

**LAMPIRAN
LIST PROGRAM**

```

% Simulasi sistem DS-CDMA
clc;
clear all;
close all;
clc;

sr=256000.0;
ml=2;
br=sr*ml;
nd=100;
ebno=1:10;

irfn=21;
IPOINT=1;
alfs=0.5;
[xh]=hrollfcoef(irfn,IPOINT,sr,alfs,1);
[xh2]=hrollfcoef(irfn,IPOINT,sr,alfs,0);

user=2;
seq=2;
stage=3;
ptap1=[1 3];
ptap2=[2 3];
regi1=[1 1 1];
regi2=[1 1 1];

switch seq
case 1
code=mseq(stage,ptap1,regi1,user);
case 2
m1=mseq(stage,ptap1,regi1);
m2=mseq(stage,ptap2,regi2);
code=goldseq(m1,m2,user);
case 3
m1=mseq(stage,ptap1,regi1);
m2=mseq(stage,ptap2,regi2);
code=[goldseq(m1,m2,user),zeros(user,1)];
end
code=code*2-1;
clen=length(code);

rfade=0;
itau=[0,8];
dlv11=[0.0,40.0];
n0=[6,7];
th1=[0.0,0.0];

```

```

itnd1=[3001,4004];
now1=2;
tstp=1/sr/IPOINT/clen;
fd=160;
flat=1;
itnde1=nd*IPOINT*clen*30;

nloop=1000;

for ii=1:nloop

    data=rand(user,nd*ml)>0.5;
    [ich,qch]=qpskmod(data,user,nd,ml);

    [ich1,qch1]=spread(ich,qch,code);

    [ich2,qch2]=compoversamp2(ich1,qch1,IPOINT);
    [ich3,qch3]=compconv2(ich2,qch2,xh);

    if user==1
        ich4=ich3;
        qch4=qch3;
    else
        ich4=sum(ich3);
        qch4=sum(qch3);
    end;

    if rfade==0
        ich5=ich4;
        qch5=qch4;
    else
        [ich5,qch5]=sefade(ich4,qch4,itaу,dlv11,th1,n0,itnd1,...
            now1,length(ich4),tstp,fd,flat);
        itnd1=itnd1+itnde1;
    end;

    for m=1:length(ebno)
        m;
        noe=0;
        nod=0;
        spow=sum(rot90(ich3.^2+qch3.^2))/nd;
        attn=sqrt(0.5*spow*sr/br*10^(-ebno(m)/10));
        [ich6,qch6]=comb2(ich5,qch5,attn);
        [ich7,qch7]=compconv2(ich6,qch6,xh2);
    end;
end;

```

```

samp1=irfn*IPOINT+1;
ich8=ich7(:,samp1:IPOINT:IPOINT*nd*clen+samp1-1);
qch8=qch7(:,samp1:IPOINT:IPOINT*nd*clen+samp1-1);
[ich9 qch9]=despread(ich8,qch8,code);
demodata=qpskdemod(ich9,qch9,user,nd,ml);

noe2=sum(sum(abs(data-demodata)));
nod2=user*nd*ml;
noe=noe+noe2;
nod=nod+nod2;
be(m)=noe/nod;
tmp=be(m);
tmp=10^((m+tmp)/10);
tmp=sqrt(tmp);
ber(m)=0.5*erfc(tmp);
end;
end;

```

```

figure;
semilogy(ber),grid;
xlabel('SNR');ylabel('BER');

```

```

be1=be;
ber1=ber;
save data21 ber1 be21

```

% Menampilkan Grafik

```

clc;
close all;
clear all;
clc;

```

```

load data21;
load data41;
load data71;
semilogy(be1,'r'),grid;hold on;
semilogy(be41,'g'),grid;hold on;
semilogy(be71,'b'),grid;hold on;
xlabel('SNR');ylabel('BER');
legend('user=2','user=4','user=7');

```

```
% autocorr.m
```

```
function [out]=autocorr(indata,tn)
```

```
if nargin<2
```

```
    tn=1;
```

```
end
```

```
N=length(indata);
```

```
out=zeros(1,N*tn);
```

```
for ii=0:N*tn-1
```

```
    out(ii+1)=sum(indata.*shift(indata,ii,0));
```

```
end
```

```
% comb2.m
```

```
function [iout,qout]=comb2(idata,qdata,attn)
```

```
v=length(idata);
```

```
h=length(attn);
```

```
iout=zeros(h,v);
```

```
qout=zeros(h,v);
```

```
for ii=1:h
```

```
    iout(ii,:)=idata+randn(1,v)*attn(ii);
```

```
    qout(ii,:)=qdata+randn(1,v)*attn(ii);
```

```
end
```

```
% comconv2.m
```

```
function [iout,qout]=comconv2(idata,qdata,filter)
```

```
iout=conv2(idata,filter);
```

```
qout=conv2(qdata,filter);
```

```
% compoversamp2.m
```

```
function [iout,qout]=compoversamp2(iin,qin,sample)
```

```
[h,v]=size(iin);
```

```
iout=zeros(h,v*sample);
```

```
qout=zeros(h,v*sample);
```

```
iout(:,1:sample:1+sample*(v-1))=iin;
```

```
qout(:,1:sample:1+sample*(v-1))=qin;
```

```
% crosscorr.m
```

```
function [out]=crosscorr(indata1,indata2,tn)
```

```
if nargin<3
```

```
    tn=1;
```

```
end
```

```
N=length(indata1);
```

```
out=zeros(1,N*tn);
```

```
for ii=0:N*tn-1
```

```
    out(ii+1)=sum(indata1.*shift(indata2,ii,0));
```

```
end
```

```
% delay.m
```

```
function [iout,qout]=delay(idata,qdata,nsamp,idel)
```

```
iout=zeros(1,nsamp);
```

```
qout=zeros(1,nsamp);
```

```
if idel~=0
```

```
    iout(1:idel)=zeros(1,idel);
```

```
    qout(1:idel)=zeros(1,idel);
```

```
end
```

```
iout(idel+1:nsamp)=idata(1:nsamp-idel);
```

```
qout(idel+1:nsamp)=qdata(1:nsamp-idel);
```

```
% despread.m
```

```
function [iout,qout]=despread(idata,qdata,code1)
```

```
switch nargin
```

```
case{0,1}
```

```
error('lack of input argument');
```

```
case 2
```

```
code1=qdata;
```

```
qdata=idata;
```

```
end
```

```
[hn,vn]=size(idata);
```

```
[hc,vc]=size(code1);
```

```
vn=fix(vn/vc);
```

```
iout=zeros(hc,vn);
```

```
qout=zeros(hc,vn);
```

```
for ii=1:hc
```

```
  iout(ii,:)=rot90(flipud(rot90(reshape(idata...
```

```
    (ii,:),vc,vn))*rot90(code1(ii,:),3));
```

```
  qout(ii,:)=rot90(flipud(rot90(reshape(qdata...
```

```
    (ii,:),vc,vn))*rot90(code1(ii,:),3));
```

```
end
```

```
% fade.m
```

```
function [iout,qout,ramp,rcos,rsin]=fade(idata,qdata,...
```

```
  nsamp,tstp,fd,no,counter,flat)
```

```
if fd~=0.0
```

```
  ac0=sqrt(1.0./(2.0.*(no+1)));
```

```
  as0=sqrt(1.0./(2.0.*no));
```

```
  ic0=counter;
```

```
  pai=3.14159265;
```

```
  wm=2.0.*pai.*fd;
```

```
  n=4.0*no+2;
```

```
  ts=tstp;
```

```
  wmts=wm.*ts;
```

```
  paino=pai./no;
```

```
  xc=zeros(1,nsamp);
```

```
  xs=zeros(1,nsamp);
```

```

ic=[1:nsamp]+ic0;

for nn=1:no
    cwn=cos(cos(2.0.*pai.*nn./no).*ic.*wmts);
    xc=xc+cos(paino.*nn).*cwn;
    xs=xs+sin(paino.*nn).*cwn;
end
cwmt=sqrt(2.0).*cos(ic.*wmts);
xc=(2.0.*xc+cwmt).*ac0;
xs=2.0.*xs.*as0;
ramp=sqrt(xc.^2+xs.^2);
rcos=xc./ramp;
rsin=xs./ramp;
if flat==1
    iout=sqrt(xc.^2+xs.^2).*idata(1:nsamp);
    qout=sqrt(xc.^2+xs.^2).*qdata(1:nsamp);
else
    iout=xc.*idata(1:nsamp)-xs.*qdata(1:nsamp);
    qout=xs.*idata(1:nsamp)+xc.*qdata(1:nsamp);
end
else
    iout=idata;
    qout=qdata;
end
end

```

```

% goldseq.m

```

```

function [gout]=goldseq(m1,m2,n)

```

```

if nargin<3
    n=1;
end
gout=zeros(n,length(m1));
for ii=1:n
    gout(ii,:)=xor(m1,m2);
    m2=shift(m2,1,0);
end
end

```



```
% hrollcoef.m
```

```
function [xh]=hrollcoef(irfn,ipoint,sr,alfs,ncc)
```

```
xi=zeros(1,irfn*ipoint+1);  
xq=zeros(1,irfn*ipoint+1);  
point=ipoint;  
tr=sr;  
tstp=1.0./tr./ipoint;  
n=ipoint.*irfn;  
mid=(n./2)+1;  
sub1=4.0.*alfs.*tr;
```

```
for i=1:n  
    icon=i-mid;  
    ym=icon;  
    if icon==0.0  
        xt=(1.0-alfs+4.0.*alfs./pi).*tr;  
    else  
        sub2=16.0.*alfs.*alfs.*ym.*ym./ipoint./ipoint;  
        if sub2~=1.0  
            x1=sin(pi*(1.0-alfs)/ipoint*ym)./pi./...  
                (1.0-sub2)./ym./tstp;  
            x2=cos(pi*(1.0+alfs)/ipoint*ym)./pi.*sub1./(1.0-sub2);  
            xt=x1+x2;  
        else  
            xt=alfs.*tr.*((1.0-2.0/pi).*cos(pi/4.0/alfs)+(1.0+2.0/pi).*...  
                sin(pi/4.0/alfs))./sqrt(2.0);  
        end  
    end  
    if ncc==0  
        xh(i)=xt./ipoint./tr;  
    elseif ncc==1  
        xh(i)=xt./tr;  
    else  
        error('ncc error');  
    end  
end
```

```
% mseq.m
```

```
function [mout]=mseq(stg,taps,inidata,n)
```

```
if nargin<4
```

```
    n=1;
```

```
end
```

```
mout=zeros(n,2^stg-1);
```

```
fpos=zeros(stg,1);
```

```
fpos(taps)=1;
```

```
for ii=1:2^stg-1
```

```
    mout(1,ii)=inidata(stg);
```

```
    num=mod(inidata*fpos,2);
```

```
    inidata(2:stg)=inidata(1:stg-1);
```

```
    inidata(1)=num;
```

```
end
```

```
if n>1
```

```
    for ii=2:n
```

```
        mout(ii,:)=shift(mout(ii-1,:),1,0);
```

```
    end
```

```
end
```

```
% qpskdemod.m
```

```
function [demodata]=qpskdemod(idata,qdata,para,nd,ml)
```

```
demodata=zeros(para,ml*nd);
```

```
demodata((1:para),(1:ml:ml*nd-1))=idata((1:para),(1:nd))>=0;
```

```
demodata((1:para),(2:ml:ml*nd))=qdata((1:para),(1:nd))>=0;
```

```
% qpskmod.m
```

```
function [iout,qout]=qpskmod(paradata,para,nd,ml)
```

```
m2=ml./2;
```

```
paradata2=paradata.*2-1;
```

```
count2=0;
```

```
for jj=1:nd
```

```
    isi=zeros(para,1);
```

```
    isq=zeros(para,1);
```

```
    for ii=1:m2
```

```
        isi=isi+2.^(m2-ii).*paradata2((1:para),ii+count2);
```

```
        isq=isq+2.^(m2-ii).*paradata2((1:para),m2+ii+count2);
```

```
    end
```

```
    iout((1:para),jj)=isi;
```

```
    qout((1:para),jj)=isq;
```

```
    count2=count2+m2;
```

```
end
```

```
% sefade.m
```

```
function [iout,qout,ramp,rcos,rsin]=sefade(idata,qdata,itau,...  
    dlvl,th,n0,itn,n1,nsamp,tstp,fd,flat)
```

```
iout=zeros(1,nsamp);
```

```
qout=zeros(1,nsamp);
```

```
total_attn=sum(10.^(-1.0.*dlvl./10.0));
```

```
for k=1:n1
```

```
    atts=10.^(-0.05.*dlvl(k));
```

```
    if dlvl(k)==40.0
```

```
        atts=0.0;
```

```
    end
```

```
    theta=th(k).*pi./180.0;
```

```
    [itmp,qtmp]=delay(idata,qdata,nsamp,itau(k));
```

```
    [itmp3,qtmp3,ramp,rcos,rsin]=fade(itmp,qtmp,...
```

```
        nsamp,tstp,fd,n0(k),itn(k),flat);
```

```
    iout=iout+atts.*itmp3./sqrt(total_attn);
```

```
    qout=qout+atts.*qtmp3./sqrt(total_attn);
```

```
end
```

```
% shift.m
```

```
function [outregi]=shift(inregi,shiftr,shiftu)
```

```
[h,v]=size(inregi);  
outregi=inregi;  
shiftr=rem(shiftr,v);  
shiftu=rem(shiftu,h);  
if shiftr>0  
    outregi(:,1:shiftr)=inregi(:,v-shiftr+1:v);  
    outregi(:,1+shiftr:v)=inregi(:,1:v-shiftr);  
elseif shiftr<0  
    outregi(:,1:v+shiftr)=inregi(:,1-shiftr:v);  
    outregi(:,v+shiftr+1:v)=inregi(:,1:-shiftr);  
end
```

```
inregi=outregi;
```

```
if shiftu>0  
    outregi(1:h-shiftu,:)=inregi(1+shiftu:h,:);  
    outregi(h-shiftu+1:h,:)=inregi(1:shiftu,:);  
elseif shiftu<0  
    outregi(1:-shiftu,:)=inregi(h+shiftu+1:h,:);  
    outregi(1-shiftu:h,:)=inregi(1:h+shiftu,:);  
end
```

```
% spread.m
```

```
function [iout,qout]=spread(idata,qdata,code1);
```

```
switch nargin  
    case {0,1}  
        error('lack of input argument');  
    case 2  
        code1=qdata;  
        qdata=idata;
```

```
end
```

```
[hn,vn]=size(idata);  
[hc,vc]=size(code1);
```

```
if hn>hc  
    error('lack of spread code sequences');  
end
```

```
iout=zeros(hn,vn*vc);
qout=zeros(hn,vn*vc);

for ii=1:hn
    iout(ii,:)=reshape(rot90(code1(ii,:),3)...
        *idata(ii,:),1,vn*vc);
    qout(ii,:)=reshape(rot90(code1(ii,:),3)...
        *qdata(ii,:),1,vn*vc);
end
```