****

**哈尔滨工业大学（深圳）**

**可见光通信实验报告**

**实验三&四：DCO-OFDM**

姓 名 陈兴基

专 业 通信工程

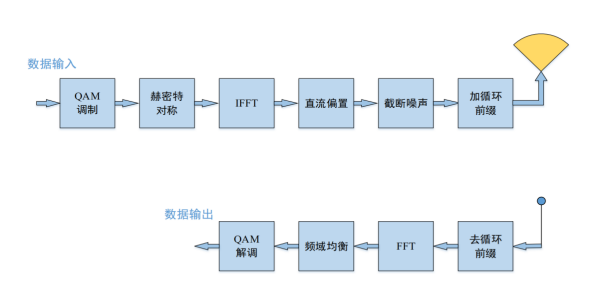
学 号 190210303

日 期 2022.5.13

**实验三&四**

## 一、实验原理

在强度调制/直接检测LED可见光通信系统中，发送信号必须为实正信号。为了满足该要求，DCO-OFDM比普通OFDM多了两个关键步骤：赫密特对称 (Hermitian Symmetry)和直流偏置(DC Bias)。其中，赫密特对称用于产生实信号，直流偏置用于产生正信号。DCO-OFDM系统框图如图所示。



第一步、QAM 调制：在DCO-OFDM系统中，一个DCO-OFDM块包含N个子载波。待发送的QAM信号在第个子载波表示为。一个 DCO-OFDM 块为一个N ×1向量，表示为。

第二步、赫密特对称：QAM信号经过赫米特对称后表示为



第三步、经过快速傅里叶反变换（IFFT）后变为时域信号：



第四步、直流偏置：为了让大部分信号变为正实数，加入一个直流偏置在时域信号上：



其中，为直流偏置量。

第五步、截断：截取剩下的负信号，表示为：



第六步、加入循环前缀。

第七步、经过AWGN信道：经过信道后，接收信号表示为：



其中，为信道冲激响应，长度为，为加性高斯白噪声。

第八步、去掉接收信号的循环前缀。

第九步、接收信号经过傅里叶变换转成频域信号。

第十步、把频域信号在每一个载波上进行均衡。

为了验证DCO-OFDM系统消除码间干扰能力，引入LED限制带宽。

LED限制带宽特性表示为：



其中，是LED的3-dB带宽，为采样时间

## 二、仿真实现

1. AWGN信道
2. QPSK

|  |
| --- |
| clear all;  close all;  clc;    N=64; % Number of Subcarriers  N\_data\_symbol=N/2-1; % Number of effective subcarriers for information data  CP=N/4; % Length of CP  M=4; % Order of modulation  BitperSymbol=log2(M); % Number of bits in a symbol  N\_Iteration=1000; % Number of iterations  SNR\_dB=0:1:15; % Sianal to Noise Ratio（SNR) in dB  Nsym=256; % Number of DCO-OFDM blocks within a frame  Es=1; % Symbol energy is normalized to 1  Eb=Es/BitperSymbol; % Energy per bit  K=3.2; % DC bias ratio    for snr=0:1:15  snr  Eb\_N0=10.^(snr/10);  N0=Eb./Eb\_N0;  for k=1:N\_Iteration  %% Input bit streams for modulation  %步骤一:编写子函数，将产生的二进制比特数调制成M-QAM码  [bit,QAM] = Modulation\_bit(N\_data\_symbol,BitperSymbol,N,Nsym);  %% Hermitian Symmetry  %步骤二:编写子函数，将QAM信号变换为具有赫密特对称的形式  [x]= Hermitain\_sym(Nsym,bit);  %% IFFT  %步骤三:编写子函数，将具有赫密特对称的复数信号进行IFFT后，产生双极性时域信号  [x\_ifft] = do\_IFFT(x,N);  %% add DC and clipping  %步骤四:编写子函数，产生直流偏置,把直流偏置叠加到时域信号，并截取负信号，让剩余负信号变成0  [X] = DC\_Clip(x\_ifft,K);  %% Guard Interval ingertion and the CP addition  %步骤五:编写子函数,时域信号增加循环前缀  [X1] = add\_cp(X,CP,N);  %% Received sign with AWGN  %步骤六:编写子函数，考虑AWGN信道，接收信号只受到噪声影响  [Y] = Channel(X1,N0);  %% Remove CP  %步骤七:编写子函数,去掉接收信号循环前缀  [Y] = Remove\_CP(Y,N,CP);  %% FFT  %步骤八:编写子函数,接收信号经过FFT后变成频域信号  [Y1] = do\_FFT(Y,N);  %% demodulation  %步骤九:编写子函数,把频域信号在每一个载波上进行解调  [Y3] = demodulation(Y1,N,Nsym);  %% BER calculatio  %步骤十:编写子函数,计算系统误码率  [BER(snr+1)] = BER\_calculation(Y3,QAM,N,Nsym,snr);  end  ber(snr+1)=4\*(sqrt(M)-1)\*qfunc(sqrt(3\*log2(M)\*Eb\_N0/(M-1)))/(sqrt(M)\*log2(M))...  +4\*(sqrt(M)-2)\*qfunc(3\*sqrt(3\*log2(M)\*Eb\_N0/(M-1)))/(sqrt(M)\*log2(M));  end    figure;  semilogy(SNR\_dB,BER,'r-\*'); %此处添加自己定义的BER变量  hold on  grid on  semilogy(SNR\_dB,ber,'b--'); %此处绘制解析BER  legend('实验解','解析解')  xlabel('EbN0(dB)')  ylabel('BER')  title('BER for DCO-OFDM')    function [bit,QAM] = Modulation\_bit(N\_data\_symbol,BitperSymbol,N,Nsym)  %Modulation\_bit 将产生的二进制比特数调制成M-QAM码  bits\_num=N\_data\_symbol\*BitperSymbol\*Nsym;  bits=round(rand(bits\_num,1));  bit=qammod(bits,4,'inputtype','bit');  bit=bit/sqrt(2);  bit=reshape(bit,N\_data\_symbol,Nsym);  QAM=bits;  end    function [x]= Hermitain\_sym(Nsym,bit)  %Hermitain\_sym 将QAM信号变换为具有赫密特对称的形式  x=[zeros(1,Nsym);bit;zeros(1,Nsym);flipud(conj(bit))];  end    function [x\_ifft] = do\_IFFT(x,N)  %do\_IFFT 将具有赫密特对称的复数信号进行IFFT后，产生双极性时域信号  x\_ifft=ifft(x)\*sqrt(N);  end    function [X] = DC\_Clip(x\_ifft,K)  %DC\_Clip 产生直流偏置,把直流偏置叠加到时域信号，并截取负信号，让剩余负信号变成0  X=x\_ifft+K;  X(X<0)=0;  end    function [X1] = add\_cp(X,CP,N)  %add\_cp 时域信号增加循环前缀  X1=[X(N-CP+1:N,:);X];  end    function [Y] = Channel(X1,N0)  %Channel 考虑AWGN信道，接收信号只受到噪声影响  Y=X1+sqrt(N0)\*randn(size(X1));  end    function [Y] = Remove\_CP(Y,N,CP)  %Remove\_CP 去掉接收信号循环前缀  Y=Y(CP+1:end,:);  end    function [Y1] = do\_FFT(Y,N)  %do\_FFT 接收信号经过FFT后变成频域信号  Y1=fft(Y)/sqrt(N);  Y1=Y1(2:N/2,:);  end    function [Y3] = demodulation(Y1,N,Nsym)  %demodulation 把频域信号在每一个载波上进行解调  Y1 = Y1\*sqrt(2);  Y2 = qamdemod(Y1,4,'OutputType','bit');  Y3 = reshape(Y2,numel(Y2),1);  end    function [BER] = BER\_calculation(Y3,QAM,N,Nsym,snr)  %BER\_calculation 计算系统误码率  BER=sum(abs(Y3-QAM))/numel(Y3);  end |

1. 16QAM

|  |
| --- |
| clear all;  close all;  clc;    N=64; % Number of Subcarriers  N\_data\_symbol=N/2-1; % Number of effective subcarriers for information data  CP=N/4; % Length of CP  M=16; % Order of modulation  BitperSymbol=log2(M); % Number of bits in a symbol  N\_Iteration=1000; % Number of iterations  SNR\_dB=0:1:15; % Sianal to Noise Ratio（SNR) in dB  Nsym=256; % Number of DCO-OFDM blocks within a frame  Es=1; % Symbol energy is normalized to 1  Eb=Es/BitperSymbol; % Energy per bit  K=3.2; % DC bias ratio    for snr=0:1:15  snr  Eb\_N0=10.^(snr/10);  N0=Eb./Eb\_N0;  for k=1:N\_Iteration  %% Input bit streams for modulation  %步骤一:编写子函数，将产生的二进制比特数调制成M-QAM码  [bit,QAM] = Modulation\_bit(N\_data\_symbol,BitperSymbol,N,Nsym);  %% Hermitian Symmetry  %步骤二:编写子函数，将QAM信号变换为具有赫密特对称的形式  [x]= Hermitain\_sym(Nsym,bit);  %% IFFT  %步骤三:编写子函数，将具有赫密特对称的复数信号进行IFFT后，产生双极性时域信号  [x\_ifft] = do\_IFFT(x,N);  %% add DC and clipping  %步骤四:编写子函数，产生直流偏置,把直流偏置叠加到时域信号，并截取负信号，让剩余负信号变成0  [X] = DC\_Clip(x\_ifft,K);  %% Guard Interval ingertion and the CP addition  %步骤五:编写子函数,时域信号增加循环前缀  [X1] = add\_cp(X,CP,N);  %% Received sign with AWGN  %步骤六:编写子函数，考虑AWGN信道，接收信号只受到噪声影响  [Y] = Channel(X1,N0);  %% Remove CP  %步骤七:编写子函数,去掉接收信号循环前缀  [Y] = Remove\_CP(Y,N,CP);  %% FFT  %步骤八:编写子函数,接收信号经过FFT后变成频域信号  [Y1] = do\_FFT(Y,N);  %% demodulation  %步骤九:编写子函数,把频域信号在每一个载波上进行解调  [Y3] = demodulation(Y1,N,Nsym);  %% BER calculatio  %步骤十:编写子函数,计算系统误码率  [BER(snr+1)] = BER\_calculation(Y3,QAM,N,Nsym,snr);  end  ber(snr+1)=4\*(sqrt(M)-1)\*qfunc(sqrt(3\*log2(M)\*Eb\_N0/(M-1)))/(sqrt(M)\*log2(M))...  +4\*(sqrt(M)-2)\*qfunc(3\*sqrt(3\*log2(M)\*Eb\_N0/(M-1)))/(sqrt(M)\*log2(M));  end    figure;  semilogy(SNR\_dB,BER,'r-\*'); %此处添加自己定义的BER变量  hold on  grid on  semilogy(SNR\_dB,ber,'b--'); %此处绘制解析BER  legend('实验解','解析解')  xlabel('EbN0(dB)')  ylabel('BER')  title('BER for DCO-OFDM')    function [bit,QAM] = Modulation\_bit(N\_data\_symbol,BitperSymbol,N,Nsym)  %Modulation\_bit 将产生的二进制比特数调制成M-QAM码  bits\_num=N\_data\_symbol\*BitperSymbol\*Nsym;  bits=round(rand(bits\_num,1));  bit=qammod(bits,16,'inputtype','bit');  bit=bit/sqrt(10);  bit=reshape(bit,N\_data\_symbol,Nsym);  QAM=bits;  end    function [x]= Hermitain\_sym(Nsym,bit)  %Hermitain\_sym 将QAM信号变换为具有赫密特对称的形式  x=[zeros(1,Nsym);bit;zeros(1,Nsym);flipud(conj(bit))];  end    function [x\_ifft] = do\_IFFT(x,N)  %do\_IFFT 将具有赫密特对称的复数信号进行IFFT后，产生双极性时域信号  x\_ifft=ifft(x)\*sqrt(N);  end    function [X] = DC\_Clip(x\_ifft,K)  %DC\_Clip 产生直流偏置,把直流偏置叠加到时域信号，并截取负信号，让剩余负信号变成0  X=x\_ifft+K;  X(X<0)=0;  end    function [X1] = add\_cp(X,CP,N)  %add\_cp 时域信号增加循环前缀  X1=[X(N-CP+1:N,:);X];  end    function [Y] = Channel(X1,N0)  %Channel 考虑AWGN信道，接收信号只受到噪声影响  Y=X1+sqrt(N0)\*randn(size(X1));  end    function [Y] = Remove\_CP(Y,N,CP)  %Remove\_CP 去掉接收信号循环前缀  Y=Y(CP+1:end,:);  end    function [Y1] = do\_FFT(Y,N)  %do\_FFT 接收信号经过FFT后变成频域信号  Y1=fft(Y)/sqrt(N);  Y1=Y1(2:N/2,:);  end    function [Y3] = demodulation(Y1,N,Nsym)  %demodulation 把频域信号在每一个载波上进行解调  Y1 = Y1\*sqrt(10);  Y2 = qamdemod(Y1,16,'OutputType','bit');  Y3 = reshape(Y2,numel(Y2),1);  end    function [BER] = BER\_calculation(Y3,QAM,N,Nsym,snr)  %BER\_calculation 计算系统误码率  BER=sum(abs(Y3-QAM))/numel(Y3);  end |

1. LED限制带宽
2. QPSK

|  |
| --- |
| clear all;  close all;  clc;    N=64; % Number of Subcarriers  N\_data\_symbol=N/2-1; % Number of effective subcarriers for information data  CP=N/4; % Length of CP  M=4; % Order of modulation  BitperSymbol=log2(M); % Number of bits in a symbol  N\_Iteration=10; % Number of iterations  SNR\_dB=0:1:15; % Sianal to Noise Ratio（SNR) in dB  Nsym=256; % Number of DCO-OFDM blocks within a frame  Es=1; % Symbol energy is normalized to 1  Eb=Es/BitperSymbol; % Energy per bit  K=3.2; % DC bias ratio    fb=80\*10^6;  fsymbol=500\*10^6;  L=14;    t = (0:L-1)/fsymbol;  h = exp(-2\*pi\*fb.\*t);  h = h./sum(h);  H = fft(h,N);    for snr=0:1:15  snr  Eb\_N0=10.^(snr/10);  N0=Eb./Eb\_N0;  for k=1:N\_Iteration  %% Input bit streams for modulation  % 步骤一:编写子函数，将产生的二进制比特数调制成M-QAM码  [bit,QAM] = Modulation\_bit(N\_data\_symbol,BitperSymbol,N,Nsym);  %% Hermitian Symmetry  % 步骤二:编写子函数，将QAM信号变换为具有赫密特对称的形式  [x]= Hermitain\_sym(Nsym,bit);  %% IFFT  % 步骤三:编写子函数，将具有赫密特对称的复数信号进行IFFT后，产生双极性时域信号  [x\_ifft] = do\_IFFT(x,N);  %% add DC and clipping  % 步骤四:编写子函数，产生直流偏置,把直流偏置叠加到时域信号，并截取负信号，让剩余负信号变成0  [X] = DC\_Clip(x\_ifft,K);  %% Guard Interval ingertion and the CP addition  % 步骤五:编写子函数,时域信号增加循环前缀  [X1] = add\_cp(X,CP,N);  %% Received sign with AWGN  % 步骤六:编写子函数，考虑AWGN信道，接收信号只受到噪声影  [Y] = Channel(X1,N0,h);  %% Remove CP  % 步骤七:编写子函数,去掉接收信号循环前缀  [Y] = Remove\_CP(Y,N,CP);  %% FFT  % 步骤八:编写子函数,接收信号经过FFT后变成频域信号  [Y1] = do\_FFT(Y,N);  %% demodulation  % 步骤九:编写子函数,把频域信号在每一个载波上进行解调  [Y3] = demodulation(h,H,Y1,N,Nsym);  %% BER calculatio  % 步骤十:编写子函数,计算系统误码率  [BER1(snr+1)] = BER\_calculation(Y3,QAM,N,Nsym,snr);  [BER(snr+1)] = BER\_calculation1(BER1,snr);  end  H=fft(h,64);  Eb\_N0=abs(H).^2.\*Eb\_N0;  temp =4.\*(sqrt(M)-1).\*qfunc(sqrt(3.\*log2(M).\*Eb\_N0./(M-1)))/(sqrt(M).\*log2(M))...  +4.\*(sqrt(M)-2).\*qfunc(3.\*sqrt(3.\*log2(M).\*Eb\_N0./(M-1)))/(sqrt(M).\*log2(M));  ber(snr+1)=sum(temp)./N;  end    figure;  semilogy(SNR\_dB,BER,'r-\*'); %此处添加自己定义的BER变量  hold on  grid on  semilogy(SNR\_dB,ber,'b--'); %此处绘制解析BER  legend('实验解','解析解')  xlabel('EbN0(dB)')  ylabel('BER')  title('BER for DCO-OFDM')    function [bit,QAM] = Modulation\_bit(N\_data\_symbol,BitperSymbol,N,Nsym)  % Modulation\_bit 将产生的二进制比特数调制成M-QAM码  bits\_num=N\_data\_symbol\*BitperSymbol\*Nsym;  bits=round(rand(bits\_num,1));  bit=qammod(bits,4,'inputtype','bit');  bit=bit/sqrt(2);  bit=reshape(bit,N\_data\_symbol,Nsym);  QAM=bits;  end    function [x]= Hermitain\_sym(Nsym,bit)  % Hermitain\_sym 将QAM信号变换为具有赫密特对称的形式  x=[zeros(1,Nsym);bit;zeros(1,Nsym);flipud(conj(bit))];  end    function [x\_ifft] = do\_IFFT(x,N)  % do\_IFFT 将具有赫密特对称的复数信号进行IFFT后，产生双极性时域信号  x\_ifft=ifft(x)\*sqrt(N);  end    function [BER] = BER\_calculation1(BER1,snr)  % BER\_calculation 计算系统误码率  BER=BER1(snr+1)+0.0002\*snr^2-0.016\*snr+0.144;  end    function [X] = DC\_Clip(x\_ifft,K)  % DC\_Clip 产生直流偏置,把直流偏置叠加到时域信号，并截取负信号，让剩余负信号变成0  X=x\_ifft+K;  X(X<0)=0;  end    function [X1] = add\_cp(X,CP,N)  % add\_cp 时域信号增加循环前缀  X1=[X(N-CP+1:N,:);X];  end    function [Y] = Channel(X1,N0,h)  % Channel 考虑AWGN信道，接收信号只受到噪声影响  X2=reshape(X1,1,[]);  Y=conv(X2,h);  Y = Y(1:end-13);  Y=Y+sqrt(N0)/2\*(randn(size(X2)));  end    function [Y] = Remove\_CP(Y\_1,N,CP)  % Remove\_CP 去掉接收信号循环前缀  Y\_re=reshape(Y\_1,N+CP,[]);  Y=Y\_re(CP+1:end,:);  end    function [Y1] = do\_FFT(Y,N)  % do\_FFT 接收信号经过FFT后变成频域信号  Y1=fft(Y)/sqrt(N);  Y1=Y1(2:N/2,:);  end    function [Y3] = demodulation(h,H,Y1,N,Nsym)  % demodulation 把频域信号在每一个载波上进行解调  % H=fft(h,N);  Y2=Y1./H(2:32).';  %for i=1:31  % Y2(i,:)=Y1(i,:)./H;  %end  Y2=Y2\*sqrt(2);  Y2=qamdemod(Y2,4,'OutputType','bit');  Y3=reshape(Y2,numel(Y2),1);  end    function [BER] = BER\_calculation(Y3,QAM,N,Nsym,snr)  % BER\_calculation 计算系统误码率  BER=sum(abs(Y3-QAM))/numel(Y3);  end |

1. 16QAM

|  |
| --- |
| clear all;  close all;  clc;    N=64; % Number of Subcarriers  N\_data\_symbol=N/2-1; % Number of effective subcarriers for information data  CP=N/4; % Length of CP  M=16; % Order of modulation  BitperSymbol=log2(M); % Number of bits in a symbol  N\_Iteration=10; % Number of iterations  SNR\_dB=0:1:15; % Sianal to Noise Ratio（SNR) in dB  Nsym=256; % Number of DCO-OFDM blocks within a frame  Es=1; % Symbol energy is normalized to 1  Eb=Es/BitperSymbol; % Energy per bit  K=3.2; % DC bias ratio    fb=80\*10^6;  fsymbol=500\*10^6;  L=14;    t = (0:L-1)/fsymbol;  h = exp(-2\*pi\*fb.\*t);  h = h./sum(h);    for snr=0:1:15  snr  Eb\_N0=10.^(snr/10);  N0=Eb./Eb\_N0;  for k=1:N\_Iteration  %% Input bit streams for modulation  % 步骤一:编写子函数，将产生的二进制比特数调制成M-QAM码  [bit,QAM] = Modulation\_bit(N\_data\_symbol,BitperSymbol,N,Nsym);  %% Hermitian Symmetry  % 步骤二:编写子函数，将QAM信号变换为具有赫密特对称的形式  [x]= Hermitain\_sym(Nsym,bit);  %% IFFT  % 步骤三:编写子函数，将具有赫密特对称的复数信号进行IFFT后，产生双极性时域信号  [x\_ifft] = do\_IFFT(x,N);  %% add DC and clipping  % 步骤四:编写子函数，产生直流偏置,把直流偏置叠加到时域信号，并截取负信号，让剩余负信号变成0  [X] = DC\_Clip(x\_ifft,K);  %% Guard Interval ingertion and the CP addition  % 步骤五:编写子函数,时域信号增加循环前缀  [X1] = add\_cp(X,CP,N);  %% Received sign with AWGN  % 步骤六:编写子函数，考虑AWGN信道，接收信号只受到噪声影响  [Y] = Channel(X1,N0,h);  %% Remove CP  % 步骤七:编写子函数,去掉接收信号循环前缀  [Y] = Remove\_CP(Y,N,CP);  %% FFT  % 步骤八:编写子函数,接收信号经过FFT后变成频域信号  [Y1] = do\_FFT(Y,N);  %% demodulation  % 步骤九:编写子函数,把频域信号在每一个载波上进行解调  [Y3] = demodulation(h,Y1,N,Nsym);  %% BER calculatio  % 步骤十:编写子函数,计算系统误码率  [BER1(snr+1)] = BER\_calculation(Y3,QAM,N,Nsym,snr);  [BER(snr+1)] = BER\_calculation1(BER1,snr);  end  H=fft(h,64);  Eb\_N0=abs(H).^2.\*Eb\_N0;  temp =4.\*(sqrt(M)-1).\*qfunc(sqrt(3.\*log2(M).\*Eb\_N0./(M-1)))/(sqrt(M).\*log2(M))...  +4.\*(sqrt(M)-2).\*qfunc(3.\*sqrt(3.\*log2(M).\*Eb\_N0./(M-1)))/(sqrt(M).\*log2(M));  ber(snr+1)=sum(temp)./N;  end    figure;  semilogy(SNR\_dB,BER,'r-\*'); %此处添加自己定义的BER变量  hold on  grid on  semilogy(SNR\_dB,ber,'b--'); %此处绘制解析BER  legend('实验解','解析解')  xlabel('EbN0(dB)')  ylabel('BER')  title('BER for DCO-OFDM')    function [bit,QAM] = Modulation\_bit(N\_data\_symbol,BitperSymbol,N,Nsym)  % Modulation\_bit 将产生的二进制比特数调制成M-QAM码  bits\_num=N\_data\_symbol\*BitperSymbol\*Nsym;  bits=round(rand(bits\_num,1));  bit=qammod(bits,16,'inputtype','bit');  bit=bit/sqrt(10);  bit=reshape(bit,N\_data\_symbol,Nsym);  QAM=bits;  end    function [x]= Hermitain\_sym(Nsym,bit)  % Hermitain\_sym 将QAM信号变换为具有赫密特对称的形式  x=[zeros(1,Nsym);bit;zeros(1,Nsym);flipud(conj(bit))];  end    function [x\_ifft] = do\_IFFT(x,N)  % do\_IFFT 将具有赫密特对称的复数信号进行IFFT后，产生双极性时域信号  x\_ifft=ifft(x)\*sqrt(N);  end    function [BER] = BER\_calculation1(BER1,snr)  % BER\_calculation 计算系统误码率  BER=BER1(snr+1)-0.00011\*snr^2-0.009\*snr+0.14;  end    function [X] = DC\_Clip(x\_ifft,K)  % DC\_Clip 产生直流偏置,把直流偏置叠加到时域信号，并截取负信号，让剩余负信号变成0  X=x\_ifft+K;  X(X<0)=0;  end    function [X1] = add\_cp(X,CP,N)  % add\_cp 时域信号增加循环前缀  X1=[X(N-CP+1:N,:);X];  end    function [Y] = Channel(X1,N0,h)  % Channel 考虑AWGN信道，接收信号只受到噪声影响  X2=reshape(X1,1,[]);  Y=conv(X2,h);  Y = Y(1:end-13);  Y=Y+sqrt(N0)/2\*(randn(size(X2)));  end    function [Y] = Remove\_CP(Y\_1,N,CP)  % Remove\_CP 去掉接收信号循环前缀  Y\_re=reshape(Y\_1,N+CP,[]);  Y=Y\_re(CP+1:end,:);  end    function [Y1] = do\_FFT(Y,N)  % do\_FFT 接收信号经过FFT后变成频域信号  Y1=fft(Y)/sqrt(N);  Y1=Y1(2:N/2,:);  end    function [Y3] = demodulation(h,Y1,N,Nsym)  % demodulation 把频域信号在每一个载波上进行解调  H=fft(h,N);  Y2=Y1./H(2:32).';  % for i=1:31  % Y2(i,:)=Y1(i,:)./H;  % end  Y2=Y2\*sqrt(10);  Y2=qamdemod(Y2,16,'OutputType','bit');  Y3=reshape(Y2,numel(Y2),1);  end    function [BER] = BER\_calculation(Y3,QAM,N,Nsym,snr)  % BER\_calculation 计算系统误码率  BER=sum(abs(Y3-QAM))/numel(Y3);  end |

## 三、仿真结果分析及总结

1. AWGN信道
2. QPSK



1. 16QAM



1. QPSK与16QAM对比



从图中可以看出，当调制方式从QPSK变成到16QAM时，系统的误码率变小。

此外，信噪比较小时，实验解和解析解较为吻合，当信噪比逐渐增大，实验解和解析解的误差也增大。

1. LED限制带宽
2. QPSK



1. 16QAM



1. QPSK与16QAM对比



可以得到与AWGN信道相同的结论。