

Contents

- [SECTION: INITIALIZATION](#)
- [New Plot](#)

```
%scalingovr.m
% scaling and overflow

% Copyright 2011
% Wayne T. Padgett
% Last Modified -
```

SECTION: INITIALIZATION

```
clear all; % cleans up first
close all;

fignum = 0;
pwr_threshold = -8;

[speech,fs] = wavread('the pipe began to leak 2.wav');
x1(1) = 0.95*speech+speech.^2*0.1*pi; % create a version with unquantized content

x1(2) = randn(10^4,1);
sourcetxt = {'speech' 'Gaussian'};

wdl_list = [16];
satwraptxt = {'wrap' 'sat' };

%Klist = 0.01:0.001:2;
Klist = logspace(-2,1,500);
verbose = 0;

plotnum = 0;

for sourceind = 1:length(x1);
    for wdlind = 1:length(wdl_list);
        for satlwrap0 = 0:1
```

New Plot

```
        plotnum = plotnum+1;

        fbits = wdl_list(wdlind);

        descrtxt(plotnum) = [sourcetxt(sourceind) ' ' num2str(fbits) ' bits; ' satwraptxt(satlwrap0+1) ];
        this_plot = descrtxt(plotnum)
        abovepwyet = 0;

        for Kindex = 1:length(Klist)
            x = x1[sourceind]*Klist(Kindex);
            [out,saterr,qerr] = q2(x,fbits,satlwrap0);

            sigpwr(plotnum) = sum(x.^2)/length(x);
            satpwr(plotnum) = sum(saterr.^2)/length(x);
            qpwr(plotnum) = sum(qerr.^2)/length(x);
            errpwr(plotnum) = sum((out-x).^2)/length(x);
            SNRdB(Kindex,plotnum) = 10*log10(sigpwr(plotnum)/errpwr(plotnum));
            SNRsat(Kindex,plotnum) = 10*log10(sigpwr(plotnum)/satpwr(plotnum));
            SNRq(Kindex,plotnum) = 10*log10(sigpwr(plotnum)/qpwr(plotnum));
            NormNoise(Kindex,plotnum) = 10*log10(errpwr(plotnum)/sigpwr(plotnum));

            pk = 1; % max representable, not max sign value:: max(abs(x));
            CF(Kindex,plotnum) = 10*log10(pk^2/sigpwr(plotnum));

            if ~abovepwyet && (NormNoise(Kindex,plotnum) > pwr_threshold)
                verbose = 1;
                abovepwyet = 1;
            else
                verbose = 0;
            end;

            if verbose,
                drawnow
                if sourceind == 1, % speech
                    h = msgbox('Playing Original Unquantized','Info');
                    player = audioplayer(x(1:min(length(x),3*fs)),fs,16);
                    playblocking(player);
                    close(h)

                    msgtxt = ['Playing Quantized ' char(descrtxt(plotnum)) ' CF: ' num2str(CF(Kindex,plotnum)) ' NormNoise: ' num2str(NormNoise(Kindex,plotnum))];
                    h = msgbox(msgtxt,'Info');
                    player = audioplayer(out(1:min(length(x),3*fs)),fs,16);
                    playblocking(player);
                    close(h)
                end % if sourceind

                titletxt = ['Quantized ' char(descrtxt(plotnum)) ' CF: ' num2str(CF(Kindex,plotnum)) ' NormNoise: ' num2str(NormNoise(Kindex,plotnum))];

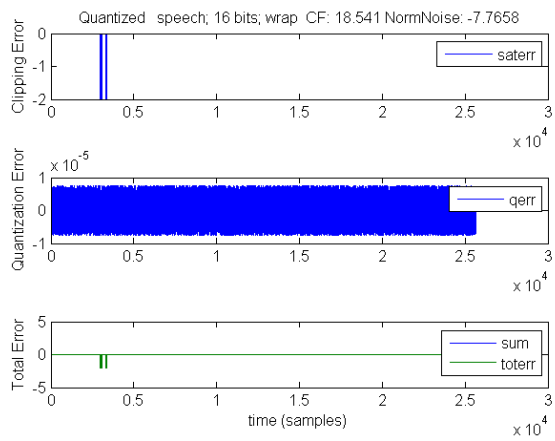
                marker(:,plotnum) = [CF(Kindex,plotnum) NormNoise(Kindex,plotnum)];
                % tmarker(:,plotnum) = [CF(Kindex,plotnum) (NormNoise(Kindex,plotnum)+(plotnum)*6)]; % scoot each text line down a little to avoid overlap
                tmarkerstep = [6 18 12 24]; % fix the labels manually
                tmarker(:,plotnum) = [CF(Kindex,plotnum) (NormNoise(Kindex,plotnum)+tmarkerstep(plotnum))]; % scoot each text line down a little to avoid overlap
                markertxt(plotnum) = {'CF: ' num2str(CF(Kindex,plotnum))}, {'NormNoise: ' num2str(NormNoise(Kindex,plotnum))};

                fignum = fignum+1;
                figure(fignum)
                nn = 1:length(x);
                subplot(311)
                plot(nn,saterr)
                title(titletxt)
                ylabel('Clipping Error')
                legend('saterr')
                subplot(312)
                plot(nn,qerr)
                ylabel('Quantization Error')
                legend('qerr')
                subplot(313)
                plot(nn,saterr+qerr,nn,out-x)
                legend('sum','toterr')
                ylabel('Total Error')
                xlabel('time (samples)')
                drawnow
            end % if verbose

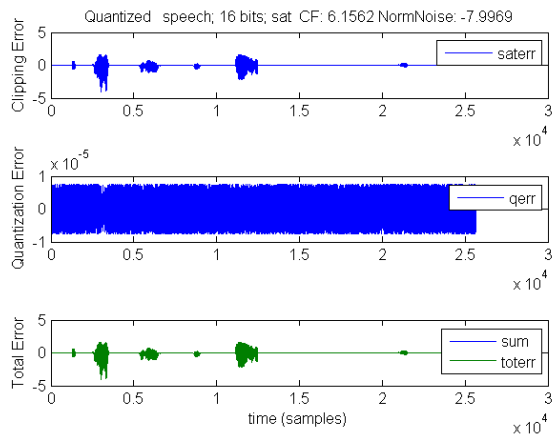
        end; % for Kindex

    this_plot =

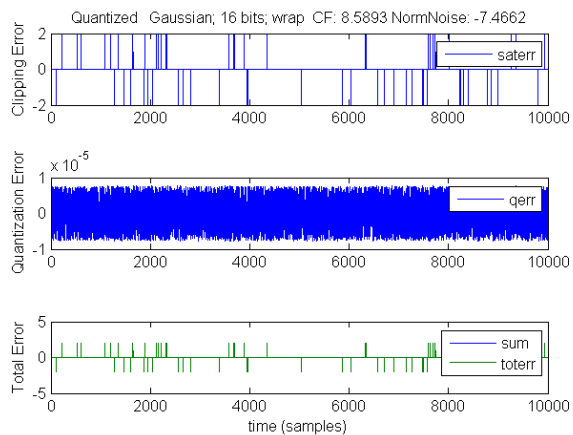
speech; 16 bits; wrap
```



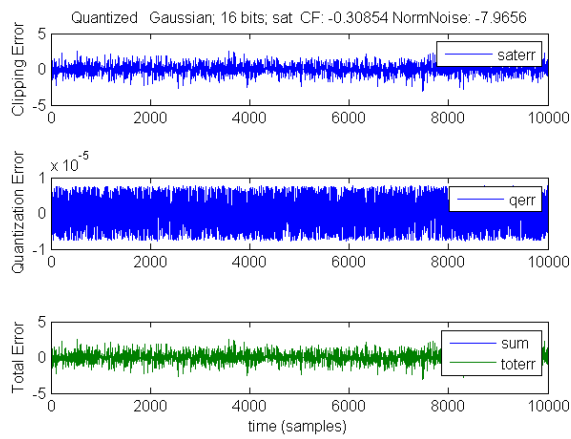
```
this_plot =  
speech; 16 bits; sat
```



```
this_plot =  
Gaussian; 16 bits; wrap
```



```
this_plot =  
Gaussian; 16 bits; sat
```



```

end; % for satlwrap0
end; % for wdlind
end; % for sourceind

% figure(2)
% plot(CF,SNRdB,'-o',CF,SNRsat,'--s',CF,SNRq,':v')
% legend('SNRtot','SNRsat','SNRq')

fignum = fignum + 1;
h = figure(fignum);
scrsz = get(0,'ScreenSize'); % left, bottom, width, height
set(h,'OuterPosition',[1 1 scrsz(3)*0.8 scrsz(4)]) % fill up the screen

plot(CF, NormNoise)
legend(descrtxt)
hold on
plot(marker(1,:),marker(2,:), 'o')
for i = 1:plotnum
    text(tmarker(1,i),tmarker(2,i),markertxt(i));
end
hold off
title('Comparison of Overflow behavior for wrapping and saturation with two different input waveforms')
xlabel('Crest Factor (dB)')
ylabel('Noise Power normalized against signal power (db)')

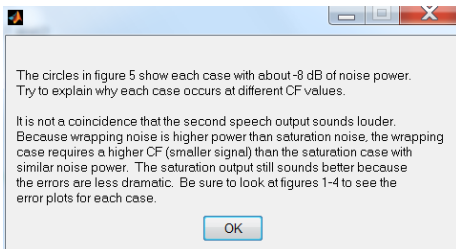
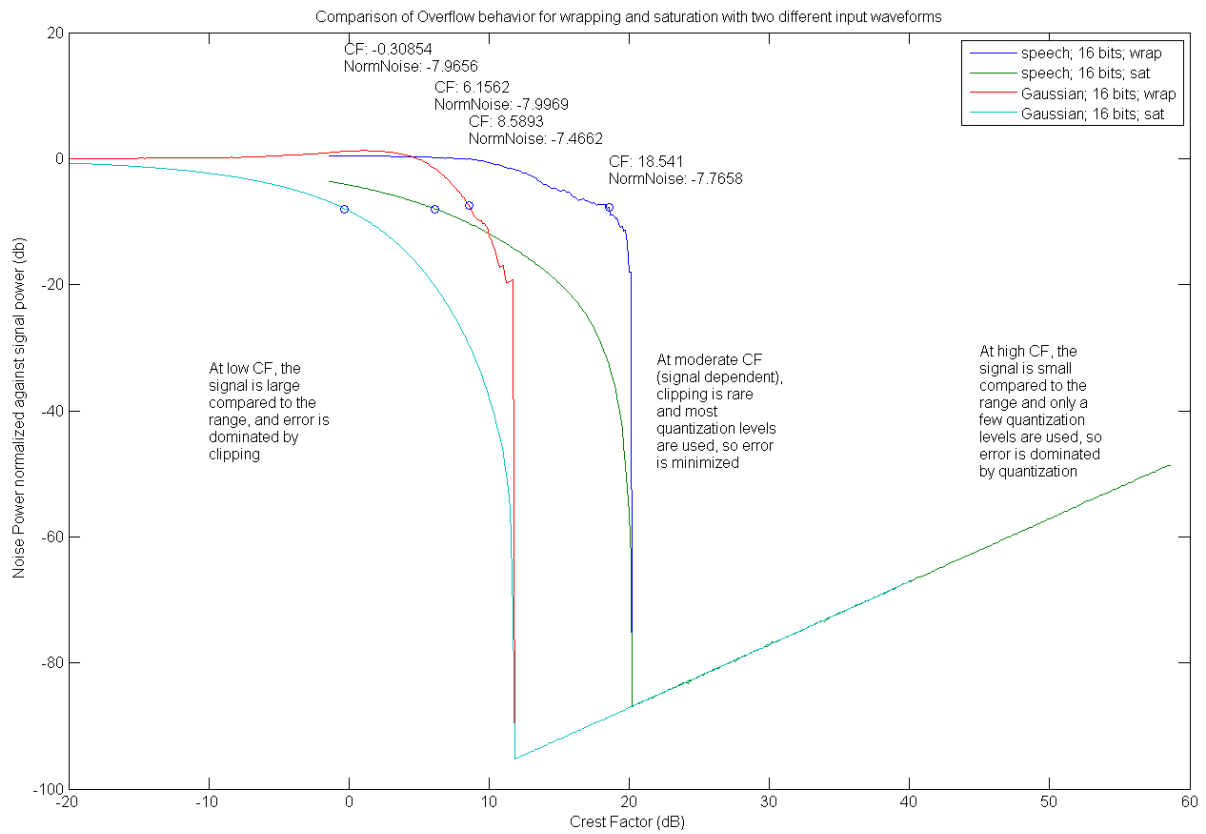
msg1 = 'At low CF, the signal is large compared to the range, and error is dominated by clipping';
msg2 = 'At moderate CF (signal dependent), clipping is rare and most quantization levels are used, so error is minimized';
msg3 = 'At high CF, the signal is small compared to the range and only a few quantization levels are used, so error is dominated by quantization';

msg1 = textwrap([],(msg1),20);
msg2 = textwrap([],(msg2),20);
msg3 = textwrap([],(msg3),20);

text(-10,-40,msg1)
text(22,-40,msg2)
text(45,-40,msg3)

msgbox(['The circles in figure 5 show each case with about -8 dB of noise power. Try to explain why each case occurs at different CF values.', ...
    ', 'It is not a coincidence that the second speech output sounds louder. Because wrapping noise is higher power than saturation noise, ', ...
    'the wrapping case requires a higher CF (smaller signal) than the saturation case with similar noise power. The saturation output still ', ...
    'sounds better because the errors are less dramatic. Be sure to look at figures 1-4 to see the error plots for each case.']));

```



Published with MATLAB® 7.10