER
    code_recv(:,:,i) = awgn(code_transm_modf(:,:,i),SER_dB(i));
end

hard_dec_decoder = ones(size(code_recv)); % Initializing array to
 store the decoded bit sequence from hard decision decoder
```

```
hard_dec_decoder(code_recv<0) = 0; % Parity checking and modifying the
 hard_dec_decoder array

INFO_hard_dec = zeros(size(INFO_mat)); % To store the received
 information through hard decision decoder
% Main loop algorithm for calculation of bit error rate in case of
 hard
% decision decoder by comparing the distances
for h=1:size_SER
    for i=1:N
        dist = zeros(1,2^k); % to store the distance of codewords

        % loop to calculate the distance of codewords
        for j=1:2^k
            dist(j) = norm(mod(codeword(j,:) +
 hard_dec_decoder(i,:,h),2),1);
        end

        % finding minimum distance index
        [min_elem,ind] = min(dist);
        INFO_hard_dec(i,:,h) = message(ind,:);
    end
    BER_hard_dec(h) = length(find(INFO_hard_dec(:,:,h) -
 INFO_mat(:,:,h)))/(4*N); % Computing the bit error rate for hard
 decision decoder
end

% Displaying the values obtained for the BER for different values of
 SER
% (in dB ) for hard decision decoder
disp('BER value for different values of SER for hard decision decoder
 is as follows :');
disp(sprintf('%-8s \t %-8s','SER (dB)','BER'));
for i=1:8
    disp(sprintf('%-8d \t %-8f',SER_dB(i),BER_hard_dec(i)));
end

INFO_soft_dec = zeros(size(INFO_mat)); % To store the received
 information through soft decision decoder
% Main loop algorithm for calculation of bit error rate in case of
 soft
% decision decoder by comparing the distances
for h=1:size_SER
    for i=1:N
        dist = zeros(1,2^k); % to store the distance of codewords

        % loop to calculate the distance of codewords
        for j=1:2^k
            dist(j) = norm((codeword_modf(j,:) - code_recv(i,:,h)),2);
        end

        % finding minimum distance index
        [min_elem,ind] = min(dist);
        INFO_soft_dec(i,:,h) = message(ind,:);
```

```
    end
    BER_soft_dec(h) = length(find(INFO_soft_dec(:,:,h) -
 INFO_mat(:,:,h)))/(4*N); % Computing the bit error rate for soft
 decision decoder
end

% Displaying the values obtained for the BER for different values of
 SER
% (in dB ) for soft decision decoder
disp('BER value for different values of SER for soft decision decoder
 is as follows :');
disp(sprintf('%-8s \t %-8s','SER (dB)','BER'));
for i=1:8
    disp(sprintf('%-8d \t %-8f',SER_dB(i),BER_soft_dec(i)));
end

% Plotting graph of Bit error rate ( BER ) vs. Symbol error rate
 ( SER )
figure;
semilogy(SER_dB,BER_hard_dec,'red');
hold on;
semilogy(SER_dB,BER_soft_dec,'blue')
xlabel('SER (dB) ->');
ylabel('BER (dB) ->');
title('19ucc023 - Mohit Akhouri','Plots of Bit Error Rate ( BER )
 vs. Symbol Error Rate ( SER in dB ) for HARD and SOFT decision
 decoders');
legend('Hard Decision Decoder','Soft Decision Decoder');
grid on;
hold off;
```

Published with MATLAB® R2020b