Labratory 5
ENGR 461: Digital Communications

Graeme Paul

June 20, 2020

1 Introduction

In this lab a image is being transmitted within MATLAB. The image is transmitted using 4-PAM modulation. Multiple MATLAB functions were used to implement the transmission. 4-PAM modulation is a method of modulation which uses the amplitude during a pulse to send information. 4-PAM has four different levels of amplitude and can encode two bits per level. The image that is transferred is a 128x128 image. In this lab the transmission is sent through a additive white Gaussian noise channel. The noisy channel can cause error depending on the SNR. The Qfunc is estimated in the lab by finding the BER of the transmission.

2 Image Transmission

2.0.1 Modulation

Pulse Amplitude modulation is used to send a signal which contains information through amplitudes sent as pulses. 4-PAM uses four levels of amplitudes which are equally spaced. The amplitudes result in signals which are equally spaced by a distance of d_{min} across the In-Phase axis of the constellation diagram. The distance is dependant on the amplitude of the pulse which is proportional to the energy. The energy of the signal in this lab is set by selecting a desired signal to noise ratio. In this lab SNR is defined as $\frac{\varepsilon_b}{N_o}$.

The desired SNR is first selected then equation 1 can be used to determine ε_b which is the energy per bit and is entirely dependant on the SNR and Noise N_o. The energy per bit then gives average signal energy which can be found with equation 3. The average signal energy then gives d using equation 4. Using equation 5 amplitudes for the modulator can be found from distance. From d the minimum distance between signals can be found to be $d_{min} = 2d$.

$$\varepsilon_b = N_0 \times 10^{\frac{\mathrm{SNR}}{10}} \tag{1}$$

$$\mathrm{SNR} = 10\log_{10}(\frac{\varepsilon_b}{N_0}) \tag{2}$$

$$\varepsilon_s = \varepsilon_b \log_2(M) \tag{3}$$

$$d = \sqrt{\frac{3\varepsilon_s}{M^2-1}} \tag{4}$$

$$A_m = (2m-1-M)d, m = 1, 2, \ldots, M \tag{5}$$

In MATLAB the modulator was implemented as a function using the amplitudes found from equation 5 and mapped according to grey coding. Grey coding encodes the bits so that only one bit changes between adjacent levels. The modulator function takes two bits as inputs then uses the grey coding mapping to decide which of the four amplitudes to send the bits as. The encoding scheme is shown in table 1. The constellation diagram for the encoding scheme can be seen in figure 1 below where the circles are the modulated signals with a unit energy, $\varepsilon_s = 1$.

Table 1: Grey Mapped Modulated Signal Table

Bits	Modulated Amplitude of Signal as Distance
00	-3d
01	-d
11	d
10	3d

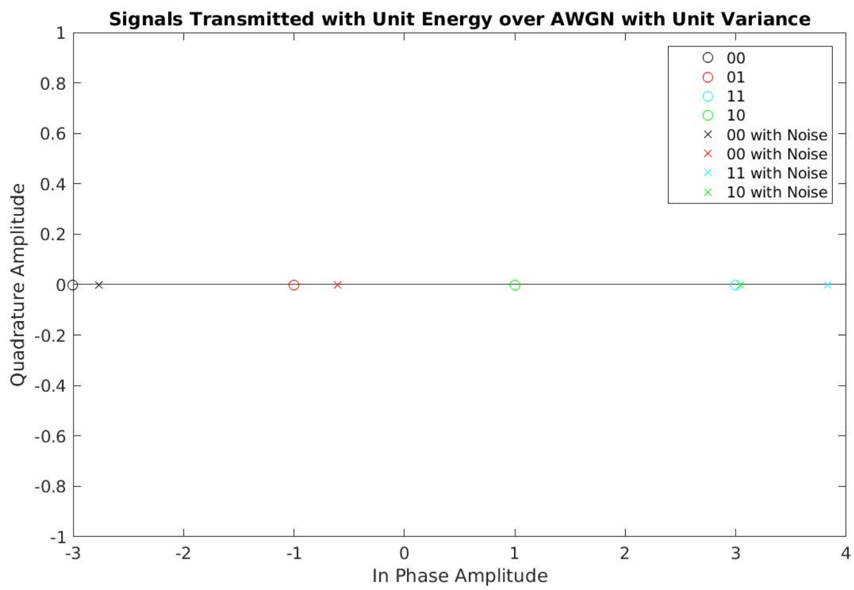

Figure 1: Signal Constellation for 4-PAM modulation

2.1 Channel Noise

In this lab the image is transmitted over a noisy channel. The noise of the channel is determined by equation 6 which is the temperature times the Boltzmann constant. This models thermal noise that would exist in a channel. With this value determined the signal to noise ratio can be determined using equation 2. When the modulated signal passes through the channel the noise is added to the modulated signal. The channel is a additive white Gaussian noise channel so the noise added to the signal and is zero mean Gaussian noise. The noise will cause the signal to shift left or right on the signal constellations. The noise + signal are plotted on the signal constellation in figure 1 as an X, in this plot the noise is zero mean unit variance causing the signal to move left or right.

$$N_0 = k_b T = 1.38 \times 10^{-23} \frac{J}{K} \times 300K = 4.1400 \times 10^{-21} J \tag{6}$$

2.1.1 Demodulation

After passing through the channel demodulation occurs. Demodulation takes the signal values and uses them to retrieve the original information. The demodulator uses a decision region which is optimal for equiprobable signal transmission, shown in the table below. These values were found from equation 4 knowing that the distance between two signals is $d_{min} = 2d$.

Table 2: Grey Mapped Modulated Signal Table and Decision Region

Bits	Modulated Amplitude of Signal	Decision Region
00	-3d	$y \leq -2d$
01	-d	$-2d < y \leq 0$
11	d	$0 < y \leq 2d$
10	3d	$y \geq 2d$

2.2 Bit Error Rate

The bit error rate can be calculated using equation 4, and the symbol error rate can be calculated using equation 5. Both the symbol error rate and the bit error rate were used to analyze the performance of the codes at various SNR.

$$\text{Bit Error Rate} = \frac{\text{Number of Bit Errors}}{\text{Number of Bits Transmitted}} \tag{7}$$

$$\text{Symbol Error Rate} = \frac{\text{Number of Symbol Errors}}{\text{Number of Symbols Transmitted}} \tag{8}$$

$$\text{SER}_{\text{4PAM}}^{\text{AWGN}} = 2(1 - \frac{1}{M})Q(\frac{d}{\sqrt{No/2}}); \tag{9}$$

3 Results

In the first part of the lab 4PAM was used to modulate a signal with a unit signal energy $\varepsilon_s = 1$. Using equation 4 the required distance can be found. This was then added to noise with zero mean and unit variance. The result is seen in figure 2, which is a copy of figure 1. The signals without noise are circles and the signals with the added noise are shown as an X.

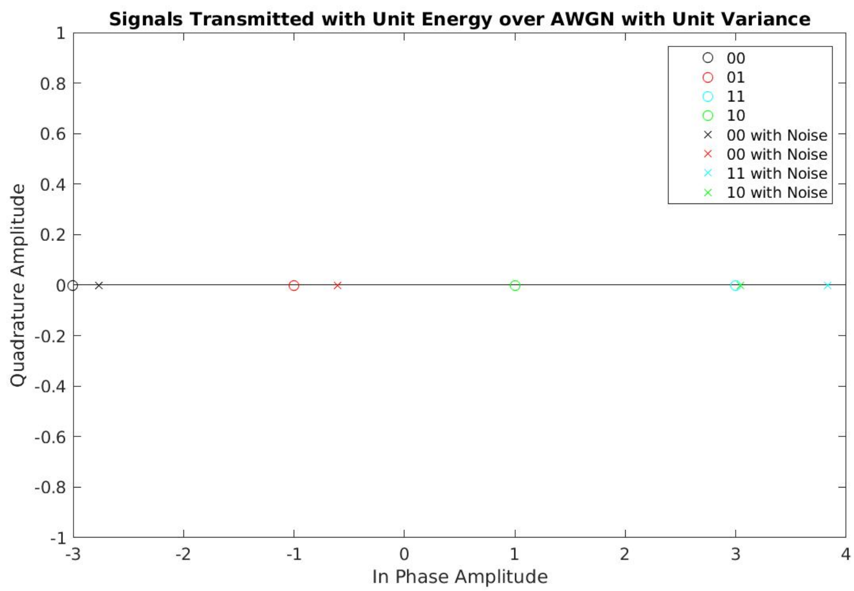

Figure 2: Signal Constellation for 4-PAM modulation

The image lena128.jpg was then sent using 4PAM transmission at various bit energy SNRs (SNR = $\frac{\varepsilon_b}{N_o}$ = 0:1:10). This was done first by using equation 1 to find ε_b for a given SNR then using equation 3 and 4 to find d, which is then used to give amplitudes with equation 5.

It can be seen from the table below that the when the SNR ($\frac{\varepsilon_b}{N_o}$) is increased the number of errors decreases. This characteristic can also be seen in figure 2 where 4-PAM estimated BER and 2-PAM theoretical and estimated BER are plotted. The graph shows the decreasing nature of BER as SNR is increased. When compared to 2-PAM it can be seen that 4-PAM has greater number of errors. Therefore at equal bit energies 2-PAM performs better than 4-PAM for signal integrity. This is due to the minimum distance of 4-PAM which is less than that of 2-PAM for equal bit energies which can be seen in equation 13 and equation 14. This causes the decision region to reduce and therefore noise is more likely to cause bit errors. In addition signals s_2 and s_3 have smaller decision regions as they are both adjacent to other signals, where as in 2-PAM both signals have decision regions which go from 0 to $+$/-∞. In addition the BER could be further increased by large spikes of noise or with bad encoding as there is also a possibility that two bit errors occur per symbol error which can also increase the number of bit errors. These three effects add up together causing 4-PAM to have more errors than 2-PAM at equal SNR ($\frac{\varepsilon_b}{N_o}$).

Using Q-func the SER for 4-PAM can be calculated using equation 11. It is plotted along with the estimated SER for the transmitted image in figure 4 and can be seen to be very close. The percent error between the two can be seen in figure 5, which shows that the two are very close. This shows that estimating the SER for 4PAM by counting the number of symbol errors is a good estimate for Qfunc.

The reconstructed images are shown in figure 6 and figure 7. It can be seen visually that 2-PAM performs better than 4-PAM at the same SNR ($\frac{\varepsilon_b}{N_o}$). These images also show how the BER reduces as the SNR increases.

If a BER of 10^{-2} was required to transmit the image then different SNRs ($\frac{\varepsilon_b}{N_o}$) would need to be used for 4-PAM and 2-PAM. Where $-20dB = 10\log(10^{-2})$, the graph can be used to find the SNR that is needed. It can be seen that 2-PAM line crosses -20dB line at about 4.2SNR($\frac{\varepsilon_b}{N_o}$) and the 4-PAM line crosses -20dB at around 7.8dBSNR($\frac{\varepsilon_b}{N_o}$). The table also aligns with these estimates showing for 2-PAM it is between 4 and 5 SNR and for 4-PAM it is between 7 and 8 SNR. Therefore this test shows that 4-PAM requires more energy per bit to transmit at the same error rate as 2-PAM.

$$\text{BER}_{\text{2PAM}}^{\text{AWGN}} = Q(\sqrt{\frac{\varepsilon_b}{N_0}}) \tag{10}$$

$$\text{SER}_{\text{4PAM}}^{\text{AWGN}} = (1.5)Q(\frac{d}{\sqrt{No/2}}); \tag{11}$$

$$d_{min,MPAM} = 2 * d = 2(\sqrt{3\frac{\varepsilon_s}{M^2-1}}) \tag{12}$$

$$d_{min,2PAM} = 2 * d = 2\sqrt{\varepsilon_b} \tag{13}$$

$$d_{min,4PAM} = 2 * d = 2\sqrt{\frac{2\varepsilon_b}{5}} \tag{14}$$

Table 3: Estimated BER for 2-PAM and 4-PAM

SNR ($\frac{\varepsilon_b}{N_o}$)dB	BER4	BER2	Errors 4PAM	Error 2PAM
0	0.142924	7.88E-02	56200	30985
1	0.120425	5.59E-02	47353	21994
2	0.098946	3.73E-02	38906	14670
3	0.078557	2.27E-02	30890	8909
4	0.059682	1.23E-02	23468	4845
5	0.042699	6.00E-03	16790	2358
6	0.028142	2.34E-03	11066	922
7	0.017293	7.22E-04	6800	284
8	0.009196	1.83E-04	3615	72
9	0.004407	2.80E-05	1733	10
10	0.001775	1.02E-05	697	4

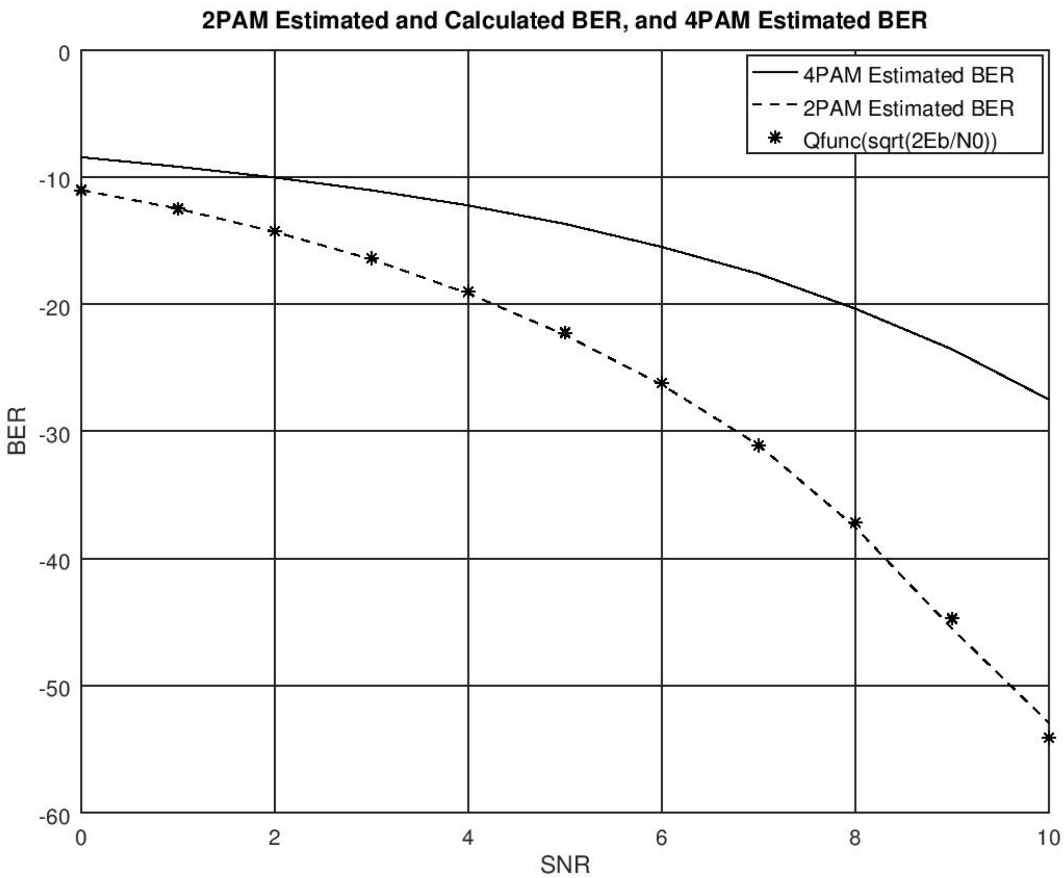

Figure 3: 2-PAM Calculated BER and Estimated and 4-PAM Estimated BER, with Grid

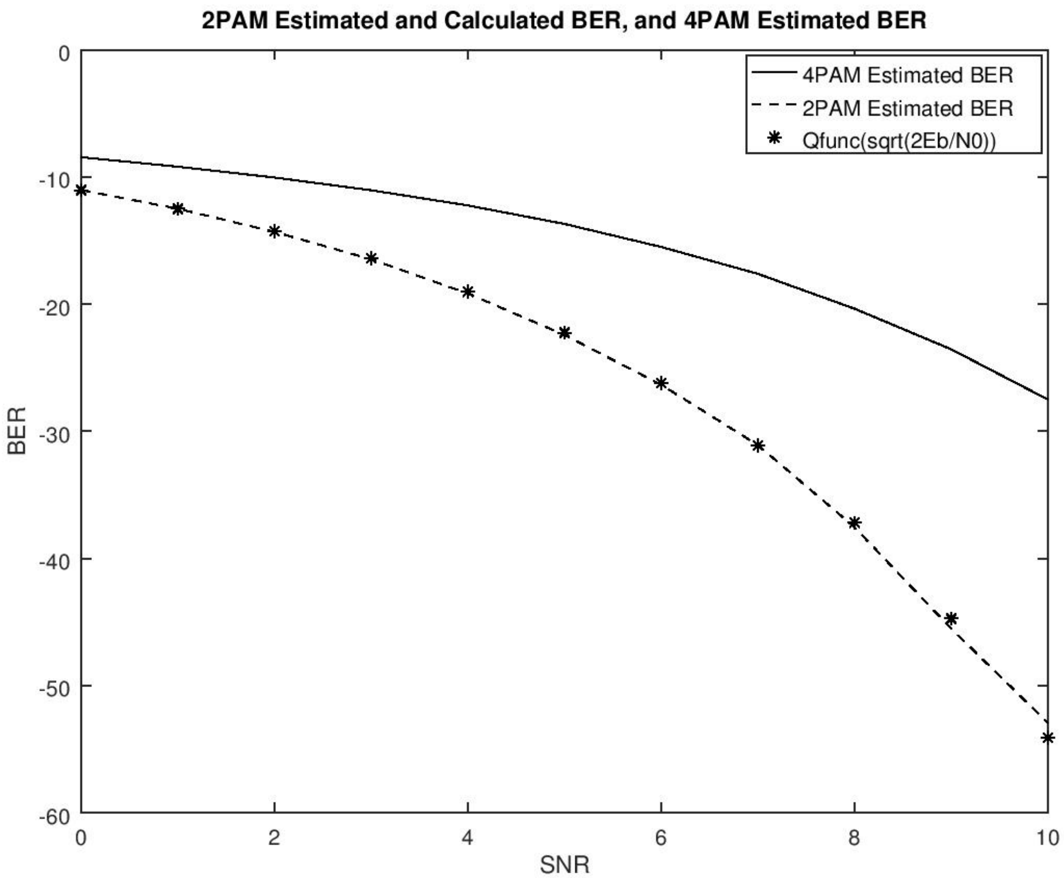

Figure 4: 2-PAM Calculated BER and Estimated and 4-PAM Estimated BER, without Grid

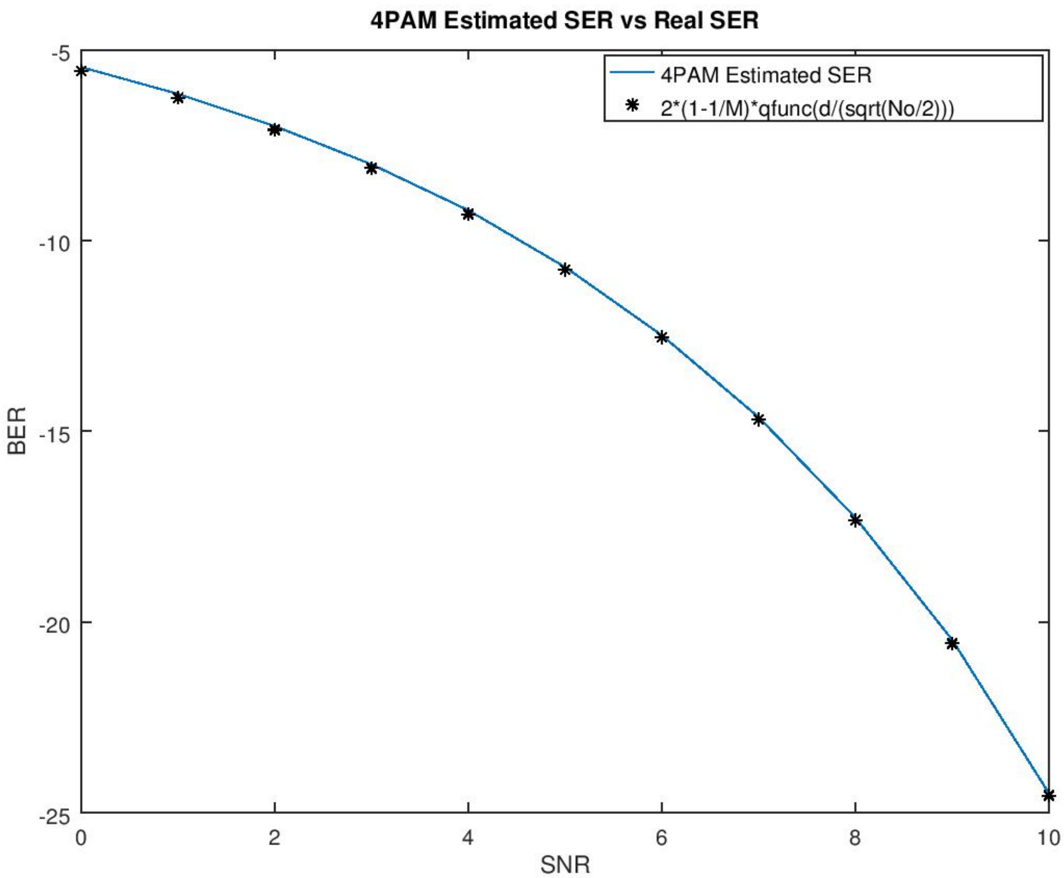

Figure 5: 4-PAM Estimated SER and 4-PAM Calculated SER

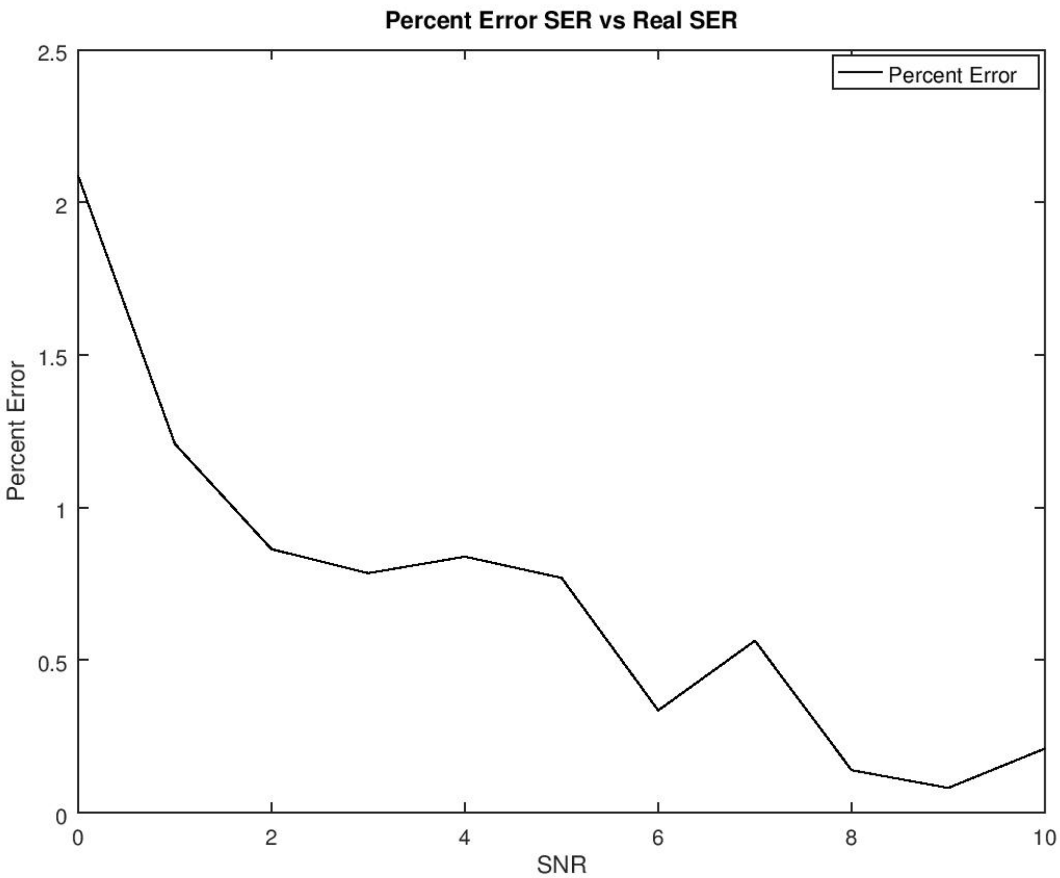

Figure 6: 4-PAM Estimated SER Error

Figure 7: Received and Transmitted Images of SNR 5 channel

Figure 8: Received and Transmitted Images of SNR 10 channel

4 Conclusion

In this lab a image was transmitted in MATLAB using 4-PAM. The image was transmitted over a additive white Gaussian noise channel with various SNRs. Two functions were created to modulate and demodulate the image using 4-PAM. Grey coding was used to encode the bits to four different amplitudes. To demodulate a decision region for equiprobable signal transmission was used and amplitudes were put back to bits. The bit error rate was estimated by finding it across the SNR range of $\frac{\varepsilon_b}{N_o}$ = 0:1:10 and compared to the bit error rate for 2-PAM found in lab 4. It was found that for the same energy per bit 4-PAM has a higher bit error rate than 2-PAM, which was mostly due to the decreased decision regions that 4-PAM has. From the reconstructed images and the graphs it can be seen that the BER goes down when SNR increases. The estimated SER for 4-PAM was compared to the theoretical from using Qfunc and it was shown that they match. Lastly the BER of 10^{-2} was desired and the SNR was found for 4PAM and 2PAM to transmit at this BER. This again showed that to transmit at the same BER 4PAM requires a larger SNR than 2PAM. Overall this lab showed the bit error performance of 4-PAM compared to 2-PAM.

5 Appendix

5.1 MATLAB Code

5.2 Modulation

```
function result = FourPAMmodem(x,d)%innput is array
    result = zeros(length(x),1);
    for i = 1:(length(x))
        if(x(i,:) == '00')
            result(i) = -3*d;%(2*1-1-4)*1; %-3d
        elseif(x(i,:)== '01')
            result(i) = -1*d;%(2*2-1-4)*1; %-1d
        elseif(x(i,:)== '11')
            result(i) = 1*d;%(2*3-1-4)*1; %1d
        else %x = "11"
            result(i) = 3*d;%(2*4-1-4)*1; %3d
        end
    end
end
```

5.3 Demodulation

```
function result = FourPAMdemodem(y,d)
result = zeros(length(y), 2);
result = char(result);
    for i = 1:(length(y))
        if(y(i) <= (-2*d))
            result(i,:) = "00";
        elseif((y(i) > (-2*d)) && (y(i) < 0))
            result(i,:) = "01";
        elseif((y(i) < 2*d) && (y(i) > 0))
            result(i,:) = "11";
        elseif(y(i) >= 2*d)
            result(i,:) = "10";
        end
    end
end
```

5.4 Lab code

```
%lab5 461
clc
%clear all
close all

%NxN image and RGB (3 colours)
N = 128;
RGB = 3;
k  = 1.38e-23
T = 300;
snr = 10;
err_arr = [];
err_arr2 = [];

qfun = zeros(1,15);
M=4;
No = k*T;



%N,N,R array image
```

```
image = imread('lena128.jpg');
%makes single col array with 8 bit rows (strings)
image_bin = dec2bin(image);
%single bit col array (string)
%image_bin_rehape = reshape(image_bin,[],1);
sizeofimage = size(image_bin);
c =  zeros(sizeofimage(1) * sizeofimage(2)/2,2);
for i = 1:sizeofimage(1)
  for j = 1: sizeofimage(2)/2
     c(4*(i-1)+j,:) =  image_bin(i,((j-1)*2+1:j*2));
  end
end
%2PAM
image_bin_rehape2 = reshape(image_bin,[],1);
image_num = str2num(image_bin_rehape2);
%4PAM
image_bin_rehape = char(c);
noisy2 = normrnd(0,sqrt(No/2),length(image_bin_rehape2),1);
noisy = noisy2(1:length(image_bin_rehape));


for z = 1:11

    snr = z-1
    eb=No * (10^(snr/10));
    es = eb*log2(M);
    d=sqrt((es*3)/((M^2)-1));

    snr_check = 10*log10(eb/No)
    %4pam
    scaled_image_pam = FourPAMmodem(image_bin_rehape,d);
    img_pam_err=scaled_image_pam+noisy;
    %2pam
    scaled_image_pam2 = pammod(image_num,2)*sqrt(eb);
    img_pam_err2=scaled_image_pam2'+noisy2;
    %
    %4
    img_pam_demod = FourPAMdemodem(img_pam_err,d);
    err_cnt = 0;
    for i=1:length(c)
      %for j = 1:2
          if(img_pam_demod(i,1) ~= image_bin_rehape(i,1))
          err_cnt = err_cnt + 1;
          end
         if(img_pam_demod(i,2) ~= image_bin_rehape(i,2))
          err_cnt = err_cnt + 1;
        end
      %end
    end
    %
    %2
    img_pam_demod2 = pamdemod(img_pam_err2,2);
    err_cnt2 = 0;
    for i=1:(length(img_pam_demod2))
      if(img_pam_demod2(i) ~= image_num(i))
          err_cnt2 = err_cnt2 + 1;
      end
    end
    if(z == 6 || z == 11)
      %4
      demod_image_bin = num2str(img_pam_demod);
      for i=1:(N*N*RGB)
          demod_image_reshape(i,:)='00000000';
      end
      for i = 1:sizeofimage(1)
        for j = 1:sizeofimage(2)/2
           demod_image_reshape(i,((j-1)*2+1:j*2)) = img_pam_demod(4*(i-1)+j,:);
        end
```

```
     end

      demod_image= bin2dec(demod_image_reshape);
      demod_image_arr = zeros(N,N,RGB,'uint8');
      for k = 1:RGB
          for j = 1:N
              for i = 1:N
                  demod_image_arr(i,j,k)=demod_image((i+((j-1)*N))+((k-1)*N*N));
              end
          end
      end
    end
    %2
    demod_image_bin2 = num2str(img_pam_demod2);
    for i=1:(N*N*RGB)
        demod_image_reshape2(i,:)='00000000';
    end
    for j = 0:(log2(N))
        for r = 1:(N*N*RGB)
            demod_image_reshape2(r,(j+1))=demod_image_bin2(r+(j*(N*N*RGB)));
        end
    end
    demod_image2= bin2dec(demod_image_reshape2);
    demod_image_arr2 = zeros(N,N,RGB,'uint8');
    for k = 1:RGB
        for j = 1:N
            for i = 1:N
                demod_image_arr2(i,j,k)=demod_image2((i+((j-1)*N))+((k-1)*N*N));
            end
        end
    end
    if(z == 6)
        figure
        subplot(3,1,1), imshow(image);
        title('SNR 5: Transmitted  Image 4PAM');
        subplot(3,1,3), imshow(demod_image_arr);
        title('SNR 5: Received Image 4PAM');
        subplot(3,1,2), imshow(demod_image_arr2);
        title('SNR 5: Received Image 2PAM');
        %subplot(3,1,1), stem(img_pam); ylim([-2 2]); title('Modulated Signal at 5 SNR');
        %subplot(3,1,2), plot(noisy); title('Noisy Signal at 5 SNR');
        %subplot(3,1,3), stem(img_pam_demod); title('Demodulated Signal at 5 SNR');
    end
    if(z == 11)
        figure
        subplot(3,1,1), imshow(image);
        title('SNR 10: Transmitted  Image 4PAM');
        subplot(3,1,3), imshow(demod_image_arr);
        title('SNR 10: Received Image 4PAM');
        subplot(3,1,2), imshow(demod_image_arr2);
        title('SNR 10: Received Image 2PAM');
        %figure
        %subplot(3,1,1), stem(img_pam);  ylim([-2 2]); title('Modulated Signal at 10 SNR');
        %subplot(3,1,2), plot(noisy);  title('Noisy Signal at 10 SNR');
        %subplot(3,1,3), stem(img_pam_demod); title('Demodulated Signal at 10 SNR');
    end
    err_cnt;
    err_cnt = err_cnt/(128*128*3*4);
    err_arr = [err_arr err_cnt]

    err_cnt2;
    err_cnt2 = err_cnt2/(128*128*3*8)
    err_arr2 = [err_arr2 err_cnt2]
end

qfun = zeros(1,11);
for l = 1:11
    snr = l-1;
```

```
    eb=No * (10^(snr/10));
    es = eb*log2(M);
    d=sqrt((es*3)/((M^2)-1));
      qfun(l) =2*(1-1/M)*qfunc(d/(sqrt(No/2)));
end
for l = 1:11
    snr = l-1;
    No = k*T;
    eb= No*(10^(snr/10));
    qfun2(l) = qfunc(sqrt(2*eb/(No)));
end
snr = 0:10;
figure
plot(snr,10*log10(err_arr));
hold on
plot(snr,10 * log10(qfun),'*k');
title('4PAM Estimated SER vs Real SER');
legend('4PAM Estimated SER','2*(1-1/M)*qfunc(d/(sqrt(No/2)))');
ylabel('BER'); xlabel('SNR')
hold off
figure
plot(snr,10*log10(err_arr2));
hold on
plot(snr,10 * log10(qfun2),'*k');
title('2PAM Estimated BER vs Real BER');
legend('2PAM Estimated BER','Qfunc(sqrt(2Eb/N0))');
ylabel('BER'); xlabel('SNR')
hold off
figure
plot(snr,10*log10(err_arr/2),'k');
hold on
plot(snr,10*log10(err_arr2),'--k');
plot(snr,10 * log10(qfun2),'*k');
title('2PAM Estimated and Calculated BER, and 4PAM Estimated BER');
legend('4PAM Estimated BER','2PAM Estimated BER','Qfunc(sqrt(2Eb/N0))');
ylabel('BER'); xlabel('SNR')
hold off
figure
plot(snr,abs(((10 * log10(qfun(1:11))) - 10*log10(err_arr(1:11)))./(10 * log10(qfun(1:11))))
    *100,'k');
title('Percent Error SER vs Real SER');
legend('Percent Error');
ylabel('Percent Error'); xlabel('SNR')
```