Corrigenda

to the Book:

Quantization Noise

by Bernard Widrow and István Kollár

Cambridge University Press, 2008

Web page: `http://www.mit.bme.hu/books/quantization/`

Last edited: March 12, 2010

Please email remarks and questions to `kollar@mit.bme.hu`

Exercises corrected vs. the book: 3.11, 5.5, 8.7, 15.6, 15.7, 15.9, 15.11.
Corrected parts are marked with black bars.

Exercise 3.11, page 55

3.11 Assume that x has discrete distribution, $P(x = -q/4) = 0.5$, and $P(x = q/4) = 0.5$. Assume furthermore that y has uniform distribution in $(-q/4, q/4)$.

(**a**) Illustrate graphically in the amplitude domain that $x + y$ has uniform distribution in the interval $(-q/2, q/2)$.

(**b**) Calculate the CFs of x and y, and show by using them that $\Phi_{x+y}(u) = \dfrac{\sin(qu/2)}{qu/2}$.

Exercise 5.5, page 109

5.5 The variance of the noise on a quantized sine wave is being determined from $N = 1000$ samples. The noise is assumed to be white, the sine is determined from LS fit (thus having small error). Apply PQN (For these calculations let us momentarily assume that the random component is uniform in $(-q/2, q/2)$). How large is the 95% confidence interval of the measured variance? How large is the difference when normally distributed noise is assumed with the same variance?
Hint: the variance in the measurement of the variance of the variable x is equal to

$$\frac{\operatorname{var}\left\{x_0^2 - \mathrm{E}\{x_0^2\}\right\}}{N} = \frac{\mathrm{E}\{x_0^4\} - (\mathrm{E}\{x_0^2\})^2}{N}\,, \qquad \textit{(E5.5.1-corr)}$$

with $x_0 = x - \mathrm{E}\{x\}$.

Exercise 8.7, page 194

8.7 Derive the general expression of Sheppard's two-dimensional corrections (relate $\mathrm{E}\{x_1^{r_1} x_2^{r_2}\}$ with the moments $\mathrm{E}\{(x_1')^{r_1}(x_2')^{r_2}\}$), using the Bernoulli numbers (see Eq. (4.25.FN1) on page 91).

Hint: prove that

$$\mathrm{E}\{x_1^{r_1} x_2^{r_2}\} = \sum_{m_1=0}^{r_1} \sum_{m_2=0}^{r_2} \binom{r_1}{m_1}\binom{r_2}{m_2}\left(1 - 2^{1-m_1}\right)\left(1 - 2^{1-m_2}\right) \times B_{m_1} B_{m_2} q_1^{m_1} q_2^{m_2} \mathrm{E}\{(x_1')^{r_1-m_1}(x_2')^{r_2-m_2}\}\,, \qquad \text{(E5.5.2)}$$

with B_{m_i} being the Bernoulli numbers.

Last sentence above Subsection 15.7.2, page 389:

This will be true for all frequency indices m, for which $0 < m < N/2$.

page 359:

Let the crossover point be designated by x_c. It can be determined by combining Eq. (14.16) with Eq. (14.14).

$$x_c = \pm\Delta \cdot \frac{q}{q_h} . \tag{14.17}$$

Another equivalent expression for the crossover point can be derived. Using (12.1), we see that

$$\Delta = q_h \cdot 2^p . \tag{14.18}$$

Dividing both sides of Eq. (14.17) by q, and substituting Eq. (12.1) for Δ, we obtain

$$\frac{x_c}{q} = \pm 2^p , \qquad \text{or} \qquad x_c = \pm q \cdot 2^p . \tag{14.19}$$

Thus, when the input to the fixed-point quantizer has the value given by Eq. (14.19) as $x = x_c$, the point of transition between fixed-point behavior and floating-point behavior is reached.[17]

Section 15.7.3, pages 392-393:

15.7.3 DIT FFT with Fixed-Point Number Representation

Quantization noise having power $2q^2/12$ for both real and imaginary parts is generated by the PQN models (Fig. 15.9). This is due to the fractional parts appearing in the additions, during complex multiplications and in the additions after multiplications by the W_N^k terms, e.g. for $N = 16$, $k = 1, 2, 3, 5, 6, 7$.

From the flow graphs of Figs. 15.13 and 15.12, one can see that not all of the FFT outputs have the same amount of quantization noise. For example, for $N = 16$, at zero frequency ($X(0)$), and at half of the sampling frequency ($X(8)$), there is no quantization noise produced by the last butterfly. At the highest frequency (or, in other words, the smallest absolute-value negative frequency), $X(15)$ has quantization noises caused by three PQN's along the way. The output $X(14)$ goes through one layer of quantization.

The output $X(N-1)$ will go through three layers of quantization for $N = 32$, and so forth. The number of such layers is $(\log_2 N) - 2$. All of these layers contribute $2q^2/12$ each in quantization noise power, assuming that after each multiplication, quantization happens. After injection, each noise variance is multiplied by 2

[17] Note that the above definition gives the crossover point as the point from where one can consider the floating-point quantizer as if it acted alone. There is an interval in which both quantizers are valid: $x_c \le x < 2x_c - q/2$. Therefore, for input signals below $2x_c - q/2$, the uniform quantizer can be considered as if it acted alone.

at each stage except the immediately following one (because two erroneous samples are added in each butterfly).

In general, the power of the quantization noise present in both the real and imaginary parts of the highest frequency output will be

$$\begin{pmatrix}\text{quantization noise power}\\ \text{in real and in imaginary}\\ \text{parts of output } X(N-1)\end{pmatrix} = \frac{2q^2}{12}\left(2^{\log_2 N-3}+2^{\log_2 N-4}+...1\right)$$

$$\approx \frac{q^2}{12}\cdot\frac{N}{2}\,. \qquad \textit{(15.25-corr)}$$

For other values of m, the quantization noise will be somewhat less than for $(N-1)$, depending on the actual value of m. For any value of N, the quantization noise power in the FFT outputs at all frequencies can be computed. It can be uniformly bounded as

$$0 \le \begin{pmatrix}\text{quantization noise power}\\ \text{in real and in imaginary}\\ \text{parts of all FFT outputs}\end{pmatrix} \le \frac{q^2}{12}\cdot\frac{N}{2}\,. \qquad (15.26)$$

This is a *loose upper bound* for the average error variance: from Fig. 15.13 it is clear that in the first rounded stage (stage 3) only the lower half of the samples need rounding, in the second one one fourth of them, etc. This means that one can decrease the average variance by 1/4 part of the total sum, and by 1/16 part, and so on:

$$\begin{pmatrix}\text{average quant. noise power}\\ \text{in real and in imaginary}\\ \text{parts of the output}\end{pmatrix} \approx \frac{q^2}{12}\cdot\frac{N}{2}\cdot(1-1/4-1/16) \approx \frac{q^2}{12}\cdot\frac{N}{3}\,. \quad \textit{(15.26b)}$$

One may recall that when the DFT is computed directly in fixed-point, with negligible error at multiplications, the quantization noise power in the real and in the imaginary parts of the DFT outputs at all frequencies with $0 < m < N/2$ is

$$\begin{pmatrix}\text{quantization noise power}\\ \text{in real and in imaginary}\\ \text{parts of all DFT outputs}\end{pmatrix} \approx N\frac{q^2}{12}\,. \qquad (15.27)$$

A great advantage of the FFT algorithm over direct computation is that the FFT algorithm requires much less computation when N is large. Another advantage is that the computed results have less quantization noise with the FFT algorithm. For example, with $N = 1024$, direct DFT computation has real and imaginary outputs having quantization noise power of $1024q^2/12$, in accord with Eq. (15.27). Comparing this with the average quantization noise power for the DIT FFT, which is about $341q^2/12$ in accord with (15.26b) above, the advantage of the FFT algorithm is noticable, although not very much.

However, this is not a typical case. For a significantly downscaled input signal, the (15.26b) term may cause unacceptably small SNR at the output. Therefore, in

practice either block floating-point FFT (see Subsection 15.8.1, page 394) is implemented on the fixed-point arithmetic (*sometimes this is also called fixed-point FFT*), or, if the control of block floating-point is not allowable in the implementation, the FFT is modified by regular downscalings (e.g. by prescaling the maximal sample of the signal before each stage to avoid overflow, or by simply downscaling by a factor of 2 in each stage). This latter is only by a factor of 2 more than necessary for a dominating sine wave in the input signal (the output peak is theoretically $AN/2$), but way too much for noise with large N values (see Subsection 15.6.3, page 387).

For the "downscaled" FFT's the analysis developed above for fixed-point is not valid. The SNR of the prescaled FFT is similar to block-float (see Subsection 15.8.1, page 394), although worse than it by something like 6 dB, depending on the implementation. The regularly downscaled FFT is good for dominantly harmonic signals. An analysis is preformed when solving the new Exercises 15.17 and 15.18.

In this section, we have not considered the errors caused by finite bit length representation of the coefficients W_N. These cause bias rather than PQN noise. Such errors are discussed in Chapter 21, and in some exercises (15.16, 15.19, 15.20, 15.21).

We have not discussed signal quantization at the input of the FFT, either. This may depend on many factors, like the nature of the stage preceding the FFT in the signal processing flowgraph. If the input is already in digital form, usually no extra quantization is necessary. If the input is of continuous amplitude, the quantization error of the complex outputs is about $\mathrm{var}\{X(k)\} \approx N \cdot q^2/12$, comparable to Eq. (15.27).

In general, the roundoff error depends on many circumstances: the type and properties of the signal, the number representation, the structure and length of the FFT, the scaling strategy, the rounding algorithm, the rounding of fractions of 0.5 LSB, overflow handling (modular or saturating), bit length of the multiplier, and existence of an accumulator with bit length larger than that of the memory. Without knowing all these, one cannot make general statements about the output quantization noise. Therefore, in the literature, formulas for specific signals and specific cases are given. We briefly deal with the two most important number representations in the next section.

Section 15.8.1, page 394:

15.8.1 FFT with Block Floating-Point Number Representation

Last but one paragraph:
In order to have an overall impression of the resulting noise, consider that for number representation with $B = 16$ bits (15 bits + sign), for $N = 512$, the dynamic range (the ratio of the sine waves that can be analyzed jointly) is limited to approximately 75 dB, see Exercise 15.5.

Exercise 15.6, page 399

15.6 Verify the theoretical results (15.25-corr), and (15.26b), see Corrigenda, page C5, for the fixed-point DIT FFT (with no block-floating scaling here), by Matlab-based computer simulation, using the tools available from the web page of this book.[18] Let $N = 256$, the number representation have $B = 32$ bits to represent numbers in $(-1, 1)$, and the input be a zero-mean white sequence uniformly distributed in $(-1/32, 1/32)$. In order to have a reference transform result, assume that the error of Matlab's `fft` command may be neglected. When determining the roundoff error,

(**a**) do not include either the effect of input quantization, or that of quantization of the trigonometric coefficients,

(**b**) include also the effect of input quantization, and of quantization of the trigonometric coefficients. Is the roundoff error much larger than without these roundoffs?

[18] `http://www.mit.bme.hu/books/quantization/`, e.g. `roundfft.m`

Exercise 15.7, page 399

15.7 Verify the theoretical result (15.27) for fixed-point DFT with no block-floating scaling, by Matlab-based computer simulation, using the tools available from the web page of this book. Let $N = 256$, the number representation use $B = 32$ bits to represent numbers in $(-1, 1)$ and the input be a zero-mean white sequence uniformly distributed in $(-1/32, 1/32)$. In order to have a reference transform result, assume that the error of Matlab's `fft` command may be neglected. What is the difference when the accumulator has a larger wordlength (e.g. by 8 bits) than the memory? When determining the roundoff error,

(**a**) do not include either the effect of input quantization, or that of quantization of the trigonometric coefficients,

(**b**) include also the effect of input quantization, and of quantization of the trigonometric coefficients. Is the roundoff error much larger than without these roundoffs?

Exercise 15.9, page 400

15.9 Verify the limit (15.28) given for block-float FFT, by Matlab-based computer simulation, using the tools available from the web page of this book. Let $N = 16, 32, \cdots, 2048$, let the number representation use $B = 18$ bits (this corresponds to Welch's 17 bits) to represent numbers in $(-1, 1)$, and the input be a zero-mean white sequence uniformly distributed in $(-1, 1)$, or in $(0, 1)$, respectively. Divide both sides by RMS$\{X\}$, to reproduce Figs. 4 and 2 of Welch.

In order to have a reference transform result, assume that the error of Matlab's `fft` command may be neglected. When determining the roundoff error,

(**a**) do not include either the effect of input quantization, or that of quantization of the trigonometric coefficients,

(**b**) include also the effect of input quantization, and of quantization of the trigonometric coefficients. Is the roundoff error much larger than without these roundoffs?

(**c**) Compare the result of **(a)** with the error caused by the roundoff of the trigonometric coefficients only.

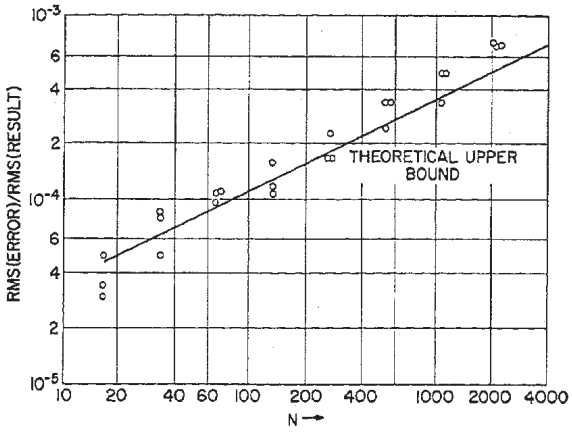

Fig. 2. Experimental error results: random numbers between 0 and 1; $B = 17$.

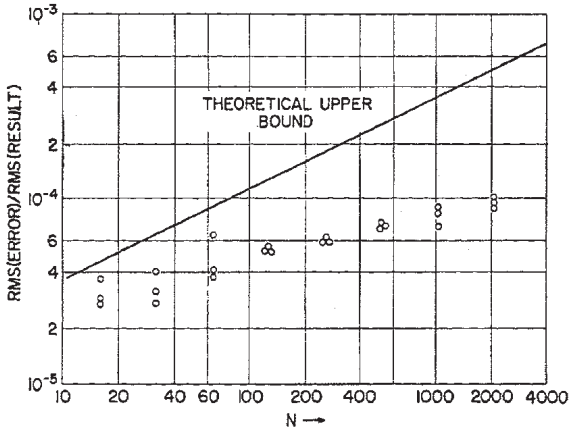

Fig. 4. Experimental error results: random numbers between −1 and 1; $B = 17$.

Figure 15.1 Welch's plots: roundoff errors and theoretical limits of the block-float FFT

Exercise 15.11, page 400

15.11 Check the error of Matlab's built-in `fft` command, by Matlab-based computer simulation, using the tools available from the web page of this book. Check the result against (15.11). Let $N = 256$, and the input be a zero-mean white sequence uniformly distributed in $(-1, 1)$. As reference, use $p > 56$. When determining the roundoff error,

(**a)** do not include the effect of input quantization,

(**b)** include also the effect of input quantization. Is the roundoff error much larger with input roundoff than without it?

The Quantization Noise Amplitude Distribution. Now, we take 101,000 samples from the error signal and construct a histogram with 101 bins that represent amplitude intervals ranging from -LSB/2 to +LSB/2. The result is shown in Figure 5 below. Quantizers and Quantization NoiseÂ¶. A quantizer is a signal processing block, that maps a continuous amplitude to a discrete amplitude. The output of the quantizer is discrete, meaning that it can only output Q different values.Â Note that since $2^b=q$ is an even number, the mid-tread quantizer cannot be symmetric in any case. In the following, we will focus on the mid-rise quantizer. Quantization NoiseÂ¶.