clear all

pkg load communications

%

function [fidnoi,fid] = gen\_fid(N,res\_f,sw,P) % N - length of the FID, res\_f - resonance freqeuncy of the observed signal, sw - spectral width; P - percentage of left out values of the FID

n=(0:N-1);

dt=1.0/sw;

t=(0:dt:(N-1)\*dt);

ny=res\_f\*t(length(t));

ny2=(res\_f-10.0)\*t(length(t));

R2=1.26/n(length(n));

f\_real=(cos(2\*pi\*ny\*n/N)+cos(2\*pi\*ny2\*n/N)).\*exp(-R2\*n);

f\_imag=(sin(2\*pi\*ny\*n/N)+sin(2\*pi\*ny2\*n/N)).\*exp(-R2\*n);

%S = gen\_scheme\_unif(N,P);

%S = gen\_scheme\_exp(N,P);

%S = gen\_scheme\_gauss(N,P);

%S = gen\_scheme\_cos2(N,P);

%S = gen\_scheme\_poiss(N,P);

S = gen\_scheme\_poiss2(N,P);

fidd=awgn((f\_real+i\*f\_imag),2,'measured');

fid=f\_real.\*S+i\*f\_imag.\*S;

fidnoi=real(fidd).\*S+i\*imag(fidd).\*S;

end

function [C,lw] = check\_res(fq,ft)

ma=max(real(ft));

left=0;

right=0;

tmp=0;

ft\_rev=ft(length(ft):-1:1);

m=length(fq);

l=1;

while (ft(l)<(ma/2))

left=l;

l++;

endwhile

l=1;

while (ft\_rev(l)<(ma/2))

right=l;

l++;

endwhile

right=length(ft)-right;

lw = fq(right)-fq(left);

if (lw >= 3.66)

C = "pass";

else

C = "fail";

endif

end

function [f,FTnonoi,lw] = fotr(N,res\_f,sw,P) % N - length of the FID, res\_f - resonance freqeuncyof the observed signal, sw - spectral width; P - percentage of left out values of the FID

C = "fail";

while (C == "fail")

[fidnoi,fid] = gen\_fid(N,res\_f,sw,P);

fnoi\_zf=[];

fnoi\_zf=[fidnoi zeros(1,15\*length(fidnoi))];

f\_zf=[];

f\_zf=[fid zeros(1,15\*length(fid))];

f=(-sw/2:sw/(length(fnoi\_zf)-1):sw/2);

FT=real(fftshift(fft(fnoi\_zf)));

FTnonoi=real(fftshift(fft(f\_zf)));

y=1;

while (y<length(f))

if (f(y)<(res\_f-25.0))

left=y;

elseif (f(y)<(res\_f+25.0))

right=y;

else

break;

endif

y++;

endwhile

noise=[FTnonoi(1:left) FTnonoi(right:length(f))];

offset=mean(noise);

FTnonoi=FTnonoi-offset;

[C,lw] = check\_res(f,FTnonoi);

endwhile

end

s\_ampl=[]; %signal amplitude

n\_ampl=[]; %noise amplitude

n\_mean=[]; %noise mean

n\_std=[]; %noise standard deviation

LW=[]; %line width

dens = 10;

res\_frq=350.0;

for z=1:100

noi=[];

[f,FT,lw] = fotr(512,res\_frq,2500.0,dens);

left\_end=0;

right\_begin=0;

m=length(f);

y=1;

while (y<m)

if (f(y)<(res\_frq-25.0))

left\_end=y;

elseif (f(y)<(res\_frq+25.0))

right\_begin=y;

else

break;

endif

y++;

endwhile

noi=[FT(1:left\_end) FT(right\_begin:length(FT))];

LW=[LW lw];

s\_ampl=[s\_ampl max(real(FT))];

n\_ampl=[n\_ampl max(real(noi))];

n\_mean=[n\_mean mean(real(noi))];

n\_std=[n\_std std(real(noi))];

endfor

%printf("Line-width: %.2f +/- %.2f\n", mean(LW), std(LW))

%printf("Signal amplitude: %.2f +/- %.2f\n", mean(s\_ampl), std(s\_ampl))

%printf("Noise amplitude: %.2f +/- %.2f\n", mean(n\_ampl), std(n\_ampl))

%printf("Noise standard deviation: %.2f +/- %.2f\n", mean(n\_std), std(n\_std))

%printf("S/N: %.2f \n", mean(s\_ampl)/mean(n\_ampl))

%plot(f(5007:5477),FT(5007:5477))

output=[f(4607:5477)' FT(4607:5477)'];

save("-text", strcat("2peaks.dat"), "output");