Qawi Harvard – ECE615 Mixed Signal Design – HW4

5.1 Develop an expression for the effective number of bits in terms of the measured signal-to-noise ratio if the input sinewave has a peak amplitude of 50% of (V_{REF+} - V_{REF-}).

Let's begin with writing out what our sinewave is:

$$v_{in}(t) = V_p \sin\left(\frac{2\pi}{T}t\right)$$

Now let's go through the process of finding its' root mean square (RMS) value:

$$V_{in,RMS} = \sqrt{\frac{1}{T}\int_0^T {v_{in}}^2\left(\frac{2\pi}{T}t\right)dt} = \sqrt{\frac{1}{T}\int_0^T V_p^2 \sin^2\left(\frac{2\pi}{T}t\right)dt}$$

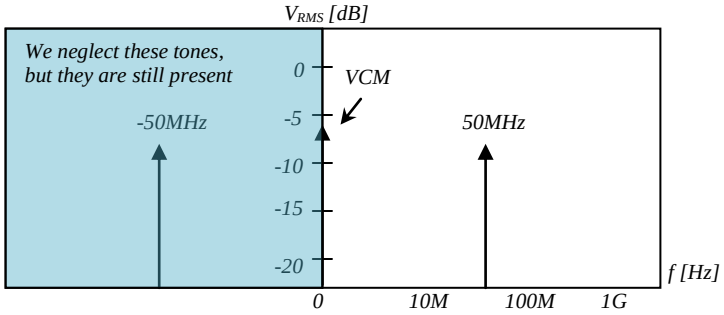
$$= V_p\sqrt{\frac{1}{T}\int_0^T \frac{1-\cos\left(\frac{4\pi}{T}t\right)}{2}dt} = V_p\sqrt{\frac{1}{T}\left(\frac{1}{2}t - \sin\left(\frac{4\pi}{T}t\right)\right)\Big|_0^T}$$

$$= V_p\sqrt{\frac{1}{T}\left(\frac{1}{2}T - 0\right) - \frac{1}{T}\left(\frac{1}{2}0 - 0\right)} = \frac{V_p}{\sqrt{2}} = V_{in,RMS}\ [V]$$

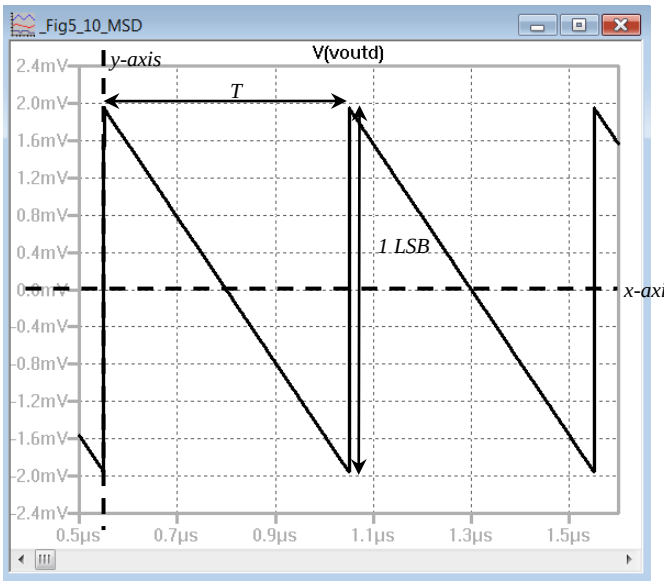
Knowing this, let us plot the power spectrum density of a sinewave centered on a common mode voltage of 0.5V, a peak voltage of 0.4V at 50MHz (values chosen arbitrarily):

$$20\log 0.4 \cong -8dB$$

$$20\log 0.5 \cong -6dB$$



Now, let's find our RMS value of the quantization noise. Figure 5.10 (in the book) shows a simulation that allows us to plot the quantization noise.



To find the RMS value of the quantization noise we can use the equation of a line to describe the saw-tooth wave seen above.

$$y = mx + b$$

$$V_{Qe}(t) = \frac{-LSB}{T}t + \frac{LSB}{2}$$

$$V_{Qe,RMS} = \sqrt{\frac{1}{T}\int_0^T \left(\frac{LSB}{2} - \frac{LSB}{T}t\right)^2 dt}$$

$$\sqrt{\frac{1}{T}\int_0^T \left(\frac{LSB^2}{4} - \frac{LSB^2}{T}t + \frac{LSB^2}{T^2}t^2\right)dt} = \sqrt{\frac{1}{T}\left(\frac{LSB^2}{4}t - \frac{LSB^2}{2T}t^2 + \frac{LSB^2}{3T^2}t^3\right)\Big|_0^T}$$

$$\sqrt{\frac{1}{T}\left(\frac{LSB^2}{4}T - \frac{LSB^2}{2T}T^2 + \frac{LSB^2}{3T^2}T^3\right) - (0 - 0 + 0)} = \sqrt{\left(\frac{LSB^2}{4} - \frac{LSB^2}{2} + \frac{LSB^2}{3}\right)}$$

$$\sqrt{\left(\frac{3LSB^2}{12} - \frac{6LSB^2}{12}T + \frac{4LSB^2}{12}\right)} = \frac{LSB}{\sqrt{12}} = V_{Qe,RMS}\ [V]$$

Now we can determine our signal to noise ratio (in dB):

$$20\log\frac{V_{in,RMS}}{V_{Qe,RMS}} = 20\log\left(\frac{\frac{V_p}{\sqrt{2}}}{\frac{LSB}{\sqrt{12}}}\right)$$

Where:

$$LSB = \frac{(V_{REF+} - V_{REF-})}{2^N}, \qquad V_p = \frac{(V_{REF+} - V_{REF-})}{2}$$

$$20\log\left(\frac{\frac{V_p}{\sqrt{2}}}{\frac{LSB}{\sqrt{12}}}\right) = 20\log\left(\frac{\sqrt{12}\frac{(V_{REF+} - V_{REF-})}{2}}{\sqrt{2}\frac{(V_{REF+} - V_{REF-})}{2^N}}\right) = 20\log\left(\frac{2^N\sqrt{12}}{2\sqrt{2}}\right)$$

$$20\log\left(\frac{2^N\sqrt{12}}{2\sqrt{2}}\right) = 20\log\left(2^N\sqrt{12}\right) - 20\log\left(2\sqrt{2}\right) = 20\left(\log 2^N + \log\sqrt{12} - \log 2 - \log\sqrt{2}\right)$$

$$= 20\left(N\log 2 + \frac{1}{2}\log 12 - \log 2 - \frac{1}{2}\log 2\right) = 6.02N + 1.76\ [dB]$$

This result relates the signal to noise ratio (SNR) to the number of bits in a data converter. If our input signal is a sinewave with a peak voltage equal to 50% of the ADC's reference range and we measure the SNR we can determine our effective number of bits:

$$SNR_{meas} = 6.02N_{eff} + 1.76 \rightarrow N_{eff} = \frac{SNR_{meas} - 1.76}{6.02}$$

This result is summarized on page 173 equations 5.10 – 5.14, this solution provides an additional verification of the equations.

5.2 When using Eq. (5.14) what is the assumed ADC input signal? Put your answer in terms of the ADC reference voltages.

Solution:

Let's get started with copying Eq. (5.14) from [1] to here, Eq. 1.

$$N_{eff} = \frac{SNR_{meas} - 1.76}{6.02} \tag{1}$$

N_{eff} is the effective number of bits, aka ENOB, of an analog-to-digital converter (ADC). It specifies the actual resolution of an ADC. In this equation, SNR_{meas} is the signal-to-noise ratio (SNR) from the measurement result. As we can see from this equation, N_{eff} goes up (higher resolution of the ADC) as SNR_{meas} increasing. The assumed input signal for Eq. 1, or Eq. (5.14) in [1], to be valid is a sinewave that ranges from V_{REF+} to V_{REF-}, Fig. 1.

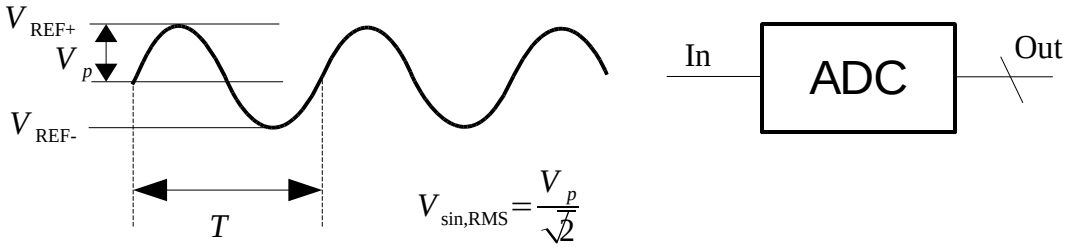


Figure 1 A sinewave input to an ADC

The reason why the input is assumed to be a sinewave is when deriving Eq. 1, we use the SNR calculated for a sinewave, or

$$SNR = 20 \cdot \log \frac{\text{Input RMS Voltage}}{\text{Noise RMS Voltage}} = 20 \cdot \log \frac{\overbrace{V_p/\sqrt{2}}^{\text{RMS voltage of a sinewave}}}{\underbrace{V_{LSB}/\sqrt{12}}_{\text{RMS voltage of quantization noise}}} \tag{2}$$

where

$$V_{LSB} = \frac{V_{REF+} - V_{REF-}}{2^N} \tag{3}$$

and N is the number of bits of the ADC. We assume that the quantization noise RMS voltage, $V_{LSB}/\sqrt{12}$ is same regardless the input signal. The RMS voltage of the input signal, however, may not be $V_p/\sqrt{12}$ if the input is not a sinewave. Let's assume the input is a sawtooth wave that ranges from V_{REF+} to V_{REF-}, Fig. 2, and calculate the SNR for this input signal.

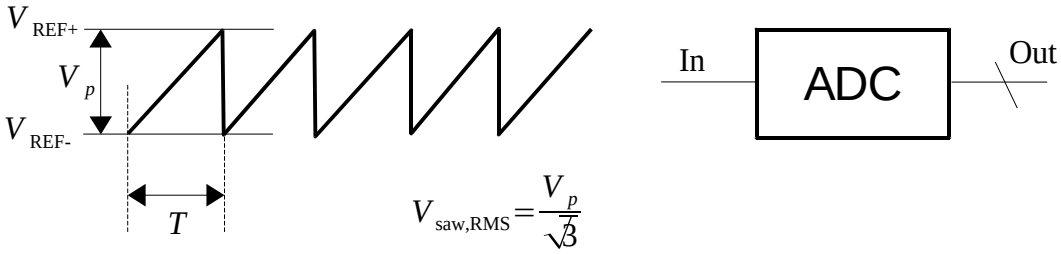


Figure 2 A sawtooth wave input to an ADC

First, we need to calculate the RMS voltage of the sawtooth wave

$$\begin{aligned} V_{saw,RMS} &= \sqrt{\frac{1}{T}\cdot\int_0^T \left(\frac{V_p}{T}\cdot t\right)^2 dt} \\ &= \sqrt{\frac{1}{T}\cdot\int_0^T \frac{V_p^2}{T^2}\cdot t^2\, dt} \\ &= \sqrt{\frac{V_p^2}{T^3}\cdot\frac{1}{3}\cdot\left(T^3-0\right)} \end{aligned} \tag{4}$$

or

$$V_{saw,RMS}=\frac{V_p}{\sqrt{3}} \tag{5}$$

Noticing that for the sawtooth wave we have defined

$$V_p = V_{REF+} - V_{REF-} \tag{6}$$

and using Eq. 3 we can calculate SNR as

$$\begin{aligned} SNR &= 20\cdot\log\frac{V_p/\sqrt{3}}{V_{LSB}/\sqrt{12}} \\ &= 20\cdot\log\frac{(V_{REF+}-V_{REF-})\cdot\frac{1}{\sqrt{3}}}{\frac{(V_{REF+}-V_{REF-})}{2^N}\cdot\frac{1}{\sqrt{12}}} \end{aligned} \tag{7}$$

or

$$SNR = 20\cdot\log 2^N\cdot 2 = 6.02\,N + 6.02 \quad \text{(in dB)} \tag{8}$$

If the SNR is from the measurement result we can manipulate Eq. 8 to get the effective number of bits, or

$$N_{eff}=\frac{SNR_{meas}-6.02}{6.02} \tag{9}$$

Which is different from Eq. 1 or Eq. (5.14) in [1].

Reference:

[1] R. J. Baker, *CMOS Mixed-signal Circuit Design, Second Edition*, Wiley-IEEE, 2009.

5.3) ***Describe, in your own words, the difference between specifying SNR and SNDR.***

Sol) The signal-to-noise ratio (SNR) of a data converter can be described as taking the ratio of the RMS signal voltage to the RMS quantization noise.

Signal-to-noise plus distortion ratio (SNDR) is measured similar to the SNR but, here the output includes the quantization error along with the distortion tones arising from the nonlinearities and mismatches in practical ADCs. SNDR is the ratio of the RMS signal voltage to the RMS quantization noise + distortion.

Measurement of Signal-to-noise plus distortion (SNDR) using SPICE:

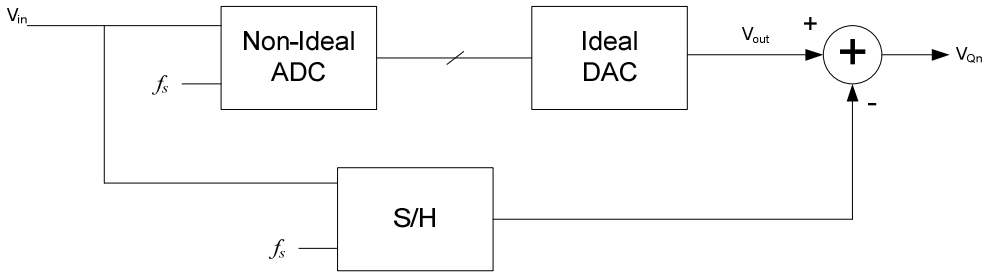


Figure 1: Block diagram for determining the quantization noise

The practical ADC quantizes the input signal and introduces quantization error along with distortion. A non-ideal ADC can be created in SPICE by changing the gains of the amplifiers in the ADC. In order to see the quantization noise introduced by the non-ideal ADC, we need to remove the tones of the input signal and its aliases. In order to remove those, we subtract the output of the DAC from a S/H version of the input signal.

ADC is basically a S/H system with a sinc-response and a quantizer to generate the output bits. When the input signal is passed through it, in frequency domain the input signal tones are generated at f_{in} , fs-f_{in}, fs+f_{in} … these are cancelled out by the tones generated by the ideal S/H input signal and all that is left in V_{Qn} is just the quantized noise and distortions.

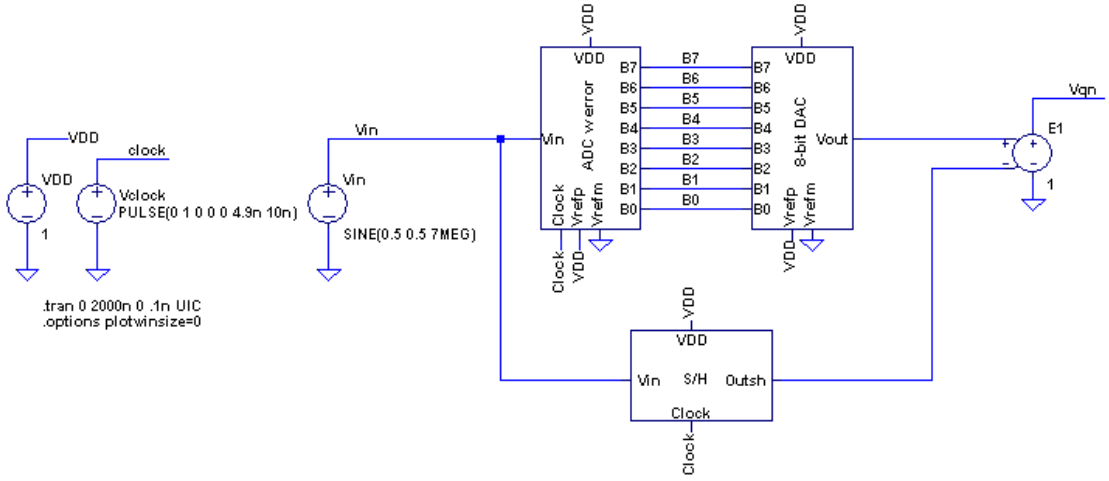


Figure 2: SPICE implementation to measure the quantization noise (with non-ideal ADC to get SNDR)

Here, our input signal is a 0.5V sinusoid centered on 0.5V with a frequency of 7MHz. In order to cancel out the input signal tones and its aliases at 7MHz, 93MHz and 107MHz from the output of the DAC we subtract the input S/H signal to see the quantization noise with distortions from the non-ideal system. The FFT of the output signal is shown below in figures 3 and 4. We see that there are no tones or aliases of the input signal. Similar to measuring the SNDR, the SNR of the system is measured using the above technique but we should also cancel out any spikes or spurious responses. This zeroing of the spurious noise is done by using ideal components in SPICE, where the gains of the amplifiers in the ADC are ideal. The FFT of the ideal output signal is shown in figures 5 and 6.

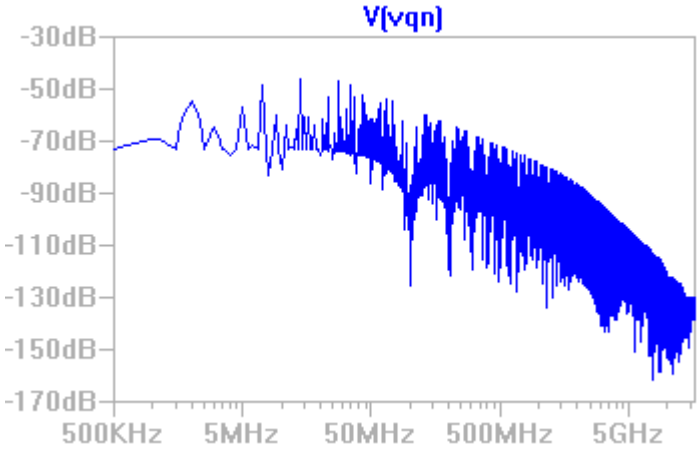


Figure 3:Noise floor of the output signal with Non-Ideal ADC

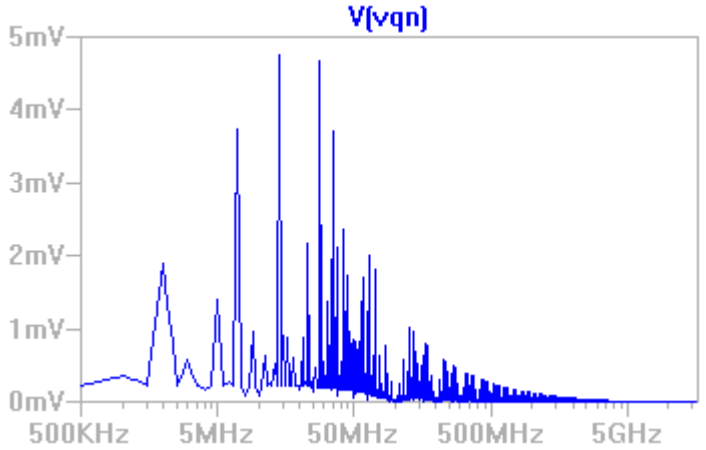


Figure 4: Linear Plot of the output signal with Non-Ideal ADC

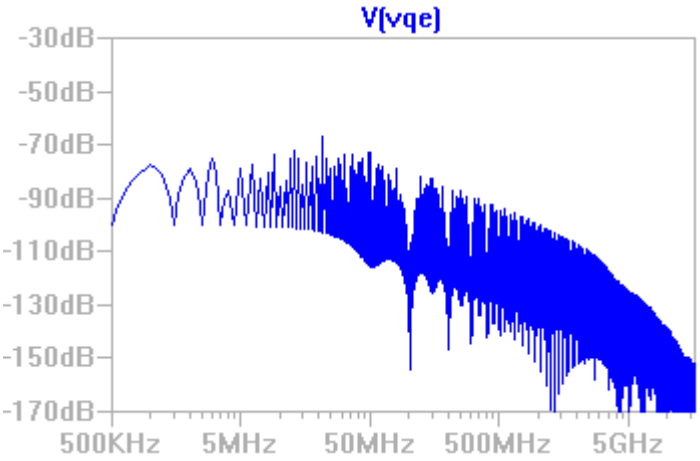


Figure 5: Noise floor of the output signal with Ideal ADC

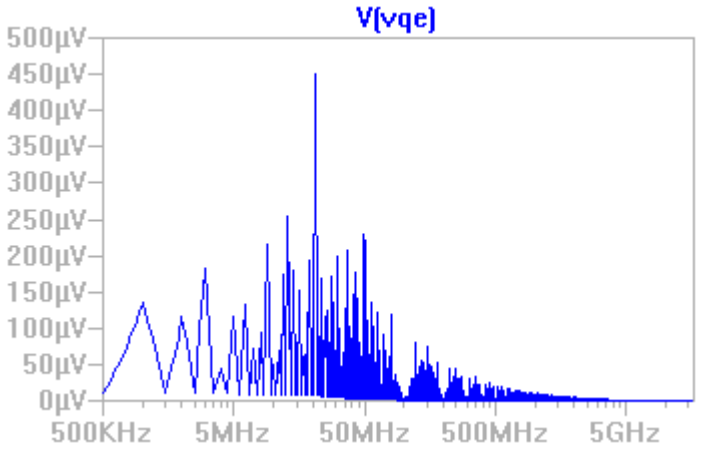


Figure 6: Linear Plot of the output signal with Non-Ideal ADC

From figures 3 and 5 we see that the noise floor moves down when we are using ideal components or while measuring SNR. Since SNDR includes distortions arising from the imperfections in the data converter and from other noise sources its considerably less when compared to a SNR. From SPICE simulations, the RMS voltage of the quantized noise plus distortions for the non-ideal ADC is 22.585mV and the RMS voltage of the quanitzed noise while using an ideal DAC is 1.1513mV. The SNDR and SNR are calculated below:

$$SNDR = 20\log\left(\frac{V_{in,RMS}}{V_{Qe+Distortion,RMS}}\right) = 20\log\left(\frac{\frac{V_{Peak}}{\sqrt{2}}}{V_{Qe+Distortion,RMS}}\right) = 20\log\left(\frac{\frac{0.5}{\sqrt{2}}}{22.585\text{mV}}\right) = 23.89dB$$

$$SNR = 20\log\left(\frac{V_{in,RMS}}{V_{Qe,RMS}}\right) = 20\log\left(\frac{\frac{V_{Peak}}{\sqrt{2}}}{V_{Qe,RMS}}\right) = 20\log\left(\frac{\frac{0.5}{\sqrt{2}}}{1.1513\text{mV}}\right) = 49.74dB$$

5.4. Using SPICE simulations with an ideal ADC and DAC, show that how coherent sampling can result in an RMS value of quantization noise larger than what is specified by Eq. (5.3). Comment on the shape of the quantization noise's spectrum.

Solution: To simplify the description of what happens to quantization noise when coherent sampling is done; let us take the mixed-signal system used in the book (Figure 5.13) to determine the quantization noise. Among the components in the system only ADC introduces quantization noise in the system. Hence by removing the input signal and its aliases from the output we can find the quantization noise in the system. A sample and hold circuit is used in the input signal path to mirror the sample and hold used in ADC. The output from this sample and hold circuit is subtracted from the DAC output to get quantization noise. In practical systems we need to adjust the delay in the input signal path to match the delay through the mixed signal system. In this example we assume that this delay is zero, as ideal ADC and DAC are used. To understand the effect of coherent sampling on quantization noise, we need to understand behavior of quantization noise when non coherent sampling is done on the system. In Section 1 let us discuss intuitively what happens to RMS value of quantization noise and its spectrum when non coherent, coherent sampling is done. In the same section let us verify the intuitive analysis using simulation examples from cmsoedu.com.

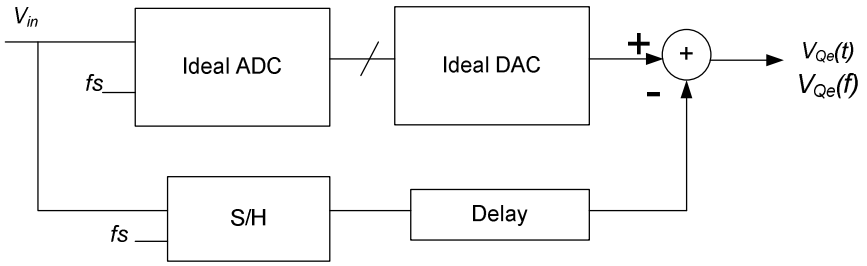


Figure 1. Example system used to estimate the quantization error (Figure 5.13 from book)

Section 1: Intuitive discussion on effect of coherent sampling on RMS value of quantization noise and the spectrum of quantization noise

The quantization noise voltage can be treated as random variable assuming Bennett's criteria hold (refer page 165 in book). The random variable falls between ± 0.5LSB as seen in Figure 2. This means that any value of quantization error between 0.5LSB and -0.5LSB is equally probable.

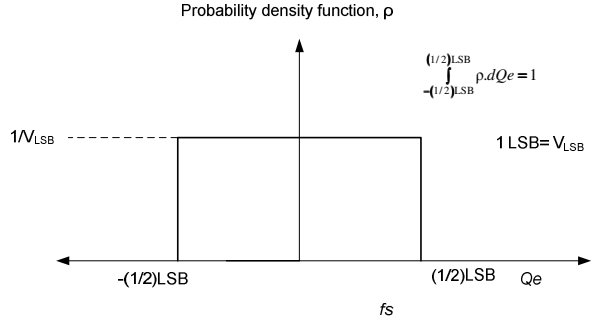


Figure 2. Probability density function for the quantization error in ADC assuming Bennett's criteria hold (Figure 5.11 from book).

Now let us consider the cases of non coherent sampling and coherent sampling.

Non Coherent Sampling:

Analysis:

When the sampling frequency is an non integer multiple of the input signal frequency e.g. fs=100MHz and fin=7MHz then in every cycle of input signal, we sample different points of the input signal and hence the noise is randomized and its probability density function is similar to figure 2. The RMS value of the noise as given from equation 5.3 in the book is

$$V_{Qe,RMS}=\frac{V_{LSB}}{\sqrt{12}} \quad (1)$$

For a 8-bit ADC $V_{LSB}=1/2^8=3.9065\text{mV}$, $V_{Qe,RMS}=1.1276\text{mV}$

Simulation: fin=7MHz, fs=100MHz

Now let us simulate the Figure 1 using examples from cmosedu.com with fs=100MHz and fin=7MHz (frequency of input signal).

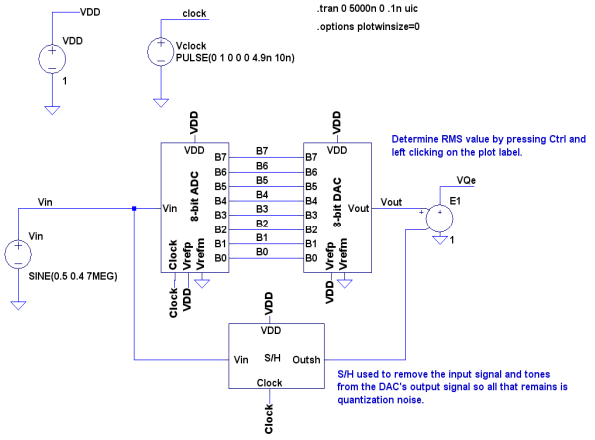
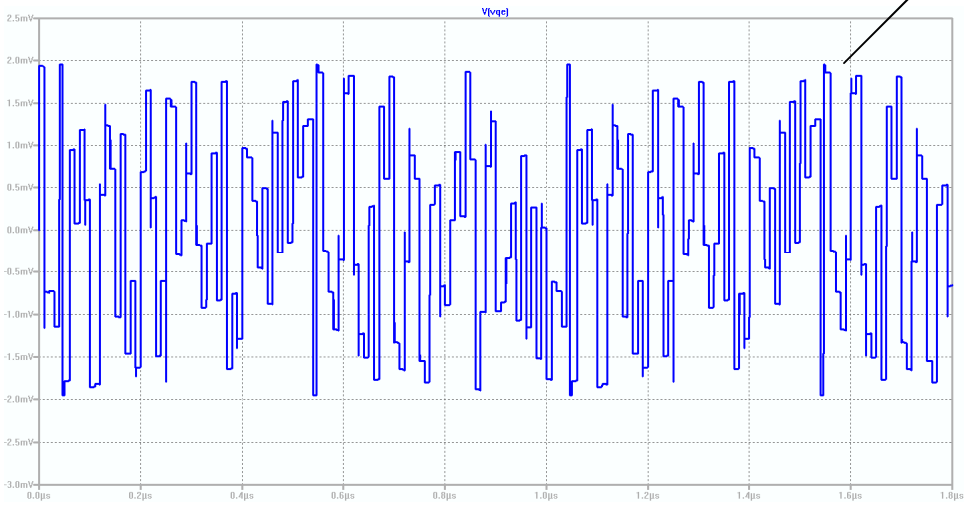


Figure 3. Simulation example of Figure 1 from cmosedu.com



Rms value of the Vqe(t) is 1.1359mV, approximately close to the value predicated from eq 1.

Figure 4. $V_{Qe}(t)$ with fin=100MHz and fs=7MHz

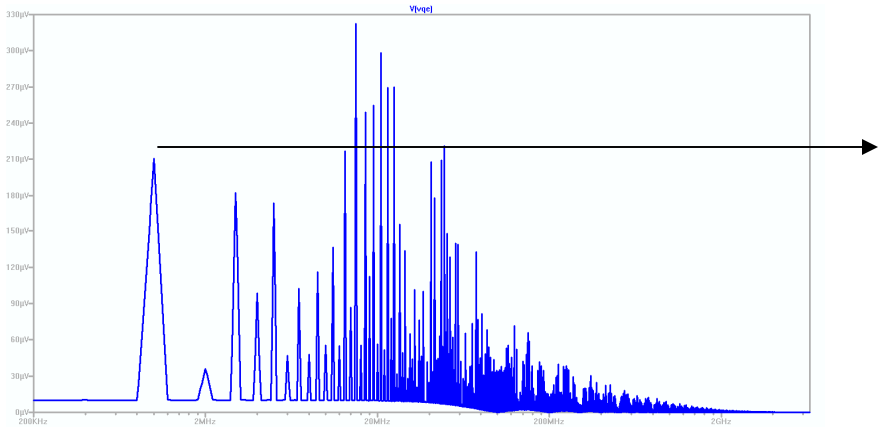


Figure 5. $V_{Qe}(f)$ with fin=100MHz and fs=7MHz (fft plot)

Notice that $V_{Qe}(t)$ is not periodic and $V_{Qe}(f)$ contains tones through out the frequency range.

Coherent Sampling:

Analysis:

When the sampling frequency is an integer multiple of input signal frequency e.g. fs=100MHz and fin=10MHz then in every cycle of input signal, we sample the same group of points of the input signal. Hence the resulting quantization noise values are limited to a group of values. Now the quantization noise is no longer a random variable following the probability density function shown in figure 2. Depending on the input signal points hit in every cycle of the input signal the quantization noise value can be larger or smaller than predicted by equation 1. Since we are concerned only about the cases where the quantization noise gets worse, let us discuss the cases where quantization noise becomes bigger. Since the quantization noise values are measured for the same group of points every cycle $V_{Qe}(t)$ becomes periodic in the time domain, and that implies tones at integer multiples of input frequencies in the frequency domain (fft plot) .

Simulation: fin=10MHz, fs=100MHz

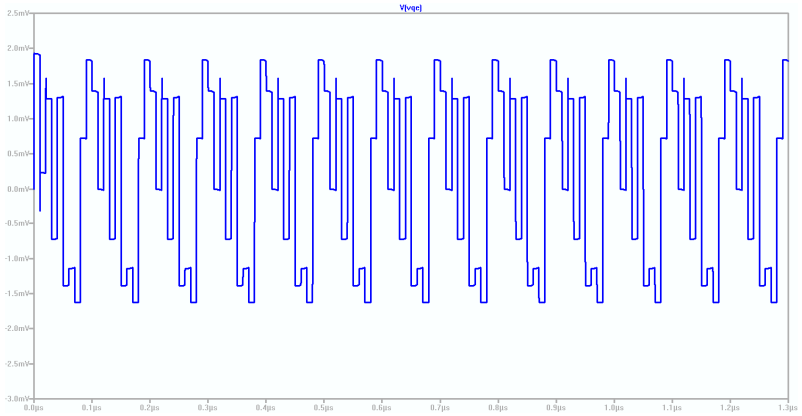


Figure 6. $V_{Qe}(t)$ with fin=10MHz and fs=100MHz

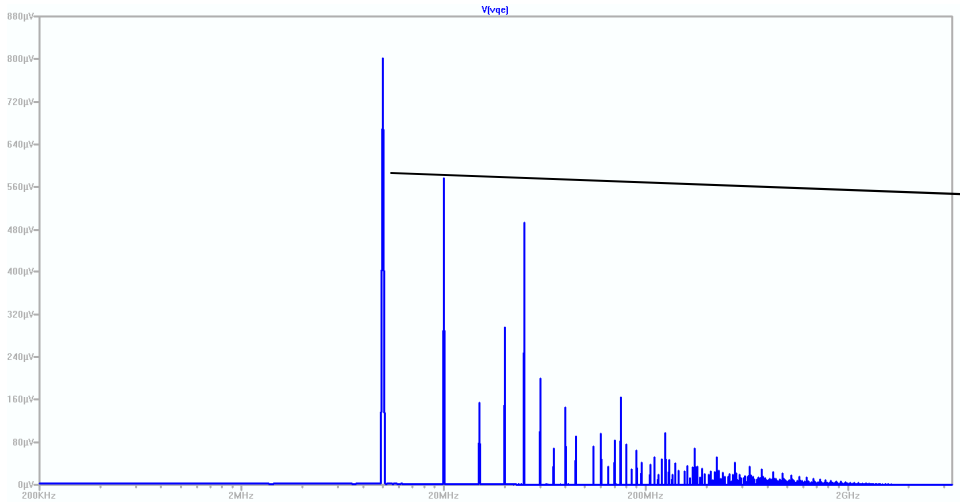


Figure 7. $V_{Qe}(f)$ with fin=10MHz and fs=100MHz (fft plot)

Since at the group of points where the samples are hit in every cycle, the overall quantization noise is bigger than predicted by equation 5.3, the RMS quantization noise is bigger. Notice that $V_{Qe}(t)$ is periodic and $V_{Qe}(f)$ contains tones at 10MHz,20MHz,40MHz etc. Also we can notice that the dominant tones are at 10MHz, 20MHz and till 200MHz.

Simulation: fin=25MHz, fs=100MHz

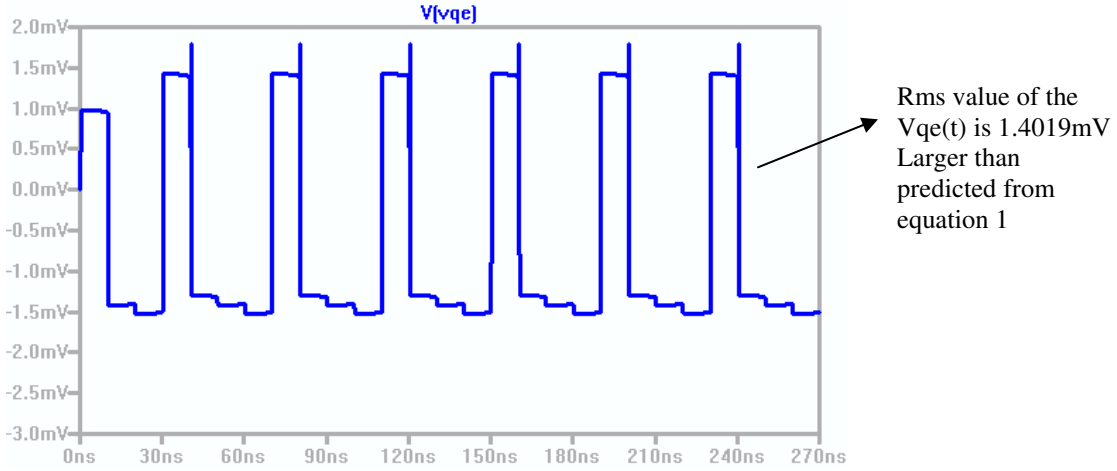


Figure 8. $V_{Qe}(t)$ with fin=25MHz and fs=100MHz

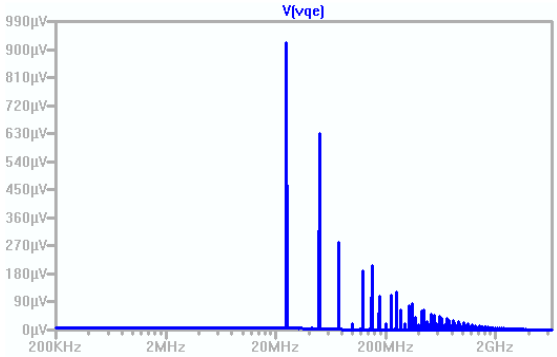


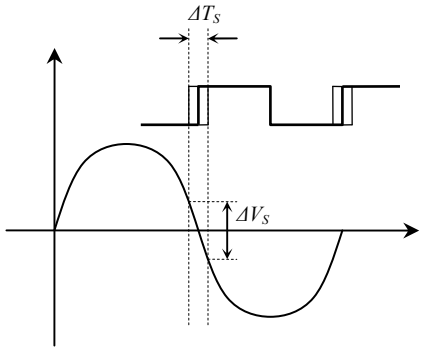
Figure 9. $V_{Qe}(f)$ with fin=25MHz and fs=100MHz (fft plot)

Notice that with fin=25MHz and fs=100MHz $V_{Qe}(t)$ is periodic with frequency=25MHz, and $V_{Qe}(f)$ has tones at integer multiples of input frequencies i.e at 25MHz,50MHz,75MHz etc. The dominant tones in the quantization spectrum are at 25MHz, 50MHz.

Summary:

1. Discussed the model used to estimate the quantization error with an ideal ADC and DAC.
2. Intuitively discussed what happens to RMS values of quantization noise, spectrum of quantization noise with non coherent and coherent sampling. Discussed the case where quantization noise becomes bigger than what is predicted from equation 1. Verified the conclusions arrived with the intuitive discussion with simulation results.

5.5 Suppose a perfectly stable clock is available (ΔT_s is zero in Eq. [5.21]). Would we still have a finite aperture window if the clock has a finite rise time? Describe why or why not?



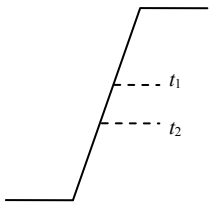
F-1 Clock uncertainty causing an error in the sampled input signal

If we want to determine the worst case error in the sampled input signal due to clock uncertainty, we sample the input signal when it is changing the fastest, or $f_{in} = f_s/2$ ($t = 1/f_{in}$).

$$\frac{d}{dt}\left(V_p \sin \pi f_s t\right) = \pi f V_p$$

$$\frac{\Delta V_s}{\Delta T_s} = \pi f V_p$$

Reviewing F-1 shows that if we have a perfectly stable sampling clock ($\Delta T_s = 0$) then our sampling error (ΔV_s) is also zero. This is only the case when we have an infinite rise time on the sampling clock.



F-2 Depicting high and low thresholds for a finite rising edge.

Due to noise, a real switch will have two distinct switching points depending on the random noise present. This noise could be from power supply noise or simply thermal noise. F-2 shows how a finite rising edge will generate two distinct switching thresholds due to random noise effects.

If we let:

$$t_1 - t_2 = \Delta T_s$$

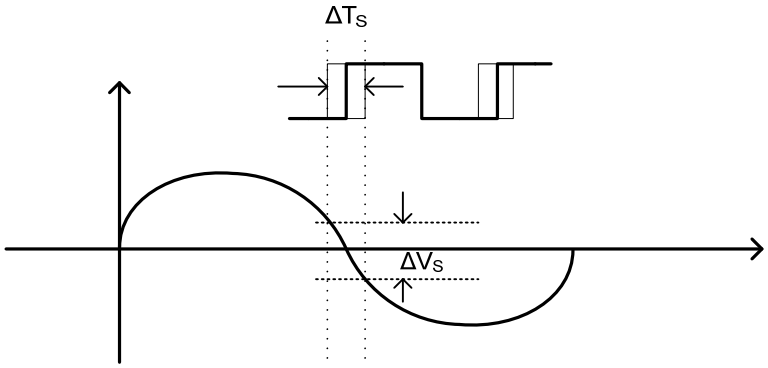
Then we can repeat the derivation:

$$\frac{d}{dt}\left(V_p \sin \pi f_s t\right) = \pi f V_p$$

$$\frac{\Delta V_s}{\Delta T_s} = \pi f V_p$$

ΔT_s is now a function of the finite rise time. The slower our clock edge, the more affect noise will have on the switching.

Kaijun Li
Problem 5.6
How do the number of bits lost because of aperture jitter change with the frequency of an ADC input sinewave? If the ADC input is a DC signal, is aperture jitter a concern? Why?



Solution:
First off, we want to determine the largest variation of the sampled voltage due to the aperture jitter, which happens when the sampling frequency is twice the input sinewave's frequency or fs=2fin,

$$\frac{\Delta V_S}{\Delta T_S} = \frac{d}{dt}\left(V_p \sin 2\pi f_{\text{in}} t\right) = 2\pi f_{in} V_p$$

It is also known that

$$\Delta V_S \le 0.5LSB = \frac{1}{2^{N+1}}\left(V_{REF+} - V_{REF-}\right) \text{ and } V_p = \left(V_{REF+} - V_{REF-}\right)/2$$

Then we have

$$\Delta T_S \ge \frac{1}{2^N} \cdot \frac{1}{2\pi f_{in}}$$

We can model it as the number of bits lost because of aperture jitter N_{loss}, and the maximum clock jitter time instant is rewritten as

$$\Delta T_S = \frac{1}{2^{N-N_{loss}}} \cdot \frac{1}{2\pi f_{in}}$$

Or the number of bits lost can be found as

$$2^{N-N_{loss}} = \frac{1}{2\pi f_{in}\Delta T_S}$$

$$\left(N - N_{loss}\right)\log 2 = -\log\left(2\pi f_{in}\Delta T_S\right)$$

$$N - N_{loss} = -\frac{1}{0.3}\log\left(2\pi f_{in}\Delta T_S\right)$$

$$N_{loss} = N + 3.32\log\left(2\pi f_{in}\Delta T_S\right)$$

To illustrate the effect of input sinewave's frequency over the number of bits lost, N is chosen as 8, and for different values of f_{in} and ΔT_S, N_{loss} is calculated. The MATLAB script is written to obtain the plot of N_{loss} vs. f_{in} for different ΔT_S.

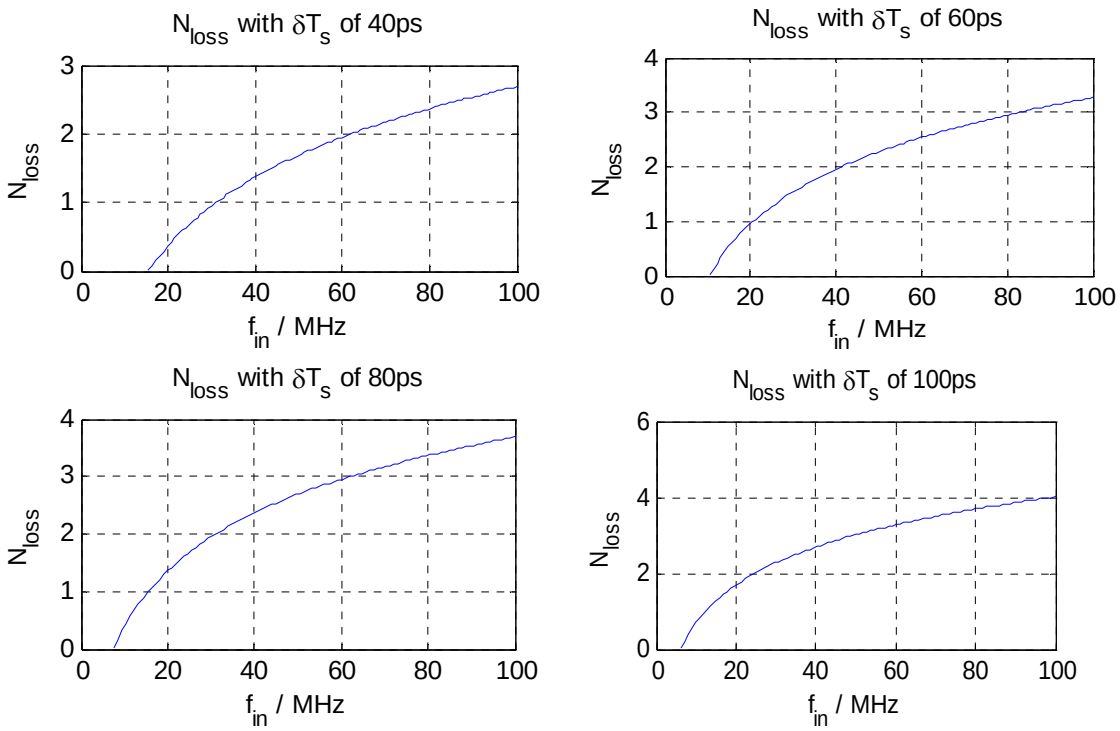


Fig. 1 N_{loss} vs. f_{in} for different ΔT_S

As seen in the above figure, the number of bits lost N_{loss} due to aperture clock jitter goes up as the input frequency increases. It also should be noted that for same input frequency, N_{loss} goes up as ΔT_S goes up. For instance, with f_{in} of 100MHz, N_{loss} changes from about 2.7bits to 4bits when ΔT_S varies from 40ps to 100ps.

So when input signal is a DC signal, there is no any transition time. So aperture jitter is not a concern any more.

```
% MATLAB script to plot the relationship between N_loss and fin
clear all;
close all;
dTs = 40e-12 : 20e-12 : 100e-12; % dTs between 40ps and 100ps
for i = 1:length(dTs)
    N_fin = 100;   % number of points for fin
    fin_min(i) = 1/(2*pi*dTs(i)*256); % when N_loss = 0;
    fin_max = 100e6;      % set maximum fin as 100MHz
    fin(i,:) = linspace(fin_min(i), fin_max, N_fin);
    for j = 1 : N_fin
        Nloss(i,j)=8+3.32*log10(2*pi*dTs(i)*fin(i,j));
    end
    figure;
    plot(1e-6*fin(i,:), Nloss(i,:))
    xlabel('f_{in} / MHz')
    ylabel('N_{loss}')
    title (['N_{loss} with \deltaT_{s} of ' num2str(dTs(i)/1e-12) 'ps']);
    grid on;
end
```

Kaijun Li
Problem 5.7
Why must Bennett's criteria be valid for the averaging filter in Fig. 5.29 to reduce the quantization noise in the digital output signal? Give an example input signal where averaging will not reduce quantization noise.

Solution:
First off, Bennett's criteria states that the signal must be busy, or in other words, the signal needs to "jump around". Then let's find out what happens after the digital codes out of the ADC are passed into the averaging filter, and this will help with understanding why the signal needs to be busy in order to reduce the quantization noise using the averaging filter.

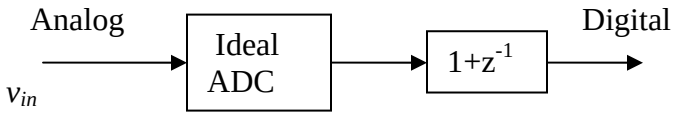
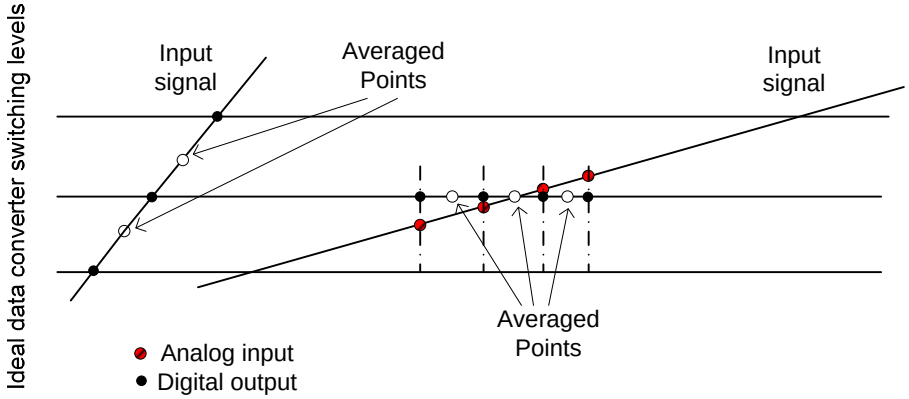


Figure 5.29. Using a digital averaging filter to reduce quantization noise

To illustrate the effect of averaging filter, Figure 1 is drawn. On the left side of Figure 1, there is a busy signal, and it is noted that the averaging points are inserted between adjacent digital output code coming out of the data converter. So effectively, a number of new switching levels are created which in turn reduce the quantization noise.



The signal on the right hand side can be seen as a slow-changing signal (or non-busy signal). Due to the small variation between adjacent points for the analog input signal, the digital output codes do not change. Consequently, the averaged points inserted are the same with the digital codes out of the data converter, which means we don't get any benefit from the averaging filter on quantization noise reduction.

A good example for a "non-busy" signal is DC signal where averaging will not reduce quantization noise.

5.8) ***Assuming Eq. (5.57) is valid, re-derive Eq. (5.13) including the effects of averaging K ADC output samples. Is Eq. (5.13) or the equation derived here valid for a slow or DC input signal? Comment on why or why not.***

Solution:

Equation 5.57 from the text book is the one shown below; it is the resulting RMS quantization noise after the input signal undergoes digitization by the ADC and is averaged by a moving averaging filter.

$$V_{Qe,RMS} = \frac{1}{\sqrt{K}} \cdot \frac{V_{LSB}}{\sqrt{12}} \tag{1.1}$$

Equation 5.13 from the book is the SNR for the output of a ADC without any averager on its output, it is given as,

$$SNR_{ideal} = 6.02N + 1.76 \ (in\ dB) \tag{1.2}$$

In order to solve for the SNR, we need to take the ratio of the input RMS voltage to the quantization RMS voltage as shown in Eq 1.2,

$$SNR = 20\log\left(\frac{V_{in,RMS}}{V_{Qe,RMS}}\right) \tag{1.3}$$

Considering the input to be a busy sine signal, we can write the RMS voltage of the input as

$$V_{in,RMS} = \frac{V_P}{\sqrt{2}} \qquad where,\ V_P = \frac{V_{REF+} - V_{REF-}}{2} \tag{1.4}$$

We know that for an ADC that has a output code with N number of bits, the LSB voltage is given by,

$$V_{LSB} = \frac{V_{REF+} - V_{REF-}}{2^N} \tag{1.5}$$

Substituting Eq. 1.5 in Eq. 1.1 we get,

$$V_{Qe,RMS} = \frac{1}{\sqrt{K}} \cdot \frac{\left(\frac{V_{REF+} - V_{REF-}}{2^N}\right)}{\sqrt{12}} = \frac{1}{\sqrt{K}} \cdot \frac{V_{REF+} - V_{REF-}}{2^N \cdot \sqrt{12}} \tag{1.6}$$

Now let's substitute both the input RMS voltage Eq. 1.4 and RMS quantization error voltage Eq. 1.6 in the Signal-to-Noise ratio equation, we get

$$SNR = 20\log\left(\frac{\frac{\cancel{V_{REF+} - V_{REF-}}}{2\sqrt{2}}}{\frac{1}{\sqrt{K}} \cdot \frac{\cancel{V_{REF+} - V_{REF-}}}{2^N \cdot \sqrt{12}}}\right) = 20\log\left(\frac{2^N \cdot \sqrt{K} \cdot \sqrt{12}}{2\sqrt{2}}\right) = 20\log\left(2^N \cdot \sqrt{K} \cdot \frac{\sqrt{6}}{2}\right) \tag{1.7}$$

$$SNR = 20\log\left(2^N\right) + 20\log\sqrt{K} + 20\log\left(\frac{\sqrt{6}}{2}\right) \tag{1.8}$$

$$SNR = 6.02N + 10 \cdot \log(K) + 1.76 \tag{1.9}$$

Avinash Rajagiri
Spring 2009

This defines the SNR for a data converter employing a digital k-order averaging or low-pass filter.

This equation is not valid for DC or low-frequency signals as we do not get any benefit of increase in SNR for those signals. Because according to the Bennett's criteria, the input signal must be busy or varying to reduce the quantization noise using an averaging filter. Using an averaging filter, two adjacent codes of an ADC are averaged and a point is inserted between them. Adding this extra switching level gives us a reduction in the quantization noise being introduced into the system. It is important to note that the output codes being generated should be linear, as averaging does not make sense if they are not linear. If the input signal is DC, then digital code for the ADC is not changing, so between samples of the input data (which is not changing), we are not changing the output of the ADC by at least 1LSB. Therefore, we do not get any benefit from averaging, so this equation cannot be applied to DC or low frequency signals where the output of the ADC does not change by at least a LSB.

5.9. If Bennett's criteria are valid, does averaging ADC outputs (or DAC inputs) put any restrictions on the bandwidth of the input signal? Why? Give an example.

Sol): We determine the spectral content of the quantization noise assuming that the Bennett's criteria are valid. It means that the following conditions hold.

- The input signal to the ADC is busy i.e. "jumping around", that means no two consecutive outputs of the ADC have the same digital code. We see it is redundant to find the quantization noise at the same digital output.
- The input (to the ADC) signal's amplitude variation falls between V_{REF+} and V_{REF-}. The ADC's LSB is much smaller than input amplitude. If the above two conditions are not met then the output spectrum will have spurs, and hence the quantization noise spectrum is affected.

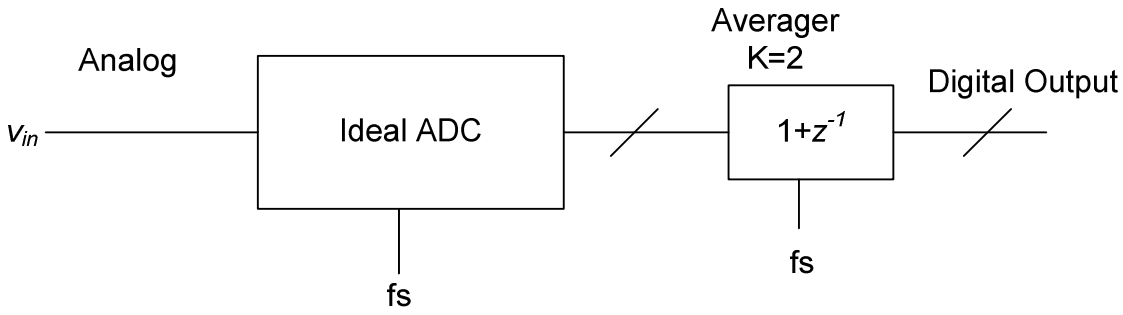


Figure 1. Using digital averaging filter to reduce quantization noise (Fig 5.29 from book)

Using the averaging filter model shown in Figure 1 we get an improved SNR as shown in equation (1).

$$SNR_{ideal} = 6.02N + 1.76 + 10 \log K \quad (1)$$

It is assumed in this model that for the ideal ADC the outputs are linear. If the outputs are not linear then the averaging the outputs can result in the same output. Hence the resolution is not improved. Note that the ideal ADC and the averaging filter are sampled at the same frequency i.e. *fs.* In general a k-bit averaging filter has transfer function H(z) given by equation (2).

$$H(z) = \frac{1 - z^{-k}}{1 - z^{-1}} \quad (2)$$

If fs is the sampling frequency we know z can be represented by

$$z = e^{j2\pi f/fs} \quad (3)$$

From this we can derive

$$|H(z)| = \left| \frac{\sin \frac{k\pi f}{fs}}{\sin \frac{\pi f}{fs}} \right| \quad (4)$$

Let us discuss the averager response when K=2, 4.

Case 1: Averager with K=2

When K=2 the transfer function H(z) reduces to

$$H(z) = \frac{1-z^{-2}}{1-z^{-1}} \; or \; 1+z^{-1} \tag{5}$$

$$|H(z)| = \left| \frac{\sin\frac{2\pi f}{fs}}{\sin\frac{\pi f}{fs}} \right| = 2\left|\cos\frac{\pi f}{fs}\right| \tag{6}$$

Equation (6) implies that we have null points in the magnitude response at integer multiples of fs/2. The magnitude decreases from peak of 2 at DC to zero at fs/2. The ADC has a sinc response because of the sample and hold circuit in it. The nyquist frequency in this case is fs/2 . The averager (or counter) decimates the frequency by K i.e. in this case K=2, the new nyquist frequency is given by

$$f_{nyquist} = (fs/2) \bullet (1/K) = fs/4 \tag{7}$$

The frequency response of the averager with k=2 is shown in Figure 2. Note that the averager response at a frequency fs/2 becomes zero.

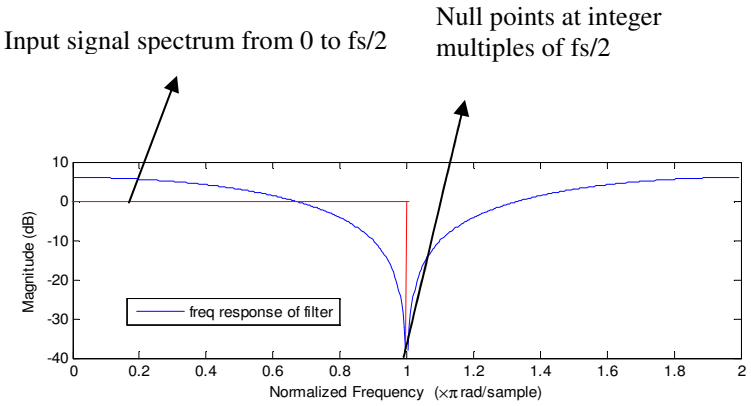


Figure 2. frequency response of averager with K=2

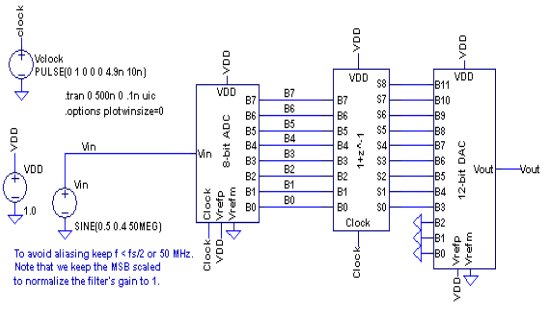


Figure 3. Using Digital Averaging filter with K=2

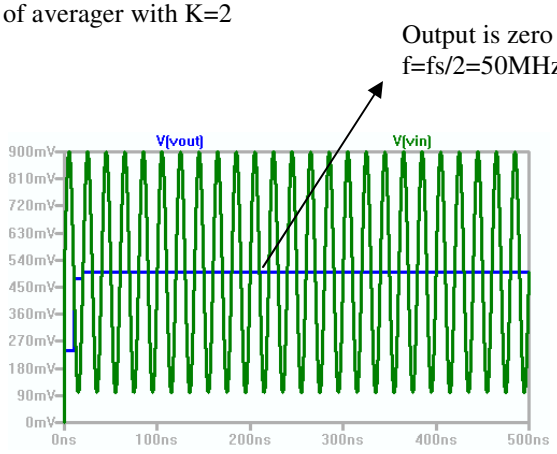


Figure 4. time domain plot at f=fs/2

For The averaging system with K=2 as shown in Figure 3 (reference: cmosedu examples), the output at f=fs/2=50MHz (Figure 4) is zero i.e. $V_{ref}/2$. This means that the allowable input signal bandwidth has become lesser than fs/2.

Case 2: Averager with K=4

From equation (4) we have

$$|H(z)| = \left| \frac{\sin \frac{4\pi f}{fs}}{\sin \frac{\pi f}{fs}} \right| \tag{8}$$

Since the averager decimates the frequency by K=4, in this case the new nyquist frequency is given by

$$f_{nyquist} = (fs/2) \bullet (1/K) = fs/8 \tag{9}$$

The frequency response of the averager with K=4 is shown in Figure 5. Note that the averager response at frequencies at integer multiples of fs/4 becomes zero.

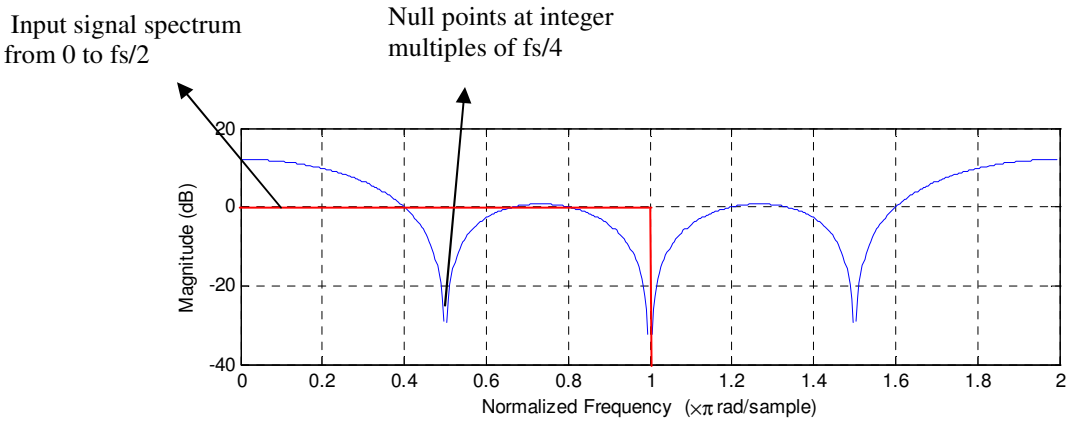


Figure 5. frequency response of averager with K=4

We see that $f_{nyquist}$ is decreasing with increasing value of K of the averager in Figure 1. We conclude that for faithful reproduction of input signal, we need to limit the input signal bandwidth to fs/2K, if we use the model as in figure 1.

Problem 5.10 – How accurate does an 8-bit ADC have to be in order to use a digital filter to average 16 output samples for a final resolution for 10-bits (see eq. 5.59)? Assume the ideal LSB of the 8-bit converter is 10mV.

Since the ideal LSB of the ADC is 10mV, and the ADC is 8-bit, the full input range of the ADC is 256*LSB, or 2.56V. The averaging of 16 bits results in a theoretical increase in resolution of two bits, as shown by equation 5.59, reprinted below (K is the number of averages).

$$N_{increased} = \frac{10log(K)}{6.02} = 2\ when\ K = 16$$

However, in order for the averaging to increase the resolution, the LSB steps must be accurately spacedby less than one LSB, exactly which is given by

$$Accuracy < 0.5LSB\frac{1}{2^{N_{increased}}}$$

For a resolution increase of two bits, (8 bits to 10 bits) with 1LSB = 10mV, the resolution of the converter must be: $0.5LSB\frac{1}{2^2} = 0.125$. This corresponds to an accuracy within 10mV*0.125, or 1.25mV, which is 0.049% of the full scale input range.

Avinash Rajagiri
Spring 2009

5.11) ***Show the detailed derivation of Eq. (5.66).***

Solution: From the text book, equation 5.66 is given as, this is the equation for the output of a feedback modulator shown in Fig. 1, which can be employed to improve the SNR of a data converter system.

$$v_{out}(f) = \overbrace{\frac{A(f)}{1+A(f)\cdot B(f)}}^{\text{Signal transfer function, STF(f)}} \cdot v_{in}(f) + \overbrace{\frac{1}{1+A(f)\cdot B(f)}}^{\text{Noise transfer function, NTF(f)}} \cdot V_{Qe}(f) \qquad (1.1)$$

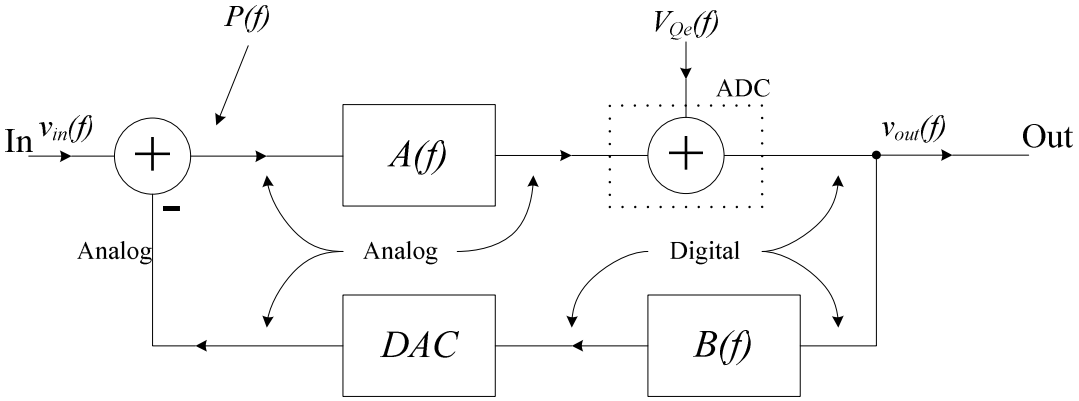


Figure 1: Block diagram of a feedback modulator

To drive the above equation, lets label the output of the first adder block as *P(f)* as shown in Fig.1.
Note that the DAC does not introduce any type of quantization noise unlike the ADC which is modeled as an additive quantization noise.

We can write the output $V_{out}(f)$, as follows,

$$v_{out}(f) = P(f)\cdot A(f) + V_{Qe}(f) \qquad (1.2)$$

Now, lets write the equation for the feedback path of the system, we get

$$P(f) = v_{in}(f) - v_{out}(f)\cdot B(f) \qquad (1.3)$$

We can clearly see now that by substituting Eq. 1.3 in Eq. 1.2, we can obtain a relationship between the input and output of the feedback modulator.

$$v_{out}(f) = \left[v_{in}(f) - v_{out}(f)\cdot B(f)\right]\cdot A(f) + V_{Qe}(f) \qquad (1.4)$$

$$v_{out}(f) = v_{in}(f)\cdot A(f) - v_{out}(f)\cdot B(f)\cdot A(f) + V_{Qe}(f) \qquad (1.5)$$

$$v_{out}(f)\left[1+B(f)\cdot A(f)\right] = v_{in}(f)\cdot A(f) + V_{Qe}(f) \qquad (1.6)$$

$$v_{out}(f) = v_{in}(f)\cdot\frac{A(f)}{1+B(f)\cdot A(f)} + \frac{V_{Qe}(f)}{1+B(f)\cdot A(f)} \qquad (1.7)$$

$$v_{out}(f) = \overbrace{\frac{A(f)}{1+B(f)\cdot A(f)}}^{\text{Signal transfer function, STF(f)}} \cdot v_{in}(f) + \overbrace{\frac{1}{1+B(f)\cdot A(f)}}^{\text{Noise transfer function, NTF(f)}} \cdot V_{Qe}(f) \qquad (1.8)$$

5.12 Summarize, and compare, the advantages and disadvantages of predictive and noise-shaping data converters.

Solution:

Predictive and noise-shaping data converters are all converters that using feedback modulator which can be represented using the block diagram shown in Fig. 1.

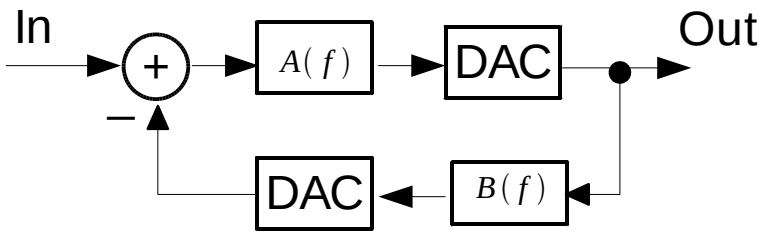


Figure 1 Block diagram of a feedback modulator.

The input and output of the modulator is related as

$$v_{out}(f)=\frac{A(f)}{1+A(f)\cdot B(f)}\cdot v_{in}(f)+\frac{1}{1+A(f)\cdot B(f)}\cdot V_{Qe}(f) \tag{1}$$

For predictive data converters, $A(f)=1$ and the Eq. 1 can be rewritten as

$$v_{out}(f)=\frac{1}{1+B(f)}\cdot v_{in}(f)+\frac{1}{1+B(f)}\cdot V_{Qe}(f) \tag{2}$$

For noise-shaping data converters, $B(f)=1$, and the Eq. 1 can be rewritten as

$$v_{out}(f)=\frac{A(f)}{1+A(f)}\cdot v_{in}(f)+\frac{1}{1+A(f)}\cdot V_{Qe}(f) \tag{3}$$

From Eqs. 2 and 3 we see that the transfer functions of signal and noise in the predictive data converters are the same. However, for noise-shaping data converters, the quantization noise in the low frequencies is pushed to higher frequencies so the signal can be obtained easier.

Since output of a predictive data converter is averaged to zero, so it does not required wide input range of the ADC in the feed forward path. The average value of a noise-shaping data converter, on the other hand, is approached to the input. Since the output of the ADC is averaged by the modulator, it does not have to be very precise. Also, filtering the output of a noise-shaping data converter is easy since the noise has been push to higher frequencies. In contrast, dealing with the output of a predictive data converter requires an analog filter with a transfer function of precisely $B(f)$, which may be a challenging task.