Lag Percision vs. Noise

D. Kaplan

The lags computed by the spectrometer are 16-bit values. However, most pulsar observations do not need this much precision and can use 4- or even 2-bit values without significant loss of signal-to-noise. Therefore, the Spigot cards can truncate the lags to a lower precision before sending them to the acquisition computer. This enables faster sampling, more lags, and/or more IFs to be recorded while maintaining the same data rate.

To preserve signal-to-noise as much as possible when truncating the data, each individual lag has a scale and offset which are applied to it prior to truncation:

$$\mathrm{lag}_{i,\mathrm{out}} = \mathrm{trunc}_b[(\mathrm{lag}_{i,16-\mathrm{bit}} + M_i)S_i], \tag{1}$$

where M_i is the offset and S_i is the scale, both for lag i, and the data are truncated to $b = 2$, 4, 8 bits as desired. What then remains is to determine the optimum values of M_i and and S_i.

We first determine the statistics of the 16-bit signal robustly (using an iterative σ-clipping algorithm), where we assume that the signal is a Gaussian with mean μ_i and standard-deviation σ_i, or $\mathrm{lag}_{i,16-\mathrm{bit}} \sim \mathcal{N}(\mu_i, \sigma_i^2)$.

To measure how good the lower-precision representation is, we define an error function

$$E(\mu_i, \sigma_i) \equiv \frac{1}{N} \sum_i (lag_{in} - lag_{out})^2. \tag{2}$$

We must scale this by $1/\sigma_i^2$ for the results to be comparable. Therefore, plotting $E(\mu_i, \sigma_i)/\sigma_i^2$ vs. (μ_i, σ_i) should allow us to determine the optimal range of (μ_i, σ_i) where the degredation is minimized. E_i/σ_i^2 is then related to the more standard signal-to-distorion ratio (SDR) by

$$SDR \equiv -10 \log_{10} \frac{E_i}{\sigma_i^2}, \tag{3}$$

where SDR is in dB.

We plot results of simulations of degredation in Figures 1 and 2. With 16 bit data, the finite precision (i.e. converting 2.5 to 2) has very little effect on E_i as long as $\sigma_i \gtrsim 100$. The largest effect is the limited range of the data, which causes miss-representations when $\sigma_i \gtrsim 7000$. The optimal range (for $\mu_i = 0$) is $100 \lesssim \sigma_i \lesssim 7000$. Changing μ_i has very little effect on $E(\mu_i, \sigma_i)$ as long as it is $|\mu_i| \lesssim 5000$. Above this, it begins to push data on one side of the Gaussian towards a limit. The optimal region has $E_i/\sigma_i^2 \lesssim 10^{-6}$. For 8-bits, this is similar to the 16-bit case, where the results are good for $20 \lesssim \sigma_i \lesssim 50$ and $|\mu_i| \lesssim 10$, and the optimal region has $E_i/\sigma_i^2 \lesssim 10^{-3}$. With the 4-bit case the effects of truncation are becoming more apparent, as only when $1.7 \lesssim \sigma_i \lesssim 3$ and $|\mu_i| \lesssim 2$ are the results good. The optimal region has $E_i/\sigma_i^2 \sim 10^{-2}$–$10^{-1}$. The 2-bit case has the smallest region with acceptable degredation. For $0.6 \lesssim \sigma_i \lesssim 1$, results are good, but $\mu_i = 0$ is required. The optimal region has $E_i/\sigma_i^2 \sim 0.8$–1.0.

The results, summarized in Table 1, show that there is a range of σ_i for which the lag_{out} have acceptable deviation from the lag_{in} ($\mu_i = 0$ is the best value in all cases). The values of σ_i that minimize E_i/σ_i^2 give SDRs comparable to the best values obtainable[1].

To then have the least degredation when truncating our input signal, we set $M_i = -\mu_i$ and $S_i = \sigma_{\rm opt}/\sigma_i$, where $\sigma_{\rm opt}$ is the optimal value of σ for a given precision, listed in Table 1. Prior to truncation the signal will then be $\sim \mathcal{N}(0, \sigma_{\rm opt}^2)$, as desired.

That there is a somewhat broad range for $\sigma_{\rm opt}$ is crucial, as we have assumed that the statistics of the incoming signal are stationary. Strong, variable sources or interference can alter the statistics. So we will normalize the signal for the middle of the optimal range (perhaps slightly to the low side of that), and then variations in signal strength will not significantly change the quality of the data unless $\Delta\sigma \sim \sigma$, which should only happen if the antenna temperature T fluctuates by $\Delta T \sim T$. This may happen due to significant RFI, but this will generally wash out the data too so that it will not limit our precision.

In the discussion above, we assumed that the data could be characterized properly. When operating in a 16-bit mode, this is easily done, but at the same time it is not crucial to set the scales and offsets precisely as a broad range of values will allow operation with negligible degredation. For modes with < 16 bits, and especially those with ≤ 4 bits, the scales and offsets are more important, but with some low-precision data it is hard to characterize it (determine its mean and standard-deviation). Therefore, the spigot computer can direct the spigot cards over the serial connection to enter a temporary 16-bit mode. This mode is not meant for normal operation and does not supply data for every time sample (if it did, it would violate the 25 Mb/s constraint on the hardware). For example, in mode 5 the data have 4-bits per lag, so the temporary 16-bit mode only supplies data every fourth time sample. In this way the data rate is held constant. While losing three out of four samples is not useful for observing, it does not degrade the data in a statistical sense, so these data can be used to determine the means and standard-deviations of every lag. The appropriate scales and offsets are then computed and sent over the serial connection, and finally the spigot cards are directed to return to the normal 4-bit mode.

---

[1]E.g. http://www.data-compression.com/vq.html

Table 1. Optimal σ_i for reduced-precision Gaussians.

Bits	Range for σ_i	$E(0,\sigma_i)/\sigma_i^2$
2	0.6 – 1	$\lesssim 1$
4	1.7 – 2.5	$\lesssim 10^{-1}$
8	20 – 40	$\lesssim 10^{-3}$
16	100 – 7000	$\lesssim 10^{-6}$

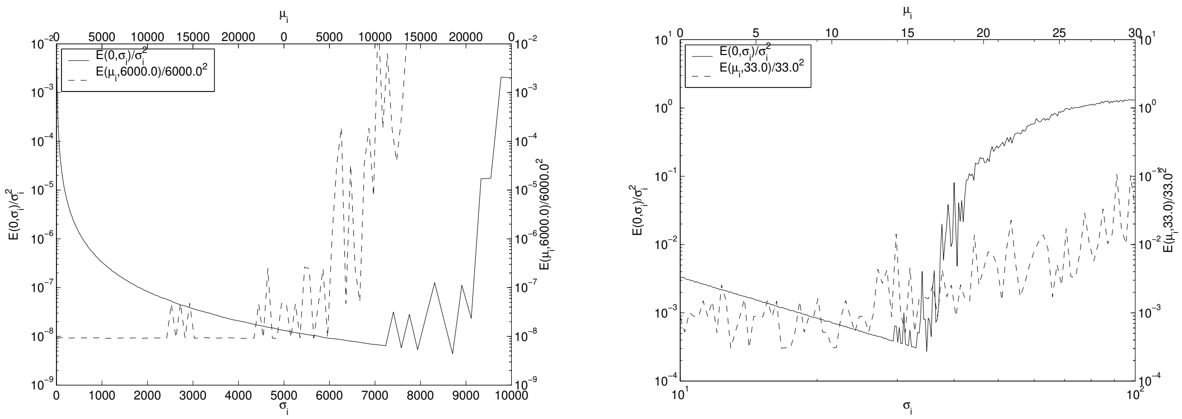

Fig. 1.— Error in estimating a Gaussian with 16-precision (right) and 8-bit precision (left). The solid curves and lower axes are for $\mu_i = 0$, while the dashed curves and upper axes are for $\sigma_i = 6000$ (16-bit) and $\sigma_i = 33$ (8-bit).

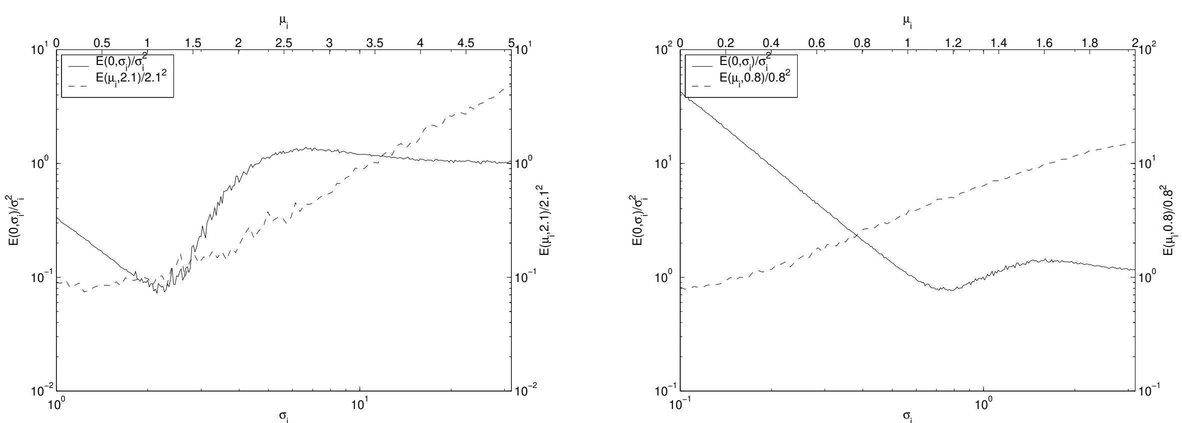

Fig. 2.— Error in estimating a Gaussian with 4-precision (right) and 2-bit precision (left). The solid curves and lower axes are for $\mu_i = 0$, while the dashed curves and upper axes are for $\sigma_i = 2.1$ (4-bit) and $\sigma_i = 0.8$ (2-bit).