Specifications and Clarifications of ALMA Correlator Details

Stephen Scott
21 June 2010
Revision 1.4

1. Integration time nomenclature
   The hierarchy described at the end of chapter two of the SSR Requirements describes what is needed from the perspective of the science. The readout restrictions for autocorrelations is a detail that does not have to be in the SSR Requirements. If the term "Dump" has other meaning in the hardware documentation, a possible resolution would be to change the SSR term to "Accumulation". [But it must be done quickly before the requirements become a project level document.]

2. Cross-correlation integration times
   The integration times available for science in cross-correlation mode are all multiples of 16 milliseconds, and result from the quantization of the read out time of the correlator hardware.

3. Auto-correlation only integration times
   When collecting only auto-correlation spectra, the minimum integration time within the hardware is 1 millisecond. However, the integration results can only be read out every 16 milliseconds. To eliminate a difficult alignment problem, the integration times available for science in auto-correlation mode are 1, 2, 4, 8, and 16 milliseconds, and any multiple of 16 milliseconds. [The correlator hardware actually sends a set of sixteen 1 msec data accumulations, but the CDP can integrate those up to the requested science integration time. By limiting the available integration times, a series of readouts can all be the same and integrations will always terminate on 48 msec Timing Event boundaries (or the evenly spaced 16 msec intervals between them) used for phase/delay setting and blanking.]

4. Integration chaining
   It shall be possible to specify the collection of a sequence of integrations, all with the same conditions. These integrations shall be collected sequentially in time, with no gaps between them.

5. Channel average integration times
   The channel average (channel 0) integration time shall be chosen in the range 0.5 to 1.024 seconds in length such that the smallest possible integral number of channel average integrations fill the spectral integration. It is acknowledged that quantization in the integration times can cause a small difference between the sum of the channel average integration times and the corresponding spectral integration time.

6. Spectral normalization
   The normalization for the correlation spectrum is specified in **SSR 2.3-R11**, but there is no similar requirement for the autocorrelation spectrum. We will specify normalization for both here.

   The correlator hardware produces raw auto and cross correlation functions of the quantized data as a function of lag. These have arbitrary scaling and are defined here as

   $$acfQuantRaw_{antI}(lag) \quad (1)$$

   $$ccfQuantRaw_{antI,antJ}(lag) \quad (2)$$

The correlation functions are normalized so that total correlation gives unity using

$$acfQuant_{antI}(lag) = \frac{acfQuantRaw_{antI}(lag)}{acfQuantRaw_{antI}(0)} \tag{3}$$

$$ccfQuant_{antI,antJ}(lag) = \frac{ccfQuantRaw_{antI,antJ}(lag)}{\sqrt{acfQuantRaw_{antI}(0)\,acfQuantRaw_{antJ}(0)}} \tag{4}$$

The correlation functions are corrected for quantization using a correction function appropriate for the digitization scheme, threshold levels, and multiplication implementation, using

$$acf_{antI}(lag) = quantCorrection(acfQuant_{antI}(lag)) \tag{5}$$

$$ccf_{antI,antJ}(lag) = quantCorrection(ccfQuant_{antI,antJ}(lag)) \tag{6}$$

Spectra are obtained from the correlation functions by Fourier transforming

$$acfSpec_{antI}(freq) = FT(acf_{antI}(lag)) - eff \tag{7}$$

$$ccfSpec_{antI,antJ}(freq) = FT(ccf_{antI,antJ}(lag)) \tag{8}$$

The *eff* factor in (7) represents the white noise contribution from the quantization noise and has a level equal to the correlator efficiency. This efficiency is a function of the rms signal level with respect to the quantization levels. Different Fourier transform implementations can have different normalizations, so we will specify a transform such that the average of all the spectral elements equals the input zero lag value. Note that for the autocorrelation functions this average is forced to unity. While the crosscorrelation spectral values will never exceed unity, the autocorrelation spectral values may be as large as the number of channels for the aberrant case where all the power is in one channel.

The crosscorrelation functions are next normalized so that the value in each spectral channel represents the fractional correlation of the power in that channel.

$$ccfSpecNorm_{antI,antJ}(f) = \frac{ccfSpec_{antI,antJ}(f)}{\sqrt{acfSpec_{antI}(f)\,acfSpec_{antJ}(f)}} \tag{9}$$

The crosscorrelation spectra are now in the fractional crosscorrelation units specified by the SSR requirements. Care is advised in the implementation of equation (9) to ensure that the result does not blow up from the division by small numbers near the band edges.

The normalization in equation (9) does an inherent bandpass correction, so the crosscorrelation spectrum on a continuum source is nominally flat. The weighting function across the channels is hence not flat, and reflects the bandpass shape of the autocorrelation spectra. This will be readily apparent in the increased scatter of the spectral channels near the band edge. Further calibration should multiply each channel by a Tsys that has been derived on a per channel basis.

It should be noted that there is limited experience in the discipline with the full normalization procedure outlined in this section and other considerations, including practical ones, may be encountered in the implementation that will require flexibility that is consistent with the novelty of the approach.

7. Delay corrections
   Errors in delay resulting from the discrete nature of the digital delay shall be accumulated over the course of the integration. The spectrum is corrected for the average delay error over the integration by the application of a linear phase slope to the data, with the slope proportional to the delay error.

8. Channel average
   The channel average for the crosscorrelation spectra is the vector average of the complex visibilities across the normalized spectrum. The astronomer may specify the range of channels to use for the channel average, e.g. to isolate a maser line or to excise a spectral line from the continuum. For a continuum source the channel average will have the same value as the channels, but with reduced noise. A significant phase slope across the band will cause a decrease in the channel average, requiring a calibration to be applied before averaging. The channel average is formed using a weighting function for the contribution from each channel. The weight is the gain from the auto spectra which is the same term used as a divisor to normalize the raw crosscorrelation spectra, making a weighted normalized crosscorrelation spectra equivalent to the raw. The next equation shows an example where the channel average is formed from a single contiguous region of the channels in the band.

$$chanAve_{antI,antJ} = \frac{\sum_{n=rBegin}^{rEnd} ccfSpec_{antI,antJ}(n)}{\sum_{n=rBegin}^{rEnd} \sqrt{acfSpec_{antI}(n)\, acfSpec_{antJ}(n)}} \qquad (10)$$

9. Archiving of autocorrelation spectra
   The autocorrelation spectra should always be taken and archived. These passbands may later be used in the calibration process and are certainly important for diagnostics of the front ends, filters, digitizers and data transmission system. They are a free spectrum analyzer. Note that there is no channel average for the autocorrelation spectra.

10. Archive data scaling (**SSR 2.3-R5**)
    The final spectral data must be efficiently stored in the archive. We will store each component of the complex visibility as a scaled 16 or 32 bit signed integers, corresponding to 2 bytes or 4 bytes of storage. The determination of the size of the integer will be done on a band basis (or sideband, if double sideband observation), with a 2byte/4byte flag and scale factor for each band. The maximum value of the correlation functions cannot be determined a priori, and in the case of a source strong enough to swamp the system noise can potentially approach unity. The noise level can be predicted and the scale factor shall be chosen so that the quantization error of the thermal noise will not be too visible and still allow a large fraction of sources to fit into 16 bits. A scaling value such that the system noise, $\frac{1}{\sqrt{B\tau}}$, is 30 after scaling will allow 15 Janskies of flux to fit into 16 bits under an optimistically high system sensitivity. In the equation for the system noise, *B* is the channel bandwidth and τ is the integration time. The quantization in the noise will be visible, particularly in the phase, at a low level, but the contribution to the noise is negligible. A suggested recipe for each spectrum in each baseband is:

- Compute a scale factor or the spectrum: $scaleFactor = 30 \times \sqrt{B\tau}$
- Multiply all the data in the spectrum by the *scaleFactor*
- Store *scaleFactor* as a header number that travels with the spectrum
- Find the absolute maximum value of the real and imaginary components. If the maximum is less than 32,768 then the data will fit in two byte integers, otherwise it will have to go into four byte integers.
- Round the data and then truncate to store in the appropriate integer.
- Store a header value giving the number of bytes used to represent the scaled data along with the spectrum.

The channel average has a different bandwidth and sampling time so it will have to be treated differently with an independent *scaleFactor*. However, the same equation that is used to compute the *scaleFactor* applies to both the multichannel data and for the channel average. As the channel average is really a single channel band, the trick will be finding an aggregation of samples that can be treated together so that the header expenditure is amortized. If the bandwidth is less than 250 MHz then 2 byte integers can be used (assuming an integration that is less than one second) because even full correlation can be accommodated using the multichannel *scaleFactor*. If the channel averages are grouped for storage (say as a vector of temporal samples that matches the multichannel integration), then these could be scaled as a group. If there is no appropriate grouping in the data format that will be used, then 250 MHz bandwidth can be used as the decision point between two and four byte integer storage.

The autocorrelation spectra will in general demand a greater dynamic range than the crosscorrelation (the noise is the same, but the correlation value for a rectangular passband is unity). For that reason they should each be treated as a separate band and scaled appropriately.

11. Autocorrelation power scaling
    Earlier we specified that the autocorrelation spectra be scaled to unity as part of its role in the expression of the crosscorrelation spectra as correlation coefficients. Storage of the autocorrelation spectra in this form allows for convenient reversal of the crosscorrelation normalization step. However, atmospheric calibration and single dish work the autocorrelation spectra must be expressed in absolute power density. The scaling factor, which is the zero lag autocorrelation value, must be carried with the data. This value must be corrected for quantization before storage so that it represents an absolute power level.

12. Tsys calibration
    The conversion of the normalized crosscorrelation spectra to antenna temperature spectra is not part of the correlator processing, but some of the issues are worth mentioning. In general, Tsys is a function of frequency and should be measured spectrally using the autocorrelation spectra and total power measurements, although it is recognized that digital filtering may make this process complicated. The double sideband temperatures measured using this technique must be split into contributions from each sidebands using an atmospheric model and independent receiver gains for each sideband. The receiver sideband gain ratios can be determined in the lab or astronomically. The Tsys spectra should be measured on source in case the source flux itself contributes to the system temperature.

13. Lab calibration
    It is interesting to extend the results of (9) to physical units that one might obtain with RF test equipment in the lab. If baseband power meters are available and measure the power in Watts, then the power measurements apply to the band measured by the channel average across the whole band. The amount of correlated power is obtained by

multiplying the crosscorrelation band average by the square root of the product of the two input power meter measurements. This correlated power is in Watts for the whole band. Division by the total bandwidth gives a power density in Watts/Hz. This same technique can be applied to the normalized crosscorrelation spectrum after dividing the power meter readings by the number of channels to correct for the narrower channel bandwidth yielding Watts/spectral channel. Using the spectral channel frequency width, the spectral power density for each channel could then be given in Watts/Hertz.