Quantizing a sine wave - N1AL 6/5/2009

Quantizing a complex signal generally results in quantization noise. Quantizing a periodic signal results in quantization spurs - discrete tones on frequencies related to the signal frequency and the sample rate. Let's see how that works with a sine wave.

$fs := 9500$ Sample rate

$f := 1000$ Sine wave frequency

We choose a sample rate that is not an integer multiple of the sine wave frequency so that the spurs don't end up on harmonics only.

$bits := 8$ Resolution

$n := 9500$ Number of samples

$i := 0 .. n-1$ Sample index

By sampling for an integer number of sine-wave cycles, we won't have to window the FFT.

$A_i := \sin\left(2 \cdot \pi \cdot \frac{f}{fs} \cdot i\right)$ Undigitized sine wave

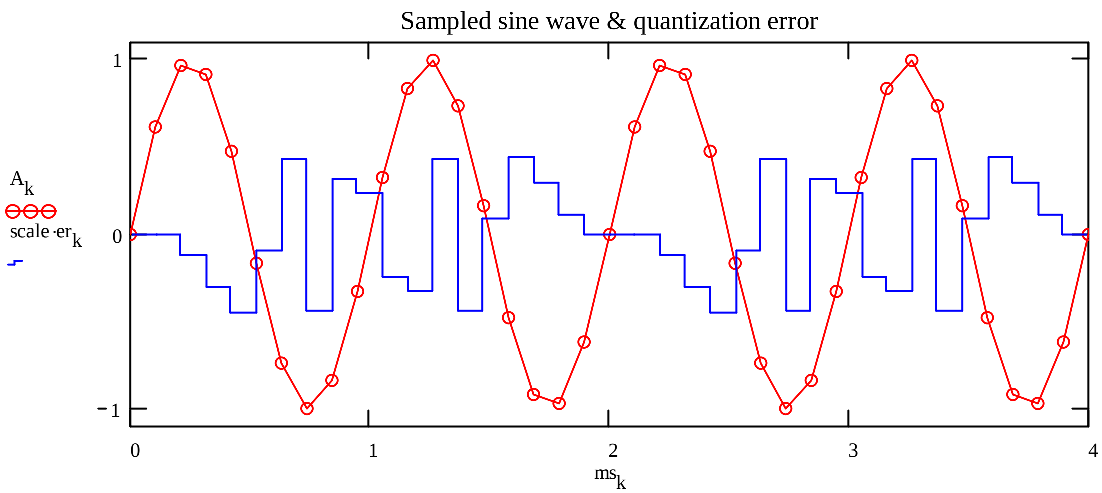
$scale := 2^{bits-1} - 1$ Scale the signal so that the peak is at full scale of the analog-to-digital conversion.

$Adig_i := \dfrac{\text{floor}\left(scale \cdot A_i + 0.5\right)}{scale}$ Digitized sine wave

$er_i := Adig_i - A_i$ The error signal is the difference between the original signal and the quantized signal

$k := 0 .. 100$ Counter for points to plot.

$ms_i := i \cdot \frac{1000}{fs}$ Time in millisconds



Notice how the error signal repeats every two cycles.

Calculate the frequency spectrum using the fast Fourier transform.

The regular FFT functions in Mathcad require that the input be a real sequence of length 2^n. Our sequence is real but of the wrong length so we have to use the "complex FFT" cfft():

$S := \mathrm{cfft}(A)$ Frequency spectrum of undigitized signal.

$Sd := \mathrm{cfft}(Adig)$ Frequency spectrum of digitized signal.

$m := \mathrm{ceil}\left(\frac{n}{2}\right)$

$j := 0 .. m$

The Fourier transform of a real signal gives a symmetrical frequency spectrum. That is, the spectrum from 0 to fs/2 is the mirror image of the spectrum from -fs/2 to 0 (which is the same as the spectrum from fs/2 to fs). So the output of the normal FFT function has only half the samples since the other half is the same anyway. The complex FFT gives the entire spectrum but we will only use half of it since we actually have a real input.

$f\_index := \mathrm{floor}\left(f \cdot \frac{n}{fs} + 0.5\right)$ $norm := \left|S_{f\_index}\right|$ $S_j := \frac{S_j}{norm}$

$normd := \left|Sd_{f\_index}\right|$ $Sd_j := \frac{Sd_j}{norm}$

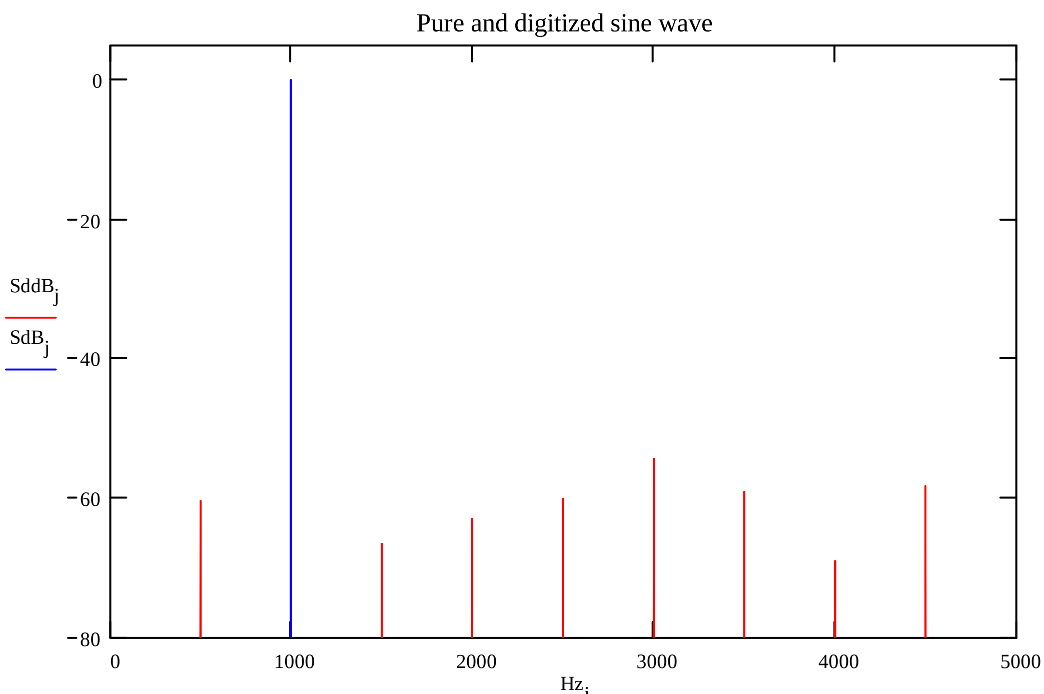
Normalize the spectra so that the value at the sine-wave frequency is at zero dB.

$SdB_j := 20 \cdot \log\left(\left|S_j\right| + 0.000001\right)$

$SddB_j := 20 \cdot \log\left(\left|Sd_j\right| + 0.000001\right)$

Convert frequency spectra into dB. The 0.000001 fudge factor is to prevent taking the log of zero in case some sample is zero.

$Hz_j := j \cdot \frac{fs}{n}$ Frequency in Hz as a function of FFT sample.



Because the error has a period of two sine-wave cycles, the spurs are at f/2 and harmonics.

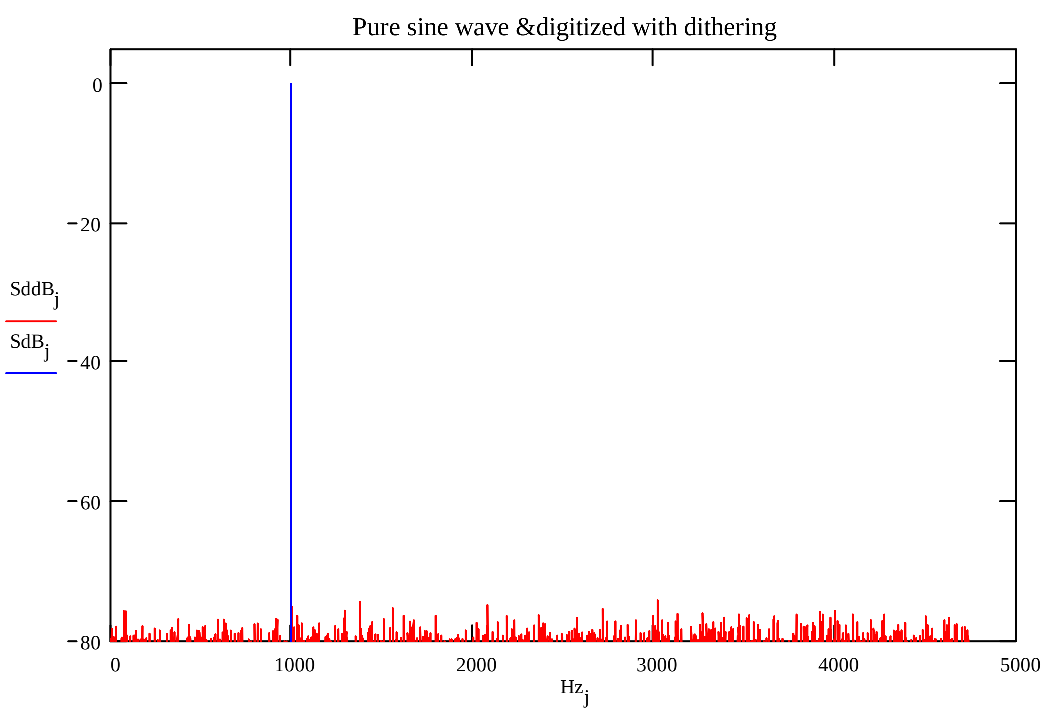
Now let's try dithering. By adding a small amount of noise to the signal before quantization the quantization error is no longer periodic. The quantization spurs are converted to quantization noise, which is preferable in many applications even though the total spurious signal power is greater.

$Adig_i := \dfrac{\text{floor}(\text{scale}\cdot A_i + \text{rnd}(1))}{\text{scale}}$ Add 1 LSB of uniformly-distributed noise.

$Sd := \text{cfft}(Adig)$ Re-calculate the frequency spectrum

$normd := \left| Sd_{f\_index} \right| \quad Sd_j := \dfrac{Sd_j}{norm}$ and re-normalize.

$SddB_j := 20\cdot\log\left(\left| Sd_j \right| + 0.000001\right)$ Convert to dB.



The discrete spurs have been "smeared out" into random noise.