Simulation Of Bit Error Rate Of QPSK Modulation System With And Without Coding

ABSTRACT

In this project, the performance of the Quadrature Phase Shift Keying (QPSK) modulation technique is observed by simulating the bit error rate (BER) and comparing it with theoretical BER for various values of Eb/N0 over Additive White Gaussian Noise (AWGN) channel. The simulation is done by using MATLAB software. Firstly the simulation is done without any coding and then to improve the BER, single error correcting hamming code (15, 11) is used as channel coding.

INTRODUCTION

QPSK modulation technique is one of the most popular digital modulation techniques, because of its easy implementation and resistance to noise. In this method, digital information is transmitted across analog channel. A series of binary digits which has to be transmitted are grouped into symbols, where each symbol consists of two bits (00, 01, 10, 11). The carrier varies in terms of phase and especially in four possible phase shifts (450, 1350, 2250, 3150). Then the bits of each symbol are modulated to in phase and quadrature bits as shown in the constellation diagram.

The QPSK transmitter system uses both the sine and cosine at the carrier frequency to transmit two separate message signals as in-phase I and quadrature Q signals. Thus a coherent receiver system is employed such that both the in-phase and quadrature signals can be recovered exactly.

EB/N0 Vs BER WITHOUT ANY CODING

In this section the performance of the QPSK modulation scheme is observed by plotting the theoretical and simulated BER over different values of Eb/N0, and the channel used here is AWGN.

Binary message is transmitted using QPSK modulation technique over AWGN channel.

In QPSK modulation the channel can be modeled as Rx= Tx+N, where Rx is the received signal, Tx is the transmitted signal and N is the Additive Gaussian White Noise random variable with zero mean and variance of ?2. The noise term is generated by using “randn” function in matlab, which gives noise with unit variance and zero mean. Then the sigma (?) is multiplied with randn function to generate noise with standard deviation. Since the transmitted signal is in complex form, the generated noise signal should also be in complex form.

Then the received signal is generated simply by adding transmitted signal and noise signal. Then this received message is compared with transmitted message to calculate simulated BER. Then the simulated BER is compared with theoretical BER which is equal to

Pb=0.5*erfc(?Eb/N0) for different values of Eb/N0.

Then the theoretical and simulated BER is plotted over different values of Eb/N0. The performance of the QPSK modulation system over AWGN can be concluded as, when the Eb/N0 value is high the Bit Error rate of the system is high.

4. EB/N0 Vs BER USING CHANNEL CODING

In order to improve the performance of the QPSK modulation system when compared to previous one, channel coding is used. The channel coding here used is (15,11) hamming code. Hamming code is the linear error correcting code which can detect and correct errors. Hamming code can detect up to two errors and can correct up to one error. After adding hamming (15,11) code in QPSK system the symbol error probability can be represented as

Where Echo/No is the coded symbol energy and it can be written as

Ec/No =(k/n)(Eb/N0); (n,k) = (15,11)

The bit error probability cab be find using the following equation,

Where t=1 and n=15

The message m, which has sequence of n bits is multiplied with G matrix (G = [P, I11)]) to generate code words. Now these code words are transmitted instead of original message bits. A parity check matrix called H matrix (H = [I4, PT]) is generated to decode the received bits. Then the two syndrome matrices are generated by multiplying HT with received matrix (S=rHT) and error pattern (S=eHT). Then these two syndrome matrices are compared two each other. Then corrected code word can be found by adding r and e. Then finally the parity bits has to drop to get estimated received message. The received message is compared with original message to calculate simulated BER. This simulated BER is compared with theoretical BER (Pb).

CONCLUSION

The simulation of bit error rate of QPSK system with and without hamming code is done using MATLAB and the results are observed by plotting theoretical and simulated BER Vs Eb/N0.

In both the cases the bit error rate of the system is becoming better with increase of Eb/N0 value. But, in the case of without coding the BER of 10-5 is achieving at the value of Eb/N0=9.5 db, while in the case of coding the BER of 10-5 is achieved at the value of Eb/N0=8.2 by having a coding gain of 1.3 db. [Has to include TA reason]

APPENDIX A

% MATLAB SIMULATION FOR BER Vs Eb/N0 of QPSK system over AWGN

clc,clear;

n=1000000;% number of bits in transmitted message

```
ebn0db = 0:1:10;

trans_data = randi([0,1],1,n); % Genaration of n random binary digits

Ibits = trans_data(1:2:end); % Inphase component

Qbits = trans_data(2:2:end); % Quadrature component

transmited_signal= sqrt(1/2)* ((2*Ibits-1)+1i*(2*Qbits-1)); % QPSK modulated signal

BER = zeros(1,length(ebn0db)); % allocating space for BER

x=1;

E=1;

for i= 1:length(ebn0db)

snr=10

(ebn0db(i)/10);

sgma=sqrt(E/(2*2*snr));

noise=sgma*(randn(1,length(transmited_signal))+1i*randn(1,length(transmited_signal)));
%generating noise signal

received_signal=transmited_signal+noise;

received_firstbit=real(received_signal)>=0;

% threshold detector to detect real part

received_secondbit=imag(received_signal)>=0;

% threshold detector to detect imaginary part

received_data=reshape([received_firstbit;received_secondbit],1,[]);

% Genarting sequence of bits

BER(x)=sum(xor(trans_data,received_data))/length(trans_data); % simulated Bit error rate

BER_theory(x)=(1/2)*erfc(sqrt(snr));

% theoritical bit erroir rate

x=x+1;
```

```
end
semilogy(ebn0db,BER_theory,'b*-','LineWidth',2);
% plotting theoritical ber vs ebn0db
hold on
semilogy(ebn0db,BER,'ro--','LineWidth',2);
% plotting simulated ber vs ebn0db
grid on;
axis([0 max(ebn0db) 10
-5 1]); % specifying the axis range
legend('Theoretical','Simulation');
xlabel('Eb/N0 (dB)');
ylabel('BER');
title('BER Vs Eb/N0 QPSK');
```

APPENDIX B

```
%MATLAB SIMULATION FOR BER Vs Eb/N0 of QPSK system with (15,11) hamming code
clc,clear;
n = 990000; % number of bits in transmitted message
ebn0db = 0:1:10;
E=1;
P = [1 1 1 1 ;
0 1 1 1 ;
1 0 1 1 ;
1 1 0 1 ;
1 1 1 0 ;
0 0 1 1 ;
```

```
0 1 0 1 ;

0 1 1 0 ;

1 0 1 0 ;

1 0 0 1 ;

1 1 0 0 ]; % parity bits

G = [P,eye(11)]; % Genarator matrix

H = [eye(4),P']; % parity check matrix

Error=vertcat(zeros(1,15),eye(15)); % table of error

Syndrome = Error*H'; % table of syndrome

for i=1:length(ebn0db)

snr=10

(ebn0db(i)/10); %d to linear value

sgma=sqrt((E/(2*2*snr))*15/11); % standard deviation

bit_error=0;

trans_data = randi([0,1],1,n); % generation n binary bits

m=reshape(trans_data,11,n/11).'; %

code_word=reshape(mod(m*G,2).',1,(n*15)/11); % Generating code word

Ibits=code_word(1:2:end); % in-phase component

Qbits=code_word(2:2:end); % Quadrature component

transmited_signal = sqrt(1/2)*((2*Ibits-1)+1i*(2*Qbits-1)); %Qpsk modulated signal

k=length(transmited_signal);

noise = sgma*(randn(1,k)+1i*randn(1,k)); %noise signal

received_signal = transmited_signal+ noise;% received signal

received_oddbit = real(received_signal)>=0; % threshold detector to detect real part

received_evenbit = imag(received_signal)>=0; % threshold detector to detect imaginary part
```

```
received_data = reshape([received_oddbit; received_evenbit],1,[]); % received data
m=length(code_word);
estimate_code_word=reshape(received_data,[15,m/15]).';
% Syndrome Decoding
s=mod(estimate_code_word*H',2);
% Error Detection and Correction
for row=1:length(code_word)/15
for counter=1:16
if s(row,:)== Syndrome (counter,:)
corrected_word(row,:)=xor(estimate_code_word (row,:),Error(counter,:));
end
end
end
corrected_msg=corrected_word(:,5:15);
corrected_vector=reshape(corrected_msg',1,[]);
% calculation of simulated BER
bit_error=sum(xor(corrected_vector,trans_data));
simulated_ber(i)=bit_error/(n);
% calculation of theoritical BER
Pc=(1/2)*erfc(sqrt(snr*(11/15)));
x=0;
for a=2:15
x=(a*nchoosek(15,a)*(Pc
a)*((1-Pc)
(15-a)))+x;
```

```
theoretical_ber(i)=(1/15)*x;
end
end
% to plot BER Vs Eb/N0 curve
close all
semilogy(ebn0db,theoretical_ber,'bs-','LineWidth',2);
hold on
semilogy(ebn0db,simulated_ber,'ko--','LineWidth',2);
grid on;
axis([0 max(ebn0db) 10
-5 1]);
legend('Theory', 'Simulation');
xlabel('Eb/N0 (db)');
ylabel('BER');
title('BER Vs Eb/N0 ,QPSK with (15,11) Hamming Code').
```