18 October 2020 2 ربيع الأول 1442

وَمَا أُوتِيتُم مِّنَ الْعِلْمِ إِلَّا قَلِيلًا

Ain Shams University – Faculty of Engineering – EECE Dept. – Integrated Circuits Lab.
Dr. Hesham Omran

Analog Integrated System Design
Lab 01
Sampling and Quantization in MATLAB

Part 1: Sampling and Windowing

The MATLAB code shown below plots normalized FFT of a sinusoidal test signal sampled by an ideal uniform sampler. The MATLAB code ensures that coherent testing condition is satisfied to avoid spectral leakage.

Study the MATLAB code very well and make sure you understand every line.

1. Report the output plot of the included MALTAB code.
    a. What is the power of the peak signal (in dBFS)?
    b. How many bins are occupied by the test signal?
    c. What is the noise floor (in dBFS)?
    d. If the sampling is ideal, what is the source of error that causes the noise floor?
2. Change the no. of cycles to intentionally violate the coherent testing condition (Hint: Check the MATLAB code comments). Report the new output plot.
    a. What is the power of the peak signal (in dBFS)?
    b. How many bins are occupied by the test signal?
    c. What is the noise floor (in dBFS)?
    d. If the sampling is ideal, what is the source of error that causes the noise floor?
3. Repeat the previous two questions while applying a Blackman Harris window (Hint: Check the MATLAB code comments).

```matlab
% Ref.: B. Boser EECS 247 (Berkeley)
% Edited by H. Omran (ASU)
clear all; close all;
% Sampling frequency. Ts = 1/fs
fs  = 4e6;
% Full-scale input amplitude
Afs = 1;
% No. of samples (This should be power of 2)
N   = 2^7;
% Input frequency
fin_required  = 250e3;
% No. of cycles Ncyc should be prime no. & gcd(Ncyc,N)=1
% Smallest prime Ncyc with gcd(Ncyc,N)=1 is 3
Ncyc = N*fin_required/fs;
if Ncyc < 3
    Ncyc = 3;
else
    Ncyc_primes = primes(Ncyc);
    Ncyc = Ncyc_primes(find(gcd(Ncyc_primes,N)==1, 1, 'last' ));
end
% Force spectral leakage (for illustration)
% Ncyc = Ncyc + 0.5;
% Actual input frequency after selecting Ncyc to be prime (coherent testing
% condition)
fin = fs*Ncyc/N;

% Time vector. N steps from 0 to (N-1)*Ts
t = linspace(0, (N-1)/fs, N);
% Sinusoidal input signal. Afs_rms = Afs/sqrt(2).
y = Afs * cos(2*pi*fin*t);
% Window, for non-coherent sampling
% sig_window = window(@blackmanharris, length(y))';
% sig_window = window(@hanning, length(y))';
% normalize the window function
% sig_window = sig_window/sqrt(var(sig_window)*2);
% Apply the Window to the signal (comment next line for coherent sampling)
% y = y.*sig_window;
% From energy theorem: Afs_rms = sqrt(2)*Afft/N => Afft = Afs*N/2
% To normalize spectrum w.r.t. sine wave peak: divide by Afft
% spectral density
s = 20 * log10(abs(fft(y)/N/Afs*2));
% Drop redundant half
s = s(1:N/2);
% Normalized frequency vector
f = (0:length(s)-1) / N;

% plot additional cycle to show continuity/discontinuity
t2 = t+max(t)+1/fs;
```

```matlab
subplot(2, 1, 1);
% plot two cycles to show continuity/discontinuity
plot(t, y, 'b',t2, y, 'r', [t(end) t2(1)], [y(end) y(1)], 'r');
% plot(t, y, 'b');
xlabel('Time');
ylabel('Amplitude');
axis tight;

subplot(2, 1, 2);
plot(f, s, 'rx'); hold on; plot(f, s, 'k'); hold off;
xlabel('Frequency  [f / f_s]');
ylabel('Magnitude  [dBFS]');
axis tight;
```

Part 2: Quantization

The MATLAB code shown below plots normalized FFT of a sinusoidal test signal sampled by an ideal uniform sampler **and quantized by an ideal B-bit ADC**.

Study the MATLAB code very well and make sure you understand every line.

1. Report the output plot of the included MALTAB code.
   a. Do you notice distortion components? Why?
   b. Calculate the SNR analytically and compare it with the SNR computed by MATLAB.
   c. Calculate the noise floor analytically and compare it with noise floor in the MATLAB plot.
   d. How much is the SFDR? Why?
2. Change the no. of cycles to satisfy the coherent testing condition (Hint: Check the MATLAB code comments). Report the new output plot.
   a. Do you notice distortion components? Why?
   b. Calculate the SNR analytically and compare it with the SNR computed by MATLAB.
   c. Calculate the noise floor analytically and compare it with noise floor in the MATLAB plot.
   d. How much is the SFDR? Why?
3. Compare the SFDR of the two cases. Comment.

```
% Ref.: B. Boser EECS 247 (Berkeley)
% Edited by H. Omran (ASU)
clear all; close all;
% Sampling frequency. Ts = 1/fs
fs  = 4e6;
% No. of samples (This should be power of 2)
N   = 2^12;
% No. of cycles Ncyc should be prime no. & gcd(Ncyc,N)=1
% fs / fin = N / Ncyc must be non-integer to ensure white quantizatoin
% noise
% Bad choice of Ncyc
Ncyc = 2^7;
% Good choice of Ncyc
% Ncyc = 2^7 + 1;
fin = fs*Ncyc/N;

% Time vector. N steps from 0 to (N-1)*Ts
t = linspace(0, (N-1)/fs, N);
% Sinusoidal input signal
y = cos(2*pi*fin*t);

% Quantization
% ADC with B bits and +/-1 full scale
B = 10;
delta = 2/(2^B-1);
thresholds = -1+delta/2:delta:1-delta/2;

yq = zeros(length(y), 1);
for i=1:N
    % quantize, scale, and remove DC
    yq(i) = sum(y(i) >= thresholds) * delta - 1;
end

% From energy theorem: Afs_rms = sqrt(2)*Afft/N => Afft = Afs*N/2
% Afs = full-scale input = 1
% To normalize spectrum w.r.t. sine wave peak: divide by Afft
% spectral density
s = abs(fft(yq)/N*2);
% Drop redundant half
s = s(1:N/2);
% Normalized frequency vector
f = (0:length(s)-1) / N;

sig_bin = N * fin/fs + 1;
Psignal = 20*log10(s(sig_bin))
sn = s;
sn(sig_bin) = 0;
Pnoise = 10*log10(sum(sn.^2))
SNR = Psignal-Pnoise
SFDR = Psignal - max(20*log10(sn))

figure;
plot(f, 20*log10(s), '-');
xlabel('Frequency  [f / f_s]');
ylabel('Amplitude  [dBFS]');
title(sprintf('N = %d    Psig = %.1fdBFS   SNR = %.1fdB   SFDR = %.1fdB', ...
    N, Psignal, SNR, SFDR));
axis([min(f) max(f) -120 10]);
```