EoR Experiment - Memo
Quantization with Four Bits

D. Backer

March 20, 2007

Abstract

Digital signal processing (DSP) at radio telescopes has progressed from 1-bit to 2-bit to higher bit representation of signals. Advances have come from advances in electronics: many-bit, high-speed, commercial analog-to-digital converters (ADCs) and field-programmable gate arrays (FPGAs). A common minimum bit format today is 4-bit, while 8-bit is used whenever possible. These quantizations are particularly relevant to both the ADC operation and to interboard and interchip communication, while wider bit ranges are typically used within FPGA computational modules.

The 1-bit and 2-bit DSP required careful attention both to setting levels for optimum SNR and to subsequent calibration to a linear power scale. See Thompson, Moran & Swenson (2nd edition, 2001; Chapter 8) for details. With 4-bit quantization there are still issues to consider. The total power of a 4-bit quantizer has a non-linear response with respect to input level as shown in §1. The response of a correlator is highly linear with respect to changes of correlation at small correlation values, but the nonlinearity with respect to input power differs from that of the total power (§3).

1 Total Power Digital Gain

Noise voltages have Gaussian statistics. The probability density function (pdf) of the voltage ϕ is defined below. The probability of the positive voltages being below a threshold t is given by the error function (erf); i.e., erf(∞)=0.5. Explicit definitions and relations are:

$$\int_0^t \phi(x)dx = 0.5 \text{ erf}(\mathrm{t}/\sqrt{2}), \quad \text{where} \quad \phi(\mathrm{x}) = \frac{1}{\sqrt{2\pi}}\exp(-\mathrm{x}^2/2) \tag{1}$$

One can then write down the probability of the voltage being in a window of voltages of width δ around voltage t:

$$P_{t|\delta} = 0.5 \text{ erf}((\mathrm{t}+0.5\delta)/\sqrt{2}) - 0.5 \text{ erf}((\mathrm{t}-0.5\delta)/\sqrt{2}) \tag{2}$$

Here, δ is the step size in an analog to digital converter (ADC) or, equivalently the step size in any requantization of a digital signal processing path. All relations above are for an rms voltage of unity; i.e., t, x, δ are scaled by the rms (σ) of the Gaussian distribution. Power in a linear system is given by σ^2, which is unity for the scaling above.

We are interested in the case of digital signal processing (DSP) where the rms noise voltage is carried by a modest number of bits, 4-8. Both ADCs and any requantization algorithms are assumed:

- to have an odd number of levels, which provides a zero 'bin' and an equal number of quantization levels on the positive and negative sides of zero; and
- to saturate, which means that voltages beyond the range of the quantized output signal of ADC/algorithm are represented as the maximum value.

The ratio of the output power to the input power can then be calculated given δ and the number of bits. Figure 1 provides the result for 4-bit quantization. What is plotted with solid line is:

$$R_{\text{out}} = 2.0 * \Sigma_0^7 t^2 P_{t|\delta} + 7\delta * 7\delta * 2 * (0.5 - 0.5 * \text{erf}(7.5\delta)/\sqrt{2}), \tag{3}$$

where I have left in numerical factors in the second term to connect clearly to the previous equations. This reproduces Figure 7 of Backer et al. (1997 PASP 109 61), which also included a numerical simulation. The calculations are done in POWER.PRO (§A).

The interpretation of Figure 1 is that the digital gain is not linear; i.e., the curve is not flat at a value of unity. At low signal levels, rms volts small with respect quantized level volts, the output power response disappears into the '0' bin. At very large signal levels the input pdf spreads well beyond the available levels, and the output piles up in the maximum bin; the total power slowly declines with respect to the input power. The 'sweet spot' is around 2.0; i.e., rms voltage appears as 2.0 in the (re)quantized signal. More will be said about optimum level for best SNR, which is a separate issue.

If one is interested in the total power for calibration and/or astronomical measurements, then the levels need to be corrected for the nonlinearity curve in Figure 1. This correction is the equivalent of the Van Vleck correction in a 1-bit system. For example, in the PAPER[1] experiment we fit the total power (auto) data to a model of the system temperature plus the beam-convolved synchrotron background temperature. This gives us a 'digital counts to Kelvin' scale, along with an estimate of system temperature in Kelvins. The 24h track amplitude currently varies by a factor of two as the galactic plane transits our dipole beam. Thus we slide up and down the voltage scale by factor of square root of two, which is significant even with good setting of input power into ADC. This correction can be determined using the POWER.PRO IDL module. Alternatively, coefficients of a power-law representation of the useful part of the curve could be developed.

This curve was also reproduced by a numerical approximation scheme that estimated probabilities in each quantized bin by evaluating $\phi(x + \delta/4.0)$ and $\phi(x - \delta/4.0)$ and multiplying the sum of these by $\delta/2.0$. The IDL routine AUTO.PRO in appendix §B does this algorithm. The results are shown in Figure 1 with diamonds. The calculations are accurate down to $\sigma/\delta \simeq 0.5$, which is below where one would want to operate a 4b system.

Other work: generate polynomial fit over V=1-3; and/or 0.5-4.0; run similar code for 5b-8b cases; consider pdf with even number of levels that split signal at V=0 and don't have a 0 bin.

2 Total Power SNR

This was explored in Backer et al. 1997 where we found a peak SNR for voltage scale at 3.0. That is, quantization intervals of 0.33 in units of rms.

3 Cross Power Digital Gain

For quantization with 4b and more the output normalized correlation is very close to the input correlation over modest range of correlation coefficients; e.g., Jenet & Anderson (1998) Figure 1. "Correlation" is defined as $\rho \equiv < uv > \sqrt{< u^2 >< v^2 >}$ assuming u, v are real, Gaussian variates. The linearity coefficient will vary if the total power changes in a fixed gain system viewing varying system temperature. This might be a concern in pushing gain calibration beyond the 1% level.

A wise choice during any requantization is to equalize high-bit data before running through a 4b quantizer at least out to 3-dB points on any passband response and toss stuff beyond that, or leave as "don't care". This does demand Monitor & Control overhead to determine what to do and to carry along meta-data about what was done. A better alternative is to always have a 'sufficiently flat' (1-2 dB?) analog response, including any system temperature vs frequency as we have with the PAPER experiment.

One can calculate the cross power given the true correlation coefficient. For input voltages u, v (not to be confused with u, v baseline projections) from two antennas (or polarizations)

[1]Precision Array to Probe Epoch of Reionization; http://astro.berkeley.edu/ dbacker/EoR/

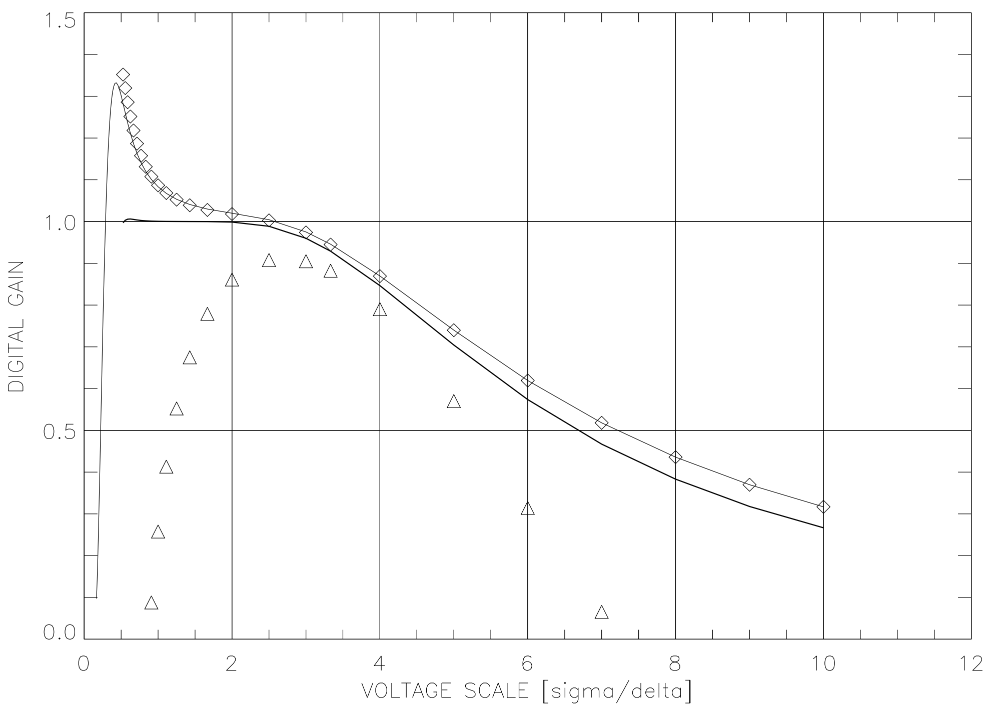

Figure 1: Thin solid line and diamonds: normalized total power vs voltage scale, where the scale is given as the ratio of rms of the gaussian noise volts (sigma) in units of the quantization (delta) volts. Thick solid line: normalized cross power vs voltage scale. Triangles: ratio of total power to cross power after subtraction of 0.97, multiplication by 10 and addition of 0.75 offset. See text for further details.

with correlation ρ the joint pdf is

$$p(u,v|\rho) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp[-(u^2+v^2-2\rho uv)/(1-\rho^2)]. \tag{4}$$

The probability of $u=i, v=j$ for quantized signals with quantization interval of δ is

$$P_{ij|\delta\rho} = \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{(i-0.5)\delta}^{(i+0.5)\delta} du \int_{(j-0.5)\delta}^{(j+0.5)\delta} dv \exp[-(u^2+v^2-2\rho uv)/(1-\rho^2)]. \tag{5}$$

As an aside, Jerry Hudson, in a 1991 RAL internal memo, worked on this expression for the case of a 2b, 4-level correlator. He rotated the coordinates 45° (u, v to x, y) to separate variables and then reduced the expression to a single integral form. The extension of the Hudson integral to a many-bit correlator is direct:

$$\begin{aligned} P_{ij|\delta\rho} &= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{x_1}^{x_2} \exp[-a^2x^2](\mathrm{erf}[\mathrm{b(x-y_1)}] - \mathrm{erf}[\mathrm{b(x-y_2)}]), \quad \text{where} \qquad (6) \\ a &\equiv 1/\sqrt{2(1+\rho)}, \quad b \equiv 1/\sqrt{2(1-\rho)}, \quad \text{and} \\ y_1 &\equiv \sqrt{2}(i+0.5)\delta, \quad y_2 \equiv \sqrt{2}(i-0.5)\delta, \quad \text{and} \\ x_1 &\equiv \sqrt{2}(j+0.5)\delta, \quad x_2 \equiv \sqrt{2}(j-0.5)\delta. \end{aligned}$$

The above, at the moment, is correct in form although I'm still checking on the exact statement of limits (x_1, x_2, y_1, y_2); but computer speed suggests just staying with 2D integral above. Here

I have simplified his investigation to the case where the levels of each ADC/quantizer are the same. Adding differences would result in much more laborious calculation and is probably better dealt with by a Monte Carlo approach.

The unnormalized cross power of a 4b digital processor can then be estimated by

$$< uv >= 2.0\Sigma_{i=1}^{7}\Sigma_{j=i+1}^{7} ij[P_{ij|\delta\rho} - P_{j\bar{i}|\delta\rho}] \tag{7}$$

The factor of 2 results from symmetry with sum just done over half of the pdf, and $\bar{i} \equiv -i$. With sufficient quantization and 'good' choice of quantization interval δ one might just approximate

$$P_{ij|\delta\rho} \simeq \frac{1}{2\pi\sqrt{1-\rho^2}} \exp[-(i^2 + j^2 - 2\rho ij)\delta^2/(1-\rho^2)]\delta^2 \tag{8}$$

Calculations were done with IDL routine CROSS.PRO given in appendix C and are shown with bold line in Figure 1 that sits just below the total power line. Again what is shown is the raw correlation response as a ratio to the expected response. The example chosen is $\rho = 0.01$. Similar results obtain over a wider range of correlations. Unlike the total power, the correlation approaches the true correlation below voltage scale of 2.0 (rms voltage σ in units of quantization interval δ).

The ratio of the cross power digital gain to the total power digital gain is not constant as the total power changes. This is illustrated by the triangles in Figure 1 that show the ratio relative to 0.97 and expanded by a factor of 10 and centered arbitrarily at 0.75; i.e, the ratio is approximately 0.97 when voltage scale is near 1.6 and again near 4.2 but rises by 1.5% between these two values. If accurate calibration is required, then corrections are necessary either in realtime via an automated leveling module/algorithm or offline via calculations such as these shown.

The 4b system is very linear for small correlations and fixed total power. The CROSS program was run for a variety of correlations: $\rho = [0.001, 0.01, 0.002]$ and $\rho = [0.01, 0.1, 0.02]$; the three values are start,end,step considered. In the first case the ratio of measured to ideal varied only in parts per million. In the second case many of the steps in δ also showed only parts per million changes in measured to ideal ratio. At $\sigma/\delta = 3.3$ the ratio increased from 0.929076 to 0.929121 as ρ was increased from 0.01 to 0.09.

4 Cross Power SNR

The optimum SNR for small signal detection is different than that for total power increment in low-bit quantization cases; e.g., memo by J. Hagen on the 3-level case for Arecibo correlators in the early 1990s. For larger number of levels we don't expect a difference. Jenet & Anderson (1998; table 3) find 0.32 for 4b case; use of approach in Thompson (1998), Thompson, Swenson & Moran (2001; 3rd edition, table 8.2) report 0.34 (with even 16 levels). Jenet & Anderson report 0.054 (1/18.5) and 0.030 (1/33.0) for the 7b and 8b cases, respectively. A relevant issue in setting of gains for an ADC would be whether lsb is to be ignored owing to dither, which would turn an 8b unit into an effective 7b unit. In general the SNR dependence on quantization level is slowly varying and other considerations maybe more important. One other consideration is operation, as in PAPER, of a fixed gain system but one with variable system temperature.

APPENDICES

A power.pro

```
PRO POWER,x,d
    ; x - array of quantization levels; e.g. [0,1,...7]
    ; d - quantization interval (delta) in units of rms (sigma)
    ; square root of 2
    s2 = sqrt(2.0)
    dh = d/2.0
    ; scale array to units of rms
    t = x*d
    ss = 0.0
    l = n_elements(t)
    for i=1, l-1 do begin
    f = t[i]*t[i]*(erf((t[i]+dh)/s2) - erf((t[i]-dh)/s2))
    ; n.b., the 2.0 times for 2-sided integral and 0.5 factors cancel
    ss = ss + f
    endfor
    ; beyond quantization limit
    ssa = ss + t[l-1]*t[l-1]*2.0*(0.5 - 0.5*erf((t[l-1]+dh)/s2))
    ; output
    print,'POWER RESULT:',d,ssa
    END
```

B auto.pro

```
PRO auto, xi,d
    ;
    ; INPUTS
    ; xi - integer set of levels
    ; d - level interval (delta) in units of rms (sigma) of Gaussian pdf
    ;
    ; find range of quantization levels
    n = n_elements(xi)
    ; scale x to quantized units of rms
    x = xi * d
    s2 = sqrt(2.0)
    dh = d/2.0
    ; initialize w +half of zero bin
    xo = d/4.0
    arg = -xo*xo/2.0
    a = sqrt(1/6.283185)*exp(arg)*d
    sum = a/2.0
    sumxx = a*xo*xo/2.0
    ;print,n,sum,sumxx
    ; loop over half axis; 2-step integrations of probability
    for i = 1, n-1 do begin
    xo = x[i]+d/4.0
    as = sqrt(1/6.283185)*exp(-xo*xo/2.0)*d/2.0
    xo = x[i]-d/4.0
    as = as + sqrt(1/6.283185)*exp(-xo*xo/2.0)*d/2.0
    ; f = 0.5 * (erf((x[i]+dh)/s2) - erf((x[i]-dh)/s2))
    sum = sum + as
    sumxx = sumxx + as*x[i]*x[i]
    ; print,i,x[i],f/as,2.0*sum,2.0*sumxx
    endfor
    ; do rest of axis for x>max
    for xx = x[n-1]+d,7.0,d do begin
    xo = xx+d/4.0
    as = sqrt(1/6.283185)*exp(-xo*xo/2.0)*d/2.0
    xo = xx-d/4.0
    as = as + sqrt(1/6.283185)*exp(-xo*xo/2.0)*d/2.0
    sum = sum + as
    sumxx = sumxx + as*x[n-1]*x[n-1]
    endfor
    print,'AUTO RESULT:',1.0/d,2.0*sumxx
    END
```

C cross.pro

```
PRO cross, xi,yj,r,d,pdf
    ;
    ; INPUTS
    ; x,y - integer set of levels; same dimensions; typically = [0:7]
    ; r - correlation between x & y variates
    ; d - level interval in units of rms (sigma) of pdf
    ;
    ; integrate half plane by including i,j and j,i to setup
    ; for <xy> calculation
    ; find range of quantization levels
    n = n_elements(xi)
    ; x[0] = y[0] = 0.0
    ; scale x,y to quantized units of rms
    x = xi * d
    y = yj * d
    ; initialize w half zero bin
    xo = d/4.0
    yo = d/4.0
    arg = (-xo*xo-yo*yo+2*r*xo*yo)/(2.0*(1-r*r))
    a = (1/6.283185/sqrt(1-r*r))*exp(arg)*d*d/4.0
    arg = (-xo*xo-yo*yo-2*r*xo*yo)/(2.0*(1-r*r))
    b = (1/6.283185/sqrt(1-r*r))*exp(arg)*d*d/4.0
    sum = (a + b)
    sumxy = (a - b)*xo*yo
    ;print,sum,sumxy
    ; loop over one quadrant, but do two quadrants simultaneously
    for i = 0, n-1 do begin
    for j = 1, n-1 do begin
    xo = x[j]+d/4.0
    yo = -y[i]-d/4.0
    arg = (-xo*xo-yo*yo+2*r*xo*yo)/(2.0*(1-r*r))
    b = (1/6.283185/sqrt(1-r*r))*exp(arg)*d*d/4.0
```

```
xo = x[i]+d/4.0
yo = y[j]+d/4.0
arg = (-xo*xo-yo*yo+2*r*xo*yo)/(2.0*(1-r*r))
a = (1/6.283185/sqrt(1-r*r))*exp(arg)*d*d/4.0
sum = sum + (a + b)
sumxy = sumxy + (a - b)*x[i]*y[j]
; print,i,j,a,b,2.0*sum,2.0*sumxy
xo = x[j]-d/4.0
yo = -y[i]-d/4.0
arg = (-xo*xo-yo*yo+2*r*xo*yo)/(2.0*(1-r*r))
b = (1/6.283185/sqrt(1-r*r))*exp(arg)*d*d/4.0
xo = x[i]+d/4.0
yo = y[j]-d/4.0
arg = (-xo*xo-yo*yo+2*r*xo*yo)/(2.0*(1-r*r))
a = (1/6.283185/sqrt(1-r*r))*exp(arg)*d*d/4.0
sum = sum + (a + b)
sumxy = sumxy + (a - b)*x[i]*y[j]
; print,i,j,a,b,2.0*sum,2.0*sumxy
xo = x[j]-d/4.0
yo = -y[i]+d/4.0
arg = (-xo*xo-yo*yo+2*r*xo*yo)/(2.0*(1-r*r))
b = (1/6.283185/sqrt(1-r*r))*exp(arg)*d*d/4.0
xo = x[i]-d/4.0
yo = y[j]-d/4.0
arg = (-xo*xo-yo*yo+2*r*xo*yo)/(2.0*(1-r*r))
a = (1/6.283185/sqrt(1-r*r))*exp(arg)*d*d/4.0
sum = sum + (a + b)
sumxy = sumxy + (a - b)*x[i]*y[j]
; print,i,j,a,b,2.0*sum,2.0*sumxy
xo = x[j]+d/4.0
yo = -y[i]+d/4.0
arg = (-xo*xo-yo*yo+2*r*xo*yo)/(2.0*(1-r*r))
b = (1/6.283185/sqrt(1-r*r))*exp(arg)*d*d/4.0
xo = x[i]-d/4.0
yo = y[j]+d/4.0
arg = (-xo*xo-yo*yo+2*r*xo*yo)/(2.0*(1-r*r))
a = (1/6.283185/sqrt(1-r*r))*exp(arg)*d*d/4.0
sum = sum + (a + b)
sumxy = sumxy + (a - b)*x[i]*y[j]
; print,i,j,a,b
; print,i,j,a,b,2.0*sum,2.0*sumxy
endfor
endfor
; do block A for x<=max; y>max
for i = 0, n-1 do begin
for yy = y[n-1]+d,7.0,d do begin
arg = (-x[i]*x[i]-yy*yy+2*r*x[i]*yy)/(2.0*(1-r*r))
a = (1/6.283185/sqrt(1-r*r))*exp(arg)*d*d
arg = (-yy*yy-y[i]*y[i]-2*r*yy*y[i])/(2.0*(1-r*r))
b = (1/6.283185/sqrt(1-r*r))*exp(arg)*d*d
sum = sum + (a + b)
sumxy = sumxy + (a - b)*x[i]*y[n-1]
endfor
```

```
endfor
;print,i,j,a,b,2.0*sum,2.0*sumxy
; do block B for y<=max; x>max
for xx = x[n-1]+d,7.0,d do begin
for j = 0, n-1 do begin
arg = (-xx*xx-y[j]*y[j]+2*r*xx*y[j])/(2.0*(1-r*r))
a = (1/6.283185/sqrt(1-r*r))*exp(arg)*d*d
arg = (-x[j]*x[j]-xx*xx-2*r*x[j]*xx)/(2.0*(1-r*r))
b = (1/6.283185/sqrt(1-r*r))*exp(arg)*d*d
sum = sum + (a + b)
sumxy = sumxy + (a - b)*x[n-1]*y[j]
endfor
endfor
;print,i,j,a,b,2.0*sum,2.0*sumxy
; do block C for x>max; y>max
for xx = x[n-1]+d,7.0, d do begin
for yy = y[n-1]+d, 7.0, d do begin
arg = (-xx*xx-yy*yy+2*r*xx*yy)/(2.0*(1-r*r))
a = (1/6.283185/sqrt(1-r*r))*exp(arg)*d*d
arg = (-yy*yy-xx*xx-2*r*yy*xx)/(2.0*(1-r*r))
b = (1/6.283185/sqrt(1-r*r))*exp(arg)*d*d
sum = sum + (a + b)
sumxy = sumxy + (a - b)*x[n-1]*y[n-1]
endfor
endfor
print,'CROSS RESULT',1/d,r,2.0*sumxy,2.0*sumxy/r
END
```