CASPER Memo 012
Consequences of One Bit Quantization on Signal to Noise Ratio

Daniel Chapman

January 2007

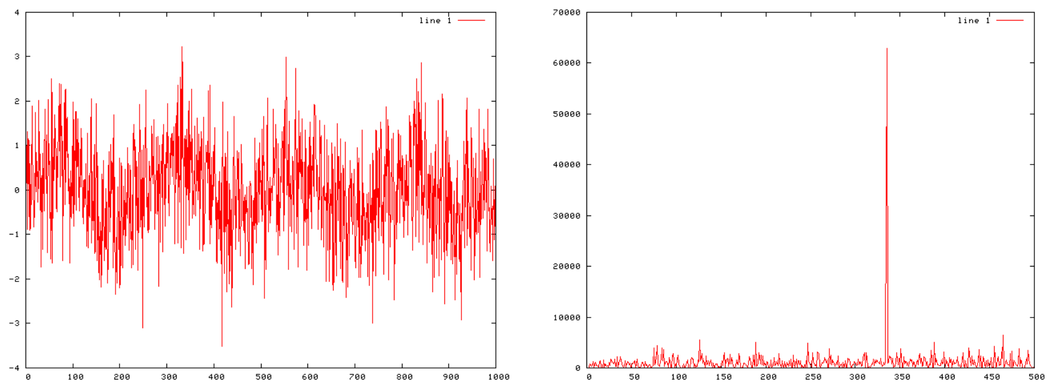
1 Background

1.1 Signal to Noise Ratio (SNR)

Imagine a set of time domain data that is composed of two elements:

- a pure sine wave with RMS amplitude A and frequency f
- and gaussian noise with a standard deviation (or RMS amplitude) of B

If we take the Fourier Transform of this data, we expect the resulting spectrum to be made up noise and one spike located at frequency f. Example:



(a) time domain (pre-fft)

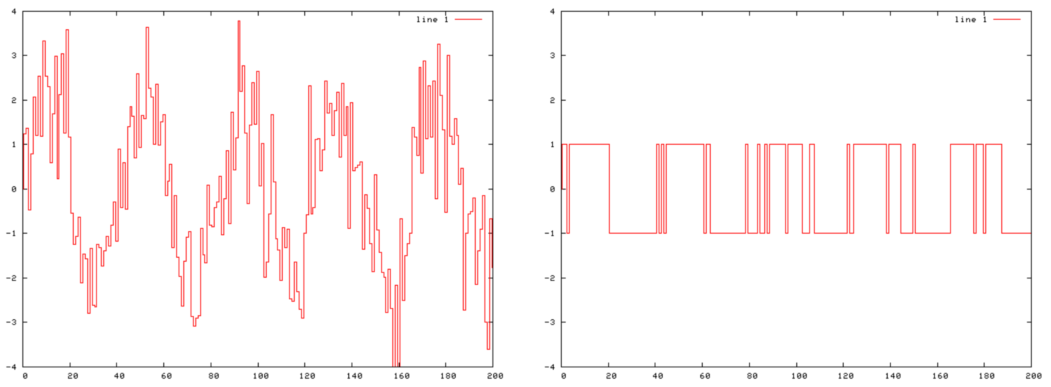
(b) frequency domain (post-fft)

Figure 1: A sine wave with rms amplitude A=0.5 on top of gaussian noise with standard deviation B=1.0

In the frequency domain data, the height of the spike divided by the height of the noise is known as the signal to noise ratio. Naturally, as you increase the strength of the sine wave relative to the gaussian noise (i.e. increase A/B) in your time domain data, you will increase the signal to noise ratio in your frequency domain data.

1.2 Quantization

Now let's imagine what happens when you quantize the input signal into a one bit number, where the bit simply represents whether the signal is positive or negative. Although the data loses a ton of precision, you can still make out the original sine wave (see Figure 2).



(a) time domain data, unquantized

(b) same data, quantized

Figure 2: A sine wave with rms amplitude A=1.4 on top of gaussian noise with standard deviation B=1.0

So, how does quantization affect the signal to noise ratio? When A/B is small, the signal to noise ratio will be low regardless of whether or not the input is quantized. As we slowly increase A/B, the sine wave will become more obvious in both data sets, just as you can identify the sine wave in both plots in figure 2 (although it is a little less clear in the quantized data).

As A/B becomes large, however, the quantized data stops improving. Imagine the difference between input data with A/B=100 and A/B=1000. In both cases, the quantized data will look like a square wave.

In contrast, unquantized data with A/B=1000 will produce a much larger signal to noise ratio than data with A/B=100, because the magnitude of sine wave is preserved. Thus, where the SNR from unquantized data continues to rise as we increase A/B, the SNR from quantized data quickly levels off.

2 Results

How do we quantify this? With a program that loops over A/B, calculating the SNR at each value.

The C code at the end of this document produces the data for the plots on the next page. The X-axis corresponds to A/B (signal RMS amplitude / noise RMS amplitude), and the Y-axis corresponds to the square root of the power of the signal divided by the power of the noise.

The first plot display how the SNR increases for both quantized and unquantized data. The second plot divides the two curves, giving you an idea of where the two curves diverge.

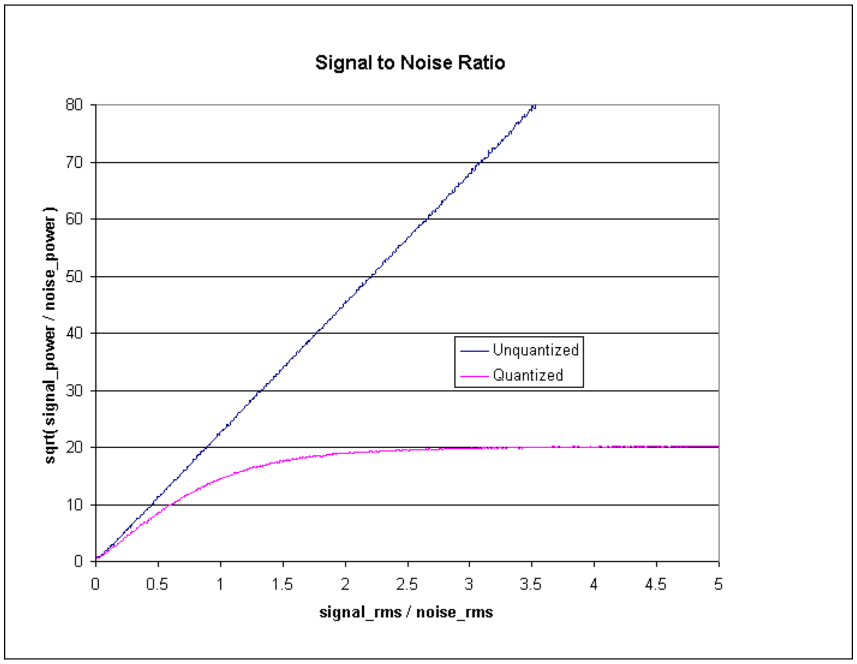


Figure 3: SNR vs. A/B for both quantized and unquantized data

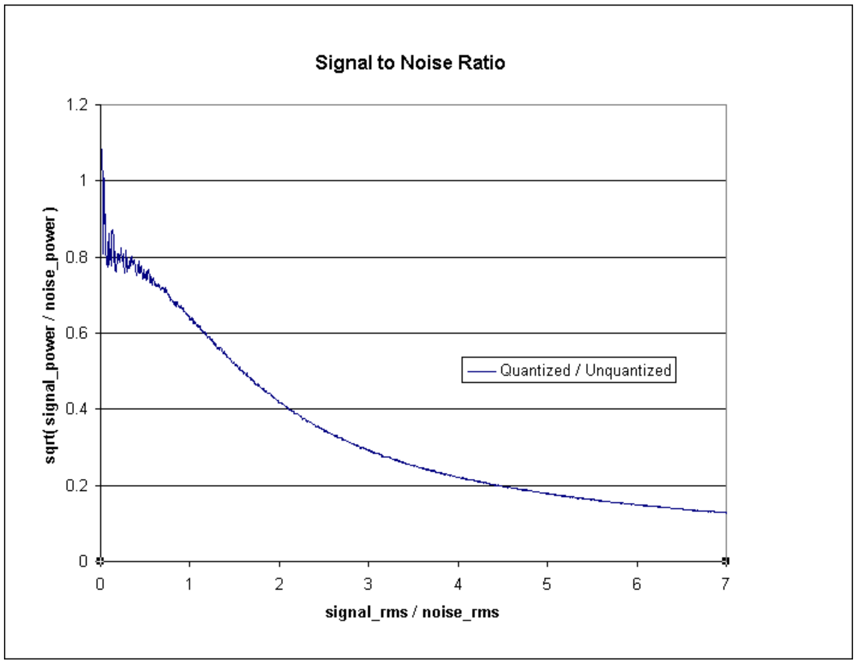


Figure 4: quantized/unquantized curves representing SNR vs. A/B

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

#define ARRAY_SIZE 1000

/* From the GNU Scientific Library, src/randist/gauss.c
   Polar (Box-Mueller) method; See Knuth v2, 3rd ed, p122 */
double rand_uniform(){
        /* return uniform (-1 to 1) */
        double value;
        value = 2.0 * (((double) rand()) / ((double) RAND_MAX)) - 1.0;
        return value;
}
double gsl_ran_gaussian (const double sigma)
{
        double x, y, r2;
        do{
                /* choose x,y in uniform square (-1,-1) to (+1,+1) */
                x = -1 + 2 * rand_uniform();
                y = -1 + 2 * rand_uniform();
                /* see if it is in the unit circle */
                r2 = x * x + y * y;
        }while (r2 > 1.0 || r2 == 0);
        /* Box-Muller transform */
        return sigma * y * sqrt (-2.0 * log (r2) / r2);
}

double quantize(double value){
        if(value < 0)
                return -1.0;
        else
                return 1.0;
}

int main(){
        int i,j,k, num_k_loops;
        double SNR[ARRAY_SIZE];
        double spectrum_i;
        double gauss;
        double A_signal;
        double sum_signal, sum_noise;
        FILE *fp_output;

        srand(time(NULL));
        num_k_loops = 100;
        for(j=0; j<ARRAY_SIZE; j++){
                SNR[j] = 0.0;
        }

        /* Rather than performing an entire fourier transform, we can compute
```

```
           just one frequency component since we know what frequency we expect
           the spike to be in.  To obtain the RMS power of the noise, we simply
           use power_time_domain = power_frequency_domain. */

        for(k=0; k<num_k_loops; k++){
                for(j=0; j<ARRAY_SIZE; j++){
                        A_signal = 10.0*((double) j)/((double) ARRAY_SIZE);
                        sum_signal = 0;
                        sum_noise = 0;
                        for(i=0; i<1024; i++){
                                gauss = gsl_ran_gaussian(1.0);
                                spectrum_i = A_signal*sin(i*0.1)+gauss;
                                /* Comment out the following line for unquantized data */
                                spectrum_i = quantize(spectrum_i);
                                sum_signal += spectrum_i*sin(i*0.1);
                                sum_noise += gauss*gauss;
                        }
                        SNR[j] += (sum_signal*sum_signal) / (sum_noise);
                }
        }
        fp_output = fopen("output_q.txt", "w");
        for(j=0; j<ARRAY_SIZE; j++){
                fprintf(fp_output, "%f\n", sqrt(SNR[j]/((double)num_k_loops)));
        }
        fclose(fp_output);
        return 0;
}
```

References

[1] J.H. Van Vleck and D. Middleton, "The spectrum of clipped noise," *Proceedings of the IEEE*, Volume 54, Issue 1, Jan. 1966, Pages 2-19

[2] Thompson, A. R., James M. Moran, and George W. Swenson. *Interferometry and Synthesis in Radio Astronomy* 2nd ed. New York: Wiley-Interscience, 2001.

[3] F. Drake, J. H. Wolfe, and C. L. Seeger (Editors), SETI Science Working Group Report, NASA Technical Paper 2244, Jan 1984, appendix B, pages 47, 48.