REFRENCES

1- SiddharthShrikantPisal, Faculty of San Diego State University, Partial Fulfillment of the Requirements for the Degree Master of Science in Electrical Engineering ,"PHYSICAL LAYER COMPARATIVE STUDY OF WiMAX AND LTE", 2012.

2- Monika Sehrawat, Ritika Singh, Priyanka Sharma,M.Tech Scholar, Assistant ProfessorDepartment of Electronics & Communication Engineering, SPGOI Rohtak (India),"PERFORMANCE COMPARISON BETWEEN OFDMAAND SC-FDMA USING RAYLEING FADINGCHANNEL",2015.

3- Abdul SamadShaikh , KhatriChandan Kumar, thesis is presented as part of Master of Science in Electrical Engineering, Blekinge Institute of Technology, "Performance Evaluation of LTE Physical Layer Using SC-FDMA & OFDMA", November 2010.

4- Hyung G. Myung and David J. Goodman ,"Single Carrier FDMA: A New Air Interface for Long Term Evolution", John Wiley & Sons, Ltd,2008.

5- GaetanoGuida, Valerio Massa, Fabio Parente, "LTE and 4G: a comparison with WiMAX, ", 2010-2011.

6- Murtadha Ali NsaifShukur, Dr. KuldipPahwa, Er. AnkurSinghal, M. M. University, Mullana, Haryana, India,"IMPLEMENTING SC-FDMA &OFDMA IN MATLAB",2013.

7- A.Z.Yonis, Departement of Communication Engineering, College of Electronics Engineering, University of Mosul, Iraq,M.F.L.Abdullah, Faculty of Electrical and Electronic Engineering, University Tun Hussein Onn Malaysia," Uplink and Downlink of LTE-Release 10 in Cellular Communications ", 2012.

8- A.Manikandan, V.Venkataramanan, M.Kavitha,S.Parvathi, "PERFORMANCE ANALYSIS OF LTE PHYSICAL LAYER BASED ON RELEASE 8&9 THROUGH SIMULINK ENVIRONMENT", 2012.

9- Gaetano Guida, Valerio Massa, Fabio Parente ,"LTE and 4G: a comparison withWiMAX",2011.

9- DieterEberle , Computer Engineering B.Sc., Berlin Institute of Technology,Germany, "LTEvs.WiMAX 4th generation telecommunication networks",2010.

10- Mihai-Florentin URSULEANU, Daniel SIMION, Adrian GRAUR and Alin Dan POTORAC ,"A Comparative Approach on WiMax and LTE Technologies",2010.

11- Qinghua Li and Guangjie Li, Intel," MIMO Techniques in WiMAX and LTE: A Feature Overview",2010.

12- M.A. Mohamed, F.W. Zaki, R.H. Mosbeh," Simulation of WiMAX Physical Layer: IEEE 802.16e ",2010.

13- Syed Hamid Ali Shah MudasarIqbalTassadaqHussain" COMPARISON BETWEEN WiMAX AND LTE", thesis ispresented aspartMaster of Science in Electrical Engineering, Blekinge Institute of Technology, August 2009.

APPENDIX

OFDMA:

```
%No.ofSubcarriers
NS=512;
%InputGeneration
x=rand(1,NS)>0.5;
fftlength=512;
nd =6;
BW=5e6;
FS=2×BW;%SamplingFrequency
%Conversion ofdatafromserialto parallel
p = series2parallel(x,NS);
%Mary Modulationof PSKand QAM
M= 2;
X=0;
forcount1=2:1:7;
if  (M==2||M==4||M==16||M==64)M=M+X
%M-ary modulationfor producingy.
If (M<=8)
%ModulationforPSK
y=modulate(modem.pskmod(M),p);
else
%ModulationforQAM
y=modulate(modem.qammod(M),p);
end
ylen=length(y);
%ApplyingMapping
q_out=ofdma_mapping(y,ylen);
%ApplyIFFToperation
outifft=ifft(q_out);
%CyclicPrefixAddition
cp(count1,:)= cyclicpad(outifft,64);
%LengthofCP
cplength=length(cp);
%Conversion ofdatafromparallelto serial
out =reshape(cp(count1,:),1,cplength);
%SignaltransmitsthroughAWGNchannel.
ynoisy=awgn(out,100,'measured');
%Addition ofRayleighFading
c=rayleighchan(1/1000,100,[02e-5],[0-9]);
rf=filter(c,ynoisy);
%Conversion ofdatafromserialto parallel
p2= series2parallel(rf,cplength);
re_par=real(p2);
%Removecyclicprefix
rcp(count1,:)= decyclicpad(p2,64);rcplength= length(rcp);
%FFT
zzfft=fft(rcp(count1,:),fftlength);
```

```
%ApplyDemapping
qq_out=ofdma_demapping(zzfft,fftlength);
outfft=qq_out;
%ApplyingIFFT.
zfft=ifft(qq_out);
if(M<=8)
%DemodulationofPSKatReciever
z=demodulate(modem.pskdemod(M),outfft);
else
%DemodulationofQAMat Reciever
z=demodulate(modem.qamdemod(M),outfft);
end
%Conversion ofdatafromparallelto serial
xdash =reshape(z,1,NS)
berr=0 ;
fora =1:1:NS;
if(xdash(:,a)==x(:,a))
berr=0;
else
berr=berr+1;
end
end
tberr(count1,:)=berr; Eb_No=0:1:NS-1;
Eb_No =0.4*Eb_No;
if(M<=8)
ber(count1,:)=berawgn(Eb_No,'psk',M,'nondiff');
Pe(count1,:)=erfc(sqrt(0.9*Eb_No)*sin(pi/M));
else
ber1(count1,:)=berawgn(0.9*Eb_No,'qam',M);
Pe(count1,:)=2*((1-(1/sqrt(M)))*erfc(sqrt((1.5*Eb_No)/(M-1))));
end
forinit=1:1:32 switchM
end
end
end
M=2^count1;
endfigure()
%PlotSNRand BER
semilogy(Eb_No,ber(2,:),'k',Eb_No,ber(3,:),'g',Eb_No,ber1(5,:),'b',Eb_No,ber1(7,:),'r');
axis([0 25 0.00011]);
xlabel('SNR[dB]')
ylabel('BER')
legend('BPSK','QPSK','16-QAM','64-QAM')
title('OFDMA')
figure()
%PlotErrorProbability
semilogy(Eb_No,Pe(2,:),'k',Eb_No,Pe(3,:),'r',Eb_No,Pe(5,:),'b',Eb_No,Pe(7,:),'g');
axis([0 50 0.00011]);
xlabel('SNR[dB]')
ylabel('ProbabilityofError')
legend('BPSK','QPSK','16-QAM','64-QAM')
```

```
title('OFDMA')
h=spectrum.periodogram;
figure()
HS=psd(h,outifft,'SpectrumType','twosided','NFFT',NS,'FS',FS);
plot(HS)
xlabel('SamplingFrequency(2×BW)inMHz')
ylabel('PowerSpectralDensity [dBm/Hz]')
title('OFDMA')
gridoff;
```

SC-FDMA:

```
%No.ofSubcarriers
NS=512;
%InputGeneration
x=rand(1,NS)>0.5;
fftlength=512;
nd =6;
BW=5e6;
FS=2×BW;%SamplingFrequency
%Conversion ofdatafromserialto parallel
par2=series2parallel(x,NS);
%Mary Modulationof PSKand QAM
M= 2;
X=0;
forcount1=2:1:7;
if(M==2||M==4||M==16||M==64)M=M+X
%M-ary modulationfor producingy.
if(M<=8)
%ModulationforPSK
y=modulate(modem.pskmod(M),par2);
else
%ModulationforQAM
y=modulate(modem.qammod(M),par2);
end
%ApplyFFToperation
out_fft=fft(y,fftlength);
%ApplyingMapping
q_out=scfdma_mapping(out_fft,fftlength);
%ApplyIFFToperation
out_ifft =ifft(q_out);
%CyclicPrefixAddition
cp(count1,:)=cyclicpad(out_ifft,64);
cplength=length(cp);
%Conversion ofdatafromparallelto serial
out=reshape(cp(count1,:),1,cplength);
%Signaltransmitsthroughan AWGNchannel.
ynoisy=awgn(out,100,'measured');
%RayleighFading
c=rayleighchan(1/1000,100,[02e-5],[0-9]);
rf=filter(c,ynoisy);
```

```
%Conversion ofdatafromserialto parallel
p = series2parallel(rf,cplength);
re_par=real(p);
%Removecyclicprefix
rcp(count1,:)=decyclicpad(p,64);
len_rcp= length(rcp);
%ApplyingFFToperation
cpfft=fft(rcp(count1,:),fftlength);
%ApplyingDemapping
qq_out=scfdma_demapping(cpfft,fftlength);
%ApplyingIFFToperation
out_fft =ifft(qq_out);
if(M<=8)
%DemodulationofPSK
z=demodulate(modem.pskdemod(M),out_fft);
else
%DemodulationofQAM
z=demodulate(modem.qamdemod(M),out_fft);
end
%Conversion ofdatafromparallelto serial
xdash =reshape(z,1,NS)
berr=0 ;
fora =1:1:NS;
if(xdash(:,a)==x(:,a))
berr=0;
else
berr=berr+1;
endend
tberr(count1,:)=berr; Eb_No=0:1:NS-1; if(M<=8)
ber(count1,:)=berawgn(0.9*Eb_No,'psk',M,'nondiff');
Pe(count1,:)=erfc(sqrt(2*Eb_No)*sin(pi/M));
else
ber1(count1,:)=berawgn(0.9*Eb_No,'qam',M);
Pe(count1,:)=2*((1-(1/sqrt(M)))*erfc(sqrt((1.5*Eb_No)/(M-1))));
end
forinit=1:1:32 switchM
endendend
M=2^count1;
endfigure()
%PlotSNRand BER
semilogy(Eb_No,ber(2,:),'*-
r',Eb_No,ber(3,:),'k',Eb_No,ber1(5,:),'b',Eb_No,ber1(7,:),'g');
axis([0 25 0.00011]);
xlabel('SNR[dB]')
ylabel('BER')
legend('BPSK','QPSK','16-QAM','64-QAM')
title('SC-FDMA')
figure()
%PlotErrorProbability
semilogy(Eb_No,Pe(2,:),'-k',Eb_No,Pe(3,:),'-r',Eb_No,Pe(5,:),'-b',Eb_No,Pe(7,:),'-c');
```

```
axis([0 50 0.00011]);
xlabel('SNR[dB]')
ylabel('ProbabilityofError')
legend('BPSK','QPSK','16-QAM','64-QAM')
title('SC-FDMA')
h=spectrum.periodogram;
figure()
HS=psd(h,out_ifft,'SpectrumType','twosided','NFFT',NS,'FS',FS);
plot(HS)
xlabel('SamplingFrequency(2×BW)inMHz')
ylabel('PowerSpectralDensity [dBm/Hz]')
title('SC-FDMA')
gridoff;
```

SerialtoParallel:

```
functiony =series2parallel(x,NS) L=length(x);
q=floor(L/NS); newvec=zeros(NS,q); fori=1:q
newvec(1:NS,i)=x((1+(i-1)*NS):i*NS);
endy=newvec;
```

CyclicPrefix:

```
functiony=cyclicpad(X,L) N=length(X(:,1));
N-L+1

Y=[X(N-L+1:N,:);X];
y=Y;
```

RemoveCyclic Prefix:

```
functiony=decyclicpad(X,L) N=length(X(:,1));
 Y=X(L+1:N,:);
y=Y;
```

Mapping:

```
function[iout,qout]=crmapping(idata,qdata,fftlength,nd);
iout=zeros(fftlength,nd);
qout=zeros(fftlength,nd);
iout(2:27,:)=idata(1:26,:);
qout(2:27,:)=qdata(1:26,:);
iout(39:64,:)=idata(27:52,:);
qout(39:64,:)=qdata(27:52,:);
```

Demapping:

```
function[iout,qout]=crdemapping(idata,qdata,fftlength,nd);
```

```
iout(1:26,:)=idata(2:27,:);
qout(1:26,:)=qdata(2:27,:);
iout(27:52,:)=idata(39:64,:);
qout(27:52,:)=qdata(39:64,:);
```

PAPR:

BPSK:

```
functionpaprSCFDMA()

dataType='B-PSK'; %Modulationformat.
NS =512;%Numberoftotalsubcarriers.
Symbols =16;%Datablocksize.
Q=NS/Symbols;%Bandwidthspreadingfactorof SC-FDMA.
BW = 5e6;% Systembandwidth.
Ts=1/BW;%samplingrate.
osf=4;%Oversamplingfactor. Nsub=NS;
Fsub= [0:Nsub-1]*BW/Nsub;%SubcarrierspacingofOFDMA.

Runs=1e3;%Numberof iterations.

papr1 =zeros(1,Runs);%InitializethePAPRresultsforsc-fdma.
papr3=zeros(1,Runs);%InitializethePAPRresultsforOFDMA

forn =1:Runs,
%Generaterandomdata.
ifdataType=='B-PSK'
tmp=round(rand(Symbols,2));
tmp=tmp*2- 1;
data=(tmp(:,1)+j*tmp(:,2))/sqrt(2);
else ifdataType=='16QAM'
dataSet= [-3+3i-1+3i1+3i3+3i...
-3+i -1+i1+i3+i...
-3-i -1-i1-i3-i...
-3-3i -1-3i1-3i3-3i];
dataSet=dataSet/ sqrt(mean(abs(dataSet).^2));
tmp=ceil(rand(Symbols,1)*16);
fork = 1:Symbols,iftmp(k)==0
tmp(k)=1;
end
data(k)=dataSet(tmp(k));
end
data=data.';
end
%Convertdatato frequencydomain.
Z1=fft(data);
Z2=fft(data);
%Initializethesubcarriers.
Y1=zeros(NS,1);
```

```
Y2=zeros(NS,1);
%Subcarrier mappingforSC-FDMA
Y1(1:Q:NS)=Z1;Y2(1:Symbols)= Z2;
%Convertdatabackto timedomain.
y1 =ifft(Y1);
y2=ifft(Y2);
%OFDMAmodulation.
%TimerangeoftheOFDMAsymbol.
t = [0:Ts/osf:Nsub*Ts];
y3 =0;
fork=1:Symbols,
y3=y3+data(k)*exp(j*2*pi*Fsub(k)*t);
end
%CalculatePAPR.
papr3(n)=10*log10(max(abs(y3).^2)/ mean(abs(y3).^2));
papr1(n)=10*log10(max(abs(y1).^2)/ mean(abs(y1).^2));
papr2(n)=10*log10(max(abs(y2).^2)/ mean(abs(y2).^2));
end
%PlotCCDF.
figure ()

[N,Z3]=hist(papr3,100);
[N,Z1]=hist(papr1,100);
[N,Z2]=hist(papr2,100);

semilogy(Z1,1-cumsum(N)/max(cumsum(N)),'b')
holdon
semilogy(Z3,1-cumsum(N)/max(cumsum(N)),'black')
holdoff
title('PAPRofSC-FDMAand OFDMAforBPSK')
xlabel('PAPR[dB]')
ylabel('{PAPR(PAPR>PAPR0)}')
gridoff;
%savedata.
savepaprSCFDMA
```

QPSK:

```
ifdataType=='QPSK'
tmp=round(rand(Symbols,4));
tmp=tmp*2- 1;
data=(tmp(:,1)+j*tmp(:,2))/sqrt(2);
```

16-QAM:

```
elseifdataType=='16QAM'
dataSet= [-3+3i-1+3i1+3i3+3i...
```

```
-3+i -1+i1+i3+i...
-3-i -1-i1-i3-i...
-3-3i -1-3i1-3i3-3i];
```

64-QAM:

```
elseifdataType=='64QAM'
dataSet= [-5+5i-1+5i1+5i5+5i...
-5+i -1+i1+i5+i...
-5-i -1-i1-i5-i...
-5-5i -1-5i1-5i5-5i];
```