```
% OFDM Code
% coding used: Convolutional coding
% Single frame size: 96 bits
% Total no. of Frames: 100
% Modulation: 16-QAM
% No. of Pilots: 4
% Cylic Extension: 25%(16)

close all
clear all
clc


%%
% Generating and coding data
t_data=randint(9600,1)';
x=1;
si=1; %for BER rows
%%
for d=1:100;
data=t_data(x:x+95);
x=x+96;
k=3;
n=6;
s1=size(data,2);  % Size of input matrix
j=s1/k;

%%
% Convolutionally encoding data
constlen=7;
codegen = [171 133];    % Polynomial
trellis = poly2trellis(constlen, codegen);
codedata = convenc(data, trellis);



%%
%Interleaving coded data

s2=size(codedata,2);
j=s2/4;
matrix=reshape(codedata,j,4);

intlvddata = matintrlv(matrix',2,2)'; % Interleave.
intlvddata=intlvddata';


%%
% Binary to decimal conversion

dec=bi2de(intlvddata','left-msb');


%%
```

```
%16-QAM Modulation

M=16;
y = qammod(dec,M);
% scatterplot(y);


%%
% Pilot insertion

lendata=length(y);
pilt=3+3j;
nofpits=4;

k=1;

for i=(1:13:52)

    pilt_data1(i)=pilt;

    for j=(i+1:i+12);
        pilt_data1(j)=y(k);
        k=k+1;
    end
end

pilt_data1=pilt_data1';   % size of pilt_data =52
pilt_data(1:52)=pilt_data1(1:52);    % upsizing to 64
pilt_data(13:64)=pilt_data1(1:52);   % upsizing to 64

for i=1:52

    pilt_data(i+6)=pilt_data1(i);

end


%%
% IFFT

ifft_sig=ifft(pilt_data',64);


%%
% Adding Cyclic Extension

cext_data=zeros(80,1);
cext_data(1:16)=ifft_sig(49:64);
for i=1:64

    cext_data(i+16)=ifft_sig(i);

end
```

```
%%
% Channel

 % SNR

 o=1;
for snr=0:2:50

ofdm_sig=awgn(cext_data,snr,'measured'); % Adding white Gaussian Noise
% figure;
% index=1:80;
% plot(index,cext_data,'b',index,ofdm_sig,'r'); %plot both signals
% legend('Original Signal to be Transmitted','Signal with AWGN');


%%
%                   RECEIVER
%%
%Removing Cyclic Extension

for i=1:64

    rxed_sig(i)=ofdm_sig(i+16);

end


%%
% FFT

ff_sig=fft(rxed_sig,64);

%%
% Pilot Synch%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


for i=1:52

    synched_sig1(i)=ff_sig(i+6);

end

k=1;

for i=(1:13:52)

    for j=(i+1:i+12);
        synched_sig(k)=synched_sig1(j);
        k=k+1;
    end
end
```

```
% scatterplot(synched_sig)


%%
% Demodulation
dem_data= qamdemod(synched_sig,16);


%%
% Decimal to binary conversion

bin=de2bi(dem_data','left-msb');
bin=bin';


%%
% De-Interleaving


deintlvddata = matdeintrlv(bin,2,2); % De-Interleave
deintlvddata=deintlvddata';
deintlvddata=deintlvddata(:)';




%%
%Decoding data
n=6;
k=3;
decodedata =vitdec(deintlvddata,trellis,5,'trunc','hard');  % decoding
datausing veterbi decoder
rxed_data=decodedata;

%%
% Calculating BER
rxed_data=rxed_data(:)';
errors=0;


c=xor(data,rxed_data);
errors=nnz(c);

% for i=1:length(data)
%
%
%     if rxed_data(i)~=data(i);
%         errors=errors+1;
%
%     end
% end
```

```
BER(si,o)=errors/length(data);
o=o+1;

 end % SNR loop ends here
 si=si+1;
end % main data loop

%%
% Time averaging for optimum results

for col=1:25;        %%%change if SNR loop Changed
    ber(1,col)=0;
for row=1:100;


        ber(1,col)=ber(1,col)+BER(row,col);
    end
end
ber=ber./100;

%%
figure
i=0:2:48;
semilogy(i,ber);
title('BER vs SNR');
ylabel('BER');
xlabel('SNR (dB)');
grid on
```

```
%DVB-T 2K Transmission
%The available bandwidth is 8 MHz
%2K is intended for mobile services
clc
clear all;
close all;
%DVB-T Parameters
Tu=224e-6; %useful OFDM symbol period
T=Tu/2048; %baseband elementary period
G=0; %choice of 1/4, 1/8, 1/16, and 1/32
delta=G*Tu; %guard band duration
Ts=delta+Tu; %total OFDM symbol period
Kmax=1705; %number of subcarriers
Kmin=0;
FS=4096; %IFFT/FFT length
q=10; %carrier period to elementary period ratio
fc=q*1/T; %carrier frequency
Rs=4*fc; %simulation period
t=0:1/Rs:Tu;
%Data generator (A)
M=Kmax+1;
rand('state',0);
a=-1+2*round(rand(M,1)).'+i*(-1+2*round(rand(M,1))).';
A=length(a);
info=zeros(FS,1);
info(1:(A/2)) = [ a(1:(A/2)).']; %Zero padding
info((FS-((A/2)-1)):FS) = [ a(((A/2)+1):A).'];
%Subcarriers generation (B)
carriers=FS.*ifft(info,FS);
tt=0:T/2:Tu;
figure(1);
subplot(211);
stem(tt(1:20),real(carriers(1:20)));
subplot(212);
stem(tt(1:20),imag(carriers(1:20)));
title('TRANSMITION');
grid on;
figure(2);
f=(2/T)*(1:(FS))/(FS);
subplot(211);
plot(f,abs(fft(carriers,FS))/FS);
subplot(212);
pwelch(carriers,[],[],[],2/T);
% D/A simulation
L = length(carriers);
chips = [ carriers.';zeros((2*q)-1,L)];
p=1/Rs:1/Rs:T/2;
g=ones(length(p),1); %pulse shape
title('TRANSMITION');
grid on;
figure(3);
stem(p,g);
dummy=conv(g,chips(:));
u=[dummy(1:length(t))]; % (C)
title('TRANSMITION');
grid on;
figure(4);
```

```
subplot(211);
plot(t(1:400),real(u(1:400)));
subplot(212);
plot(t(1:400),imag(u(1:400)));
title('TRANSMITION');
grid on;
figure(5);
ff=(Rs)*(1:(q*FS))/(q*FS);
subplot(211);
plot(ff,abs(fft(u,q*FS))/FS);
subplot(212);
pwelch(u,[],[],[],Rs);
[b,a] = butter(13,1/20); %reconstruction filter
[H,F] = FREQZ(b,a,FS,Rs);
title('TRANSMITION');
grid on;
figure(6);
plot(F,20*log10(abs(H)));
uoft = filter(b,a,u); %baseband signal (D)
title('TRANSMITION');
grid on;
figure(7);
subplot(211);
plot(t(80:480),real(uoft(80:480)));
subplot(212);
plot(t(80:480),imag(uoft(80:480)));
title('TRANSMITION');
grid on;
figure(8);
subplot(211);
plot(ff,abs(fft(uoft,q*FS))/FS);
subplot(212);
pwelch(uoft,[],[],[],Rs);
%Upconverter
s_tilde=(uoft.').*exp(1i*2*pi*fc*t);
s=real(s_tilde); %passband signal (E)
title('TRANSMITION');
grid on;
figure(9);
plot(t(80:480),s(80:480));
title('TRANSMITION');
grid on;
figure(10);
subplot(211);
%plot(ff,abs(fft(((real(uoft).').*cos(2*pi*fc*t)),q*FS))/FS);
%plot(ff,abs(fft(((imag(uoft).').*sin(2*pi*fc*t)),q*FS))/FS);
plot(ff,abs(fft(s,q*FS))/FS);
subplot(212);
%pwelch(((real(uoft).').*cos(2*pi*fc*t)),[],[],[],Rs);
%pwelch(((imag(uoft).').*sin(2*pi*fc*t)),[],[],[],Rs);
pwelch(s,[],[],[],Rs);
title('TRANSMITION');
grid on;
```

```
%DVB-T 2K Reception
clc
clear all;
close all;
Tu=224e-6; %useful OFDM symbol period
T=Tu/2048; %baseband elementary period
G=0; %choice of 1/4, 1/8, 1/16, and 1/32
delta=G*Tu; %guard band duration
Ts=delta+Tu; %total OFDM symbol period
Kmax=1705; %number of subcarriers
Kmin=0;
FS=4096; %IFFT/FFT length
q=10; %carrier period to elementary period ratio
fc=q*1/T; %carrier frequency
Rs=4*fc; %simulation period
t=0:1/Rs:Tu;
tt=0:T/2:Tu;
%Data generator
sM=2;
[x,y] = meshgrid((-sM+1):2:(sM-1),(-sM+1):2:(sM-1));
alphabet = x(:) + 1i*y(:);
N=Kmax+1;
rand('state',0);
a=-1+2*round(rand(N,1)).'+i*(-1+2*round(rand(N,1))).';
A=length(a);
info=zeros(FS,1);
info(1:(A/2)) = [ a(1:(A/2)).'];
info((FS-((A/2)-1)):FS) = [ a(((A/2)+1):A).'];
carriers=FS.*ifft(info,FS);
%Upconverter
L = length(carriers);
chips = [ carriers.';zeros((2*q)-1,L)];
p=1/Rs:1/Rs:T/2;
g=ones(length(p),1);
dummy=conv(g,chips(:));
u=[dummy; zeros(46,1)];
[b,aa] = butter(13,1/20);
uoft = filter(b,aa,u);
delay=64; %Reconstruction filter delay
s_tilde=(uoft(delay+(1:length(t))).').*exp(1i*2*pi*fc*t);
s=real(s_tilde);
%OFDM RECEPTION
%Downconversion
r_tilde=exp(-1i*2*pi*fc*t).*s; %(F)
figure(1);
subplot(211);
plot(t,real(r_tilde));
axis([0e-7 12e-7 -60 60]);
title('RECEPTION');
grid on;
figure(1);
subplot(212);
plot(t,imag(r_tilde));
axis([0e-7 12e-7 -100 150]);
title('RECEPTION');
grid on;
figure(2);
```

```
ff=(Rs)*(1:(q*FS))/(q*FS);
subplot(211);
plot(ff,abs(fft(r_tilde,q*FS))/FS);
title('RECEPTION');
grid on;
figure(2);
subplot(212);
pwelch(r_tilde,[],[],[],Rs);
%Carrier suppression
[B,AA] = butter(3,1/2);
r_info=2*filter(B,AA,r_tilde); %Baseband signal continuous-time (G)
figure(3);
subplot(211);
plot(t,real(r_info));
axis([0 12e-7 -60 60]);
title('RECEPTION');
grid on;
figure(3);
subplot(212);
plot(t,imag(r_info));
axis([0 12e-7 -100 150]);
title('RECEPTION');
grid on;
figure(4);
f=(2/T)*(1:(FS))/(FS);
subplot(211);
plot(ff,abs(fft(r_info,q*FS))/FS);
title('RECEPTION');
grid on;
subplot(212);
pwelch(r_info,[],[],[],Rs);
%Sampling
r_data=real(r_info(1:(2*q):length(t)))... %Baseband signal, discretetime
+1i*imag(r_info(1:(2*q):length(t))); % (H)
figure(5);
subplot(211);
stem(tt(1:20),(real(r_data(1:20))));
axis([0 12e-7 -60 60]);
title('RECEPTION');
grid on;
figure(5);
subplot(212);
stem(tt(1:20),(imag(r_data(1:20))));
axis([0 12e-7 -100 150]);
title('RECEPTION');
grid on;
figure(6);
f=(2/T)*(1:(FS))/(FS);
subplot(211);
plot(f,abs(fft(r_data,FS))/FS);
title('RECEPTION');
grid on;
subplot(212);
pwelch(r_data,[],[],[],2/T);
%FFT
info_2N=(1/FS).*fft(r_data,FS); % (I)
info_h=[info_2N(1:A/2) info_2N((FS-((A/2)-1)):FS)];
```

```
%Slicing
for k=1:N,
a_hat(k)=alphabet((info_h(k)-alphabet)==min(info_h(k)-alphabet)); %
end;
figure(7)
plot(info_h((1:A)),'.k');
title('info-h Received Constellation')
axis square;
axis equal;
title('RECEPTION');
grid on;
figure(8)
plot(a_hat((1:A)),'or');
title('a_hat 4-QAM')
axis square;
axis equal;
title('RECEPTION');
grid on;
axis([-1.5 1.5 -1.5 1.5]);
```