

ALMA Memo 583

Algorithms and formulas for hybrid correlator data correction

Giovanni Comoretto
INAF - Osservatorio di Arcetri

Last revised: 2008-11-05

Abstract: This report describes the algorithms and formulas used to reconstruct the hybrid auto or cross spectra from the ALMA correlator output.

This process includes the quantization correction ("Van Vleck correction") for the 2 bit quantization, Fourier Transform, calibration for tunable filter passband response and for different amplitude among sub-channels, stitching of the sub-channel spectra in a single composite spectrum, and final correction for the sampler 3-bit quantization.

The final product consists in normalized ACF and CCF spectra, processed and presented in a consistent and uniform way for all correlation modes.

1 Introduction

All acronyms and mathematical symbols are defined in Appendix C

The ALMA correlator produces a series of correlation functions (auto correlations, ACF, or cross correlations, CCF), of a digital representation of the radio signal, quantized either at 2, 3 or 4 bit. Each CF may represent a segment of delay lags (e.g. in CCF positive and negative lags are computed separately), and/or a different segment of the data, either spectral (in frequency division mode) or temporal (in time division mode). For sample representations greater than 2 bits/sample, different parts of a single correlation function are also computed separately, but the relevant re-assembly is done in hardware by the LTA.

Having so many different configuration modes result in a quite complex data processing structure, with several alternate branches.

In this document we will deal with the data processing required to produce a normalized ACF or CCF spectrum. The data processing for ALMA is described in document "*Specifications and Clarifications in ALMA correlator*", by S. Scott [11]. The processing described in this report replaces the procedure described there up to point 6. Subsequent processing remains unchanged.

Normalization of the spectra is performed as described in Scott [11]: for autocorrelation spectra, the average spectral density across the spectrum is unitary, and the quantity given in the cross spectra is the spectral cross correlation coefficient. The digital total power is given in units of the sampler quantization step.

The processing is kept as consistent as possible for all the correlator modes, i.e. time and frequency division, 2, 3 and 4 bit quantizations.

1.1 Scope

This document is relevant to the implementation of the correlator quasi real time software for processing the observed spectra. It provides general suggestions on software implementation topics, but do not suggest specific software implementations. Whenever possible different alternative approaches are suggested. It does not address astronomical calibration issues.

The input of the process is a set of raw correlation coefficients for each sub-channel, correlator plane and antenna (for ACF spectrum) or baseline (for cross spectrum). The output is a set of normalized auto and cross spectra, together with digital total power data.

1.2 Description of the problem

The usual processing of correlation data, described in [11], consists in lag normalization, correction for quantization, and subsequent calibration using a reference spectrum (usually the ACF signal) to correct for bandpass effects. This approach is not appropriate for the ALMA correlator in frequency division mode, for the following reasons:

- In frequency division mode, the signal is quantized twice, i.e. at the analog 3-bit sampler and after spectral processing in the Tunable Filterbank. Both these quantizations introduce a nonlinear distortion in the correlation function, that must be corrected.
- The bandpass response due to the digital filter is fully deterministic, both in amplitude and in phase, and therefore can be corrected using the filter design data.
- Having several subchannels, it is easier to realign and combine them in a single spectrum at an earlier processing stage than to deal with separate spectra that must be individually calibrated.

1.3 Processing steps

The general steps required for the processing of a set of raw correlation lags, to produce a set of auto-correlation and crosscorrelation coefficients is performed in the following steps:

1. Lag averaging: this includes assembling a complete CF from correlator products computed in different correlator planes
2. Normalization of raw lags, to obtain the quantized correlation product (chapter 2)
3. 2-bit (or 4-bit) quantization correction, using the appropriate correction function. The correction is dependent from the total power levels derived by zero-lag ACF (chapter 3)
4. FFT with appropriate taper, and with 1/2 channel frequency shift (chapter 4)
5. Bandshape correction, using a precomputed shape and with data-dependent correction (chapter 5)
6. Re-gridding of the corrected spectra inside the final composite spectrum (chapter 6)
7. Normalization of the output spectrum, and computation of the associate power level
8. Correction for the 3 bit quantization in the sampler (chapter 7)
9. Final rescaling of the output spectra, to comply with the conventions of [11] (chapter 8)

The processing corresponds in reverse to the signal processing in the station hardware, as shown in fig. 1.

For TDM only steps 1–4 and 9 are required. In frequency division mode, steps 1–5 must be separately performed on all sub-channels, that are combined together in step 6 and further processed in steps 7–9 as a single entity.

In FDM, the resulting spectrum does not usually span the full 2 GHz band, and may contain *holes* in the frequency coverage.

In the following chapters, each of the above operations is described in more detail.

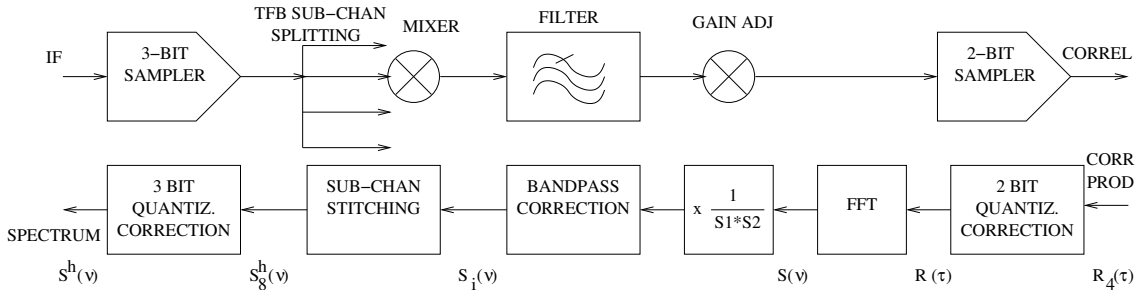


Figure 1: Signal processing in the station hardware (top row), and corresponding data processing in the software (bottom row)

2 Normalization of raw lags

The Short Term Accumulator (STA) in the correlator chip accumulates the correlation product $R_4(\tau)$ over the chosen dump time. For hardware considerations, the quantity accumulated is $R_4 + 9$, where the offset of 9 has been chosen to avoid negative numbers. The total number of samples in a millisecond is $(125000 - 270)$ where 125000 is the number of clock cycles, 270 is the number of clock cycles lost in the correlator chip dump. Dump time may be either 1 or 16 ms, after which the STA results are accumulated in the LTA.

The lower 10 bits of the register are not read (6 bits for 1 ms dump time), and thus the raw lag is divided by 64 or 1024¹. Therefore the offset in a dump period is $V_{s16ms} = 9(125000 - 270)/64 - 0.5 = 17539.65625$. Subtracting 0.5 from V_s compensates the error of truncation vs. rounding during STA readout. The total bias in a complete integration is then

$$V_s = V_{s16ms} N_p N_d \tag{1}$$

where N_p is the number of correlator planes co-added in the LTA, and $N_d = t_{int}/t_{dump}$ is the number of correlator chip dumps.

3 and 4 bit modes use 4 planes of the correlator, to synthesize a 4 bit multiplier from four 2 bit multipliers. For 3 bit mode only the lower 3 bits are used, i.e. the sample values have a restrict range, but the same hardware multiplier is used. The 4 planes are co-added with relative weights of 1, 4, 4 and 16, so the total offset implied in this scheme is increased by a factor of 25 (225 instead of 9). The simplest way to consider this is to include an extra factor of 25 in N_p .

The values of N_p for the available correlator modes are listed in tab. 1. This is the product of a factor of 32 for TDM, a factor of 25 for 3x3 and 4x4 quantization, and a factor of 2 for the oversampled modes at 62.5 MHz bandwidth (at 31.25 MHz oversampling is performed in the correlator hardware, without involving the LTA).

Mode	Quant.	Oversample	Band	N_p
TDM	2x2		2GHz	32
TDM	3x3		2GHz	800
FDM	2x2	no	any	1
FDM	2x2	yes	62.5	2
FDM	2x2	yes	31.25	1
FDM	4x4	no	any	25
FDM	4x4	yes	62.5	50
FDM	4x4	yes	31.25	25

Table 1: N_p for the available correlator modes. Bandwidth refers to the single sub-channel

¹Actually, one bit is discarded in the multiplication table, as the offset multiplication result is always an even number, and 5 or 9 bits are discarded in the STA readout

The normalization process is then expressed by the formula

$$R_4(\tau) = 9K \frac{L(\tau) - V_s}{V_s} \quad (2)$$

where $L(\tau)$ is the LTA raw lag output for delay τ .

The factor K is 1 for 2 bit modes, and 25 for 3 and 4 bit modes.

It is important to note that R_4 is a correlation product, and includes total power information for the 2 bit quantized signal. In particular R_4 is generally not equal to 1.0 for the zero-lag autocorrelation, but is typically around 3.5, 11 and 35 for respectively an optimally quantized 2, 3 and 4 bit signal. ²

3 Quantization correction

The correlation functions must be corrected for the effect of the quantization. This applies both at the quantization of the analog signal in the analog sampler, and at the requantization of the multi-bit digital signal after the Tunable Filterbank. The mathematical treatment is the same, irrespective of the analog or digital nature of the quantized signal.

Quantization correction depends on the exact ratio of the quantization step to the RMS amplitude for the two signals correlated (one signal in the case of ACF). In the ALMA quantization process, thresholds (or signal level) are set initially to the optimum value with respect to the signal level, within a relatively coarse accuracy (± 0.25 dB for the analog 2 or 3-bit quantization, $\approx 1\%$ for the 2-bit quantization inside the TFB), and then kept constant. During the observation, level may change, and thus accurate measurement of this ratio must be performed in order to reconstruct the signal.

This aspect is particularly important for a hybrid correlator, as the accurate relative calibration of the individual sub-channels is essential for accurate realignment of the reconstructed spectrum. A total power error of the order of the dynamic range (10^{-4} to 10^{-5}) is thus required to correctly align each sub-channel in the composite spectrum and avoid *platforming*.

The RMS signal amplitude before the quantization can be measured either directly, or inferred from zero-lag ACF. A direct measure of the digital total power in each sub-channel is in principle possible, but would saturate the the control bus bandwidth in most realistic scenarios. The latter method is usually much simpler, as the zero-lag ACF is directly available from the ACF data. Some bookkeeping is required to associate the zero-lag ACF of the two signals composing a CCF.

3.1 Zero delay autocorrelations

For Gaussian noise, the zero delay autocorrelation of a 2 bit sampled signal of amplitude σ expressed in units of the quantization step l is given by the formula:

$$R(0) = 9 - 8 \operatorname{erf} \left(\frac{1}{\sqrt{2}\sigma} \right) \quad (3)$$

The relation can be directly inverted, using the `inverse_erf` routine derived by Blair et al., and implemented as part of the GBT software by F. Schwab [12].

For multi-bit quantization, a similar relation can be derived. If N is the total number of levels, and assuming multiplicative weight $2k + 1$ for level k (the multiplication scheme adopted in the ALMA correlator), the zero-lag ACF is given by:

$$R(0) = (N - 1)^2 - \sum_{k=1}^{N/2-1} 8k \operatorname{erf} \left(\frac{k}{\sqrt{2}\sigma} \right) \quad (4)$$

It is possible to recover σ for all the antennas, BBCs and sub-channels by numerically inverting this relation (that is strictly monotonically crescent). A computer routine to invert this relation with standard numeric algorithm is also given by Schwab.

²The signal level must be adjusted at the sampler input for the quantization scheme adopted. The signal RMS amplitude at the analog sampler input should be set to 1.70 times the (3 bit) sampler quantization step in 3 bit TDM and all FDM modes, and to 2.01 times the quantization step in 2 bit TDM mode, ± 0.5 dB. This implies that in 2 bit TDM mode the RF level must be raised by 1.4 dB.

3.2 Quantization correction

The digital cross correlation R_4 of the 2-bit quantized signal is a function of the cross correlation ρ of the two *continuous*³ signals x_1, x_2 before the digitization stage, of the two normalized quantization steps l_1 and l_2 (or the normalized signal amplitudes σ_1, σ_2), and of the multiplication scheme adopted. The 2 bit 4 level multiplication scheme adopted in the ALMA correlator is implicitly assumed here. It is important to note that the function $\rho(R_4)$ depends strongly on l_1, l_2 , i.e. on the amplitude of the input signals. Correcting R_4 assuming a constant input level is absolutely not acceptable.

Several numeric approaches to compute $R_4(\rho)$ are available. The process is however quite computing intensive. It is possible to speed up this process by tabulating ρ as a function of R_4, σ_1, σ_2 , but the number of tabulated points needed for a meaningful interpolation with the required accuracy is quite high.

Several efficient algorithms to implement this quantization correction have been developed. For example, a set of routines developed for GBT are publicly available [12], and have been adapted to the ALMA multiplication table, for the 2, 3 and 4 bit cases. These routines compute a spline interpolation table for given values of σ_1, σ_2 , and apply it to the correlation function. Calculating the table is still quite computer intensive, and for short integration times it is beyond the available computing power. If, as it is usually the case, the RMS of the signal stays constant for a significant amount of time, one can compute the table for each sub-channel and baseline only once, and apply it to all the correlation functions for the whole time interval.

An efficient relation for $R_4(l_1, l_2, \rho)$ have been found by Schwab:

$$R_4(l_1, l_2, \rho) = -8 + 2 \left(\frac{\arcsin(\rho)}{\pi} + 3\operatorname{erf}\left(\frac{l_1}{\sqrt{2}}\right) + \operatorname{erf}\left(\frac{l_2}{\sqrt{2}}\right) \right) + 8(L(0, l_1, \rho) + L(0, l_2, \rho) + L(l_1, -l_2, \rho) + L(l_1, l_2, \rho))$$

where $L(l_1, l_2, \rho)$, is the cumulative bivariate probability that $x_1 > l_1$ and $x_2 > l_2$, and $l_{1,2} = 1/\sigma_{1,2}$.

3.2.1 Polynomial approximation

An alternative approach has been described in [3]. The relation $R_4(l_1, l_2, \rho)$ has been expanded in powers of ρ up to ρ^5 , and inverted up to the same accuracy. The resulting relation is quite accurate up to $\rho = 0.2$, i.e. for almost all the correlation points.

The approximate relation has the form $R_4 = a\rho + b\rho^3 + c\rho^5$, where a, b, c are given by:

$$a = \frac{2}{\pi} (1 + 2 \exp(-l_1^2/2)) (1 + 2 \exp(-l_2^2/2)) \quad (5)$$

$$b = \frac{1}{3\pi} (1 + 2(1 - l_1^2) \exp(-l_1^2/2)) (1 + 2(1 - l_2^2) \exp(-l_2^2/2)) \quad (6)$$

$$c = \frac{1}{60\pi} (3 + 2(3 - 6l_1^2 + l_1^4)) \exp(-l_1^2/2) (3 + 2(3 - 6l_2^2 + l_2^4)) \exp(-l_2^2/2) \quad (7)$$

and the inverse relation is

$$\rho = \frac{1}{a}R_4 - \frac{b}{a^4}R_4^2 + \left(\frac{3b^2 - ac}{a^7}\right)R_4^5 \quad (8)$$

These relations are symmetric, i.e. $\rho(-R_4) = -\rho(R_4)$, if no DC bias is present in the sampled data.

The correlation coefficient $\rho(\tau)$ must then be multiplied by $\sigma_1\sigma_2$, to obtain the cross product between the input continuous signals, $R(\tau)$.

This approach requires less computing time than the spline interpolation technique, as the computing intensive expression $R_4(l_1, l_2, \rho)$ needs not to be computed to build the spline interpolation table. If it is not known in advance whether the interpolation will be required (i.e. if there are correlation products with $|\rho| > 0.2$) the spline interpolation table must be computed anyway, and the polynomial relation can be used to speed up the construction of the spline table for the inner part of the correction curve. The speed-up is modest, around a factor of 2, but worth the extra effort.

³Here with *continuous* we may intend either an analog signal or a digital signal represented with many bit accuracy

3.2.2 3 and 4 bit quantization correction

The 3 and 4 bit cases can be treated in a similar way. From [3] for the generic N level case the N bit correlation coefficient R_N can be expressed as a polynomial in odd powers of ρ , truncated to ρ^5 . The particular case $N = 4$ gives the coefficients in equations 5-7. 3 bit quantization requires $N = 8$ even if the samples are actually quantized with 4 bits, as only 8 separate values are used.

$$R_N(\rho) = \frac{2}{\pi} \sum_{k=1}^5 \frac{\rho^k}{k!} C_k(l_1) C_k(l_2) \quad (9)$$

$$C_1 = 1 + 2 \sum_{i=1}^{N/2-1} \exp(-(il)^2/2) \quad (10)$$

$$C_3 = 1 + 2 \sum_{i=1}^{N/2-1} (1 - (il)^2) \exp(-(il)^2/2) \quad (11)$$

$$C_5 = 3 + 2 \sum_{i=1}^{N/2-1} (3 - 6(il)^2 + (il)^4) \exp(-(il)^2/2) \quad (12)$$

where $il = i/\sigma$ is the i^{th} threshold, and $l = 1/\sigma$ is the threshold spacing normalized to the signal RMS amplitude.

4 Fourier Transform

For detailed discussion on the Fourier Transform, we refer to chapter 2.1 in [2]. There it is shown that, if the sub-channels are overlapped by an even number of spectral points, it is convenient to compute the Fourier transform in a set of frequencies ν_i given by

$$\nu_j = (j + 1/2)\Delta\nu, \quad j = 0 \dots (N - 1) \quad (13)$$

where N is the number of (positive) correlation lags and $\Delta\nu = 1/(2N\Delta\tau)$ is the spectral point spacing. In this way, it is possible to discard an integer number of spectral points on each side of the band. Incidentally, this is convenient also for time division mode, or for modes with a single sub-channel, as the spectral points are spaced evenly across the observed spectrum, without a "special" first and last point.

Computationally, the simplest way to obtain this shift in the frequency channels is by multiplying the correlation function $R(\tau)$ before Fourier transform by an exponential:

$$S(\nu_j) = F \left[C(\tau_k) \exp \left(2\pi i \frac{k}{4N} \right) \right] \quad (14)$$

where $F[C]$ denotes the Fourier transform, and the index k assumes the values $[-N \dots (N - 1)]$. An efficient algorithm to perform modified FFT on real data is described in appendix A.

If the sub-channels are overlapped by an odd number of channels, the conventional FFT, computed on frequencies $\nu_j = j\Delta\nu$ with $j = 0 \dots N$, can be used. Since the first and last channels of conventional FFT of a real data set has half width, it is possible to discard a semi-integer number of channels from each side of the spectrum. To simplify the data reduction pipeline, and to guarantee a coherent interpretation of the output channel frequencies, the 1/2 spectral point offset is always applied. This constrains the number of overlapped points to be even (or zero). In the present correlator firmware, the subchannel overlap is fixed to 1/16 of the subchannel width, and the number of overlapped channels is always a power of 2.

In any case, discarding of unwanted spectral points is performed *after* the calibration described in the next section.

4.1 Tapering functions

Appropriate tapering $T(\tau)$ must be used before Fourier transform for apodization of the spectrum. ALMA specifications require that at least the following tapering functions $w(i)$ must be provided. The index i runs from 0 (zero lag ACF) to $N/2$, with N the two sided size of the correlation. The function is even, $w(-i) = -w(i)$.

- Uniform: constant weight (no tapering)
- Bartlett (triangular): $w(i) = 1 - |2i/N|$
- Welch (parabolic): $w(i) = 1 - (2i/N)^2$
- Hanning: $w(i) = 0.5 + 0.5 \cos(2\pi i/N)$
- Hamming: $w(i) = 0.54 + 0.46 \cos(2\pi i/N)$
- Blackmann: $w(i) = 0.42 + 0.5 \cos(2\pi i/N) + 0.08 \cos(4\pi i/N)$
- Blackmann-Harris: $w(i) = 0.35875 + 0.48829 \cos(2\pi i/N) + 0.14128 \cos(4\pi i/N) + 0.01168 \cos(6\pi i/N)$

According to [11], in the ALMA processing software the FFT routine is normalized so that the average spectral density over the whole spectrum is equal to unity for the autocorrelation spectra, and scaled accordingly for cross spectra. The quantity recorded for cross spectra is thus the correlation coefficient as a function of frequency.

In this way, absolute amplitude is lost, and is recovered in later processing by appropriate astronomical calibration. In a hybrid correlator however it is important to keep track of the actual signal amplitude, and therefore the correlation is normalized to the amplitude $\sigma_1\sigma_2$ of the two input data streams. In this way the average (real) spectral density for ACF is equal to the input total power, in units of the quantization step.

The final normalization to conform to the ALMA convention is done only at the end of the processing, as described in chapter 8, after the sub-channels are stitched together and the 3 bit quantization correction has been applied.

5 Bandshape correction

The resulting spectrum must be corrected for the filter shape and slope, as described in [2].

The combined response for the two digital filters and the tapering can be precomputed for all the observation modes. For each spectral channel j in the sub-channel, the integral response $a(j)$ and its first momentum $m(j)$ are tabulated in a *response file*. For symmetric filter taps and tapering function, these values are real. $a(j)$ is symmetric with respect to the sub-channel center, while $m(j)$ is antisymmetric. A separate response file is available for each spectral resolution, sub-channel width and tapering function. A procedure to compute these values from the filter taps and the tapering function $w(j)$ is reported in appendix B. With the 7 tapering functions above, 2 sub-channel widths (62.5 and 31.25 MHz) and 8 resolutions (64-8192 spectral points), a total of 102 calibration files are required.

The calibration algorithm is then:

1. Each (complex) spectral point is divided by $a(j)$, obtaining a first corrected spectrum, $S'(j)$.
2. S' is differentiated, obtaining $d(j) = (S'(j+1) - S'(j-1))/2$. The first and last points are linearly extrapolated from the two next ones, as in $d(0) = 2d(1) - d(2)$.
3. The spectrum is corrected for the small offset in channel barycenter induced by the filter slope: $S(j) = S'(j) + m(j)d(j)$.
4. A new estimate of $d(j)$ is obtained, and used to estimate a better correction for $S'(j)$

Only 2 iterations of steps 2-3 are sufficient, as more iterations just spread around numeric noise.

The correction function contains an appropriate scale factor that includes all the processing in the two stage filters.

The spectrum amplitude is affected by four more factors:

- A factor of 2^{22} or 2^{18} , to account for 11 or 9 bits discarded before the second quantization unit, respectively for 2 and 4 bit quantization
- A multiplicative, programmable, factor, used to rescale the filter output for optimal 2 bit quantization. This factor is determined in the SCC before the correlation, and is typically different for each sub-channel. The cross spectrum must be divided by these factors, one for each of the two baseline antennas
- A *conversion gain* of the digital mixer. This value is dependent on the particular sine table used in the mixer, but is constant and deterministic. For the mixer table adopted in the TFB, it is equal to 5.21 [1]
- A factor equal to the frequency division factor used (32 for normal TFB mode, or 64 for half bandwidth mode)⁴

At the end of this process, the rescaled spectra from each sub-channel are calibrated and corrected to the same scale, corresponding to the output of the 3-bit sampler. The quantity in the spectra is proportional to a spectral density, i.e. power per unit frequency and the scale is the same irrespective of spectral resolution and observing mode.

6 Sub-channel stitching

Each corrected spectrum must be re-gridded in the final hybrid spectrum, dropping the first and last channels. The digital local oscillator for each sub-channel has a tuning step of $(2/2^{16})$ GHz. i.e. $1/32$ of a channel at the minimum resolution. It is thus possible to tune each sub-channel in order to place all spectral points on a common frequency grid. In this case the re-gridding operation corresponds to computing the correct index in the composite spectrum.

The amount of spectrum to be dropped is a fixed quantity in frequency, related to the filter shoulder shape and the tapering function $T(\tau)$. This is independent from the spectral resolution, and from the sub-channel width (this latter is either 62.5 MHz or 31.25 MHz).

Thus if the sub-channel width is B , the number of spectral points N , the spectral resolution per point $\Delta\nu = B/N$, the number of spectral points to delete at each end is:

$$N_d = \frac{62.5\text{MHz}}{32\Delta\nu} = \frac{62.5\text{MHz}}{B} \frac{N}{32} \quad (15)$$

Let $S_k(j)$ be the spectral point j of sub-channel k . Let assume that sub-channels are numbered from 0 to $(M-1)$, spectral points from 0 to $(N-1)$ and the spectra produced with the modified FFT routine that computes spectral points centered at frequency $\nu_j = \Delta\nu(j + 1/2)$. Let ν_{0k} be the local oscillator frequency for sub-channel k . It is always possible to set ν_{0k} to an exact multiple of $\Delta\nu$, and this restriction can be enforced without losing significant flexibility. We assume thus that $\nu_{0k} = n_{0k}\Delta\nu$. For continuous coverage starting at frequency $\nu_0 = n_0\Delta\nu$, we have $n_{0k} = n_0 + (k + 1/2)(N - 2N_d)$. Then the frequency for spectral point j of sub-channel k is given by

⁴This factor is due to the convention used for the FFT normalization. The FFT routine normalizes the average spectral density to the total power in the time domain. In the frequency division mode, the average spectral density is calculated on a number of spectral points that is $1/32$ or $1/64$ of the total number of points in the whole channel. For example, for uniform spectral density, the total power in each sub-channel is $1/32$ the total power over the whole IF bandwidth, and thus the average spectral density calculated for each sub-channel is $1/32$ the total average. The factor is physically linked to the change in sampling rate, or in delay spacing in the correlator, and is not dependent on the actual bandwidth observed.

$$\nu_{jk} = \nu_{0k} - \frac{B}{2} + \frac{B(j+1/2)}{N} \quad (16)$$

$$= \Delta\nu(n_0 + k(N - 2N_d) + j - N_d) + \frac{\Delta\nu}{2} \quad (17)$$

If the resulting spectrum is represented by an array with the same indexing convention, i.e. spectral point l has frequency $\nu_l = \nu_0 + \Delta\nu(l + 1/2)$, the stitching process is a copy of arrays $S_j(k)$ to the final spectrum $S_8^h(l)$. The suffix 8 here indicates that S_8 is computed on a 3 bit, 8 level signal representation, i.e. has not been corrected for 3 bit quantization.

$$S_8^h(l) = S_j(k) \quad j = N_d \dots (N - N_d - 1), k = 0 \dots (M - 1) \quad (18)$$

$$l = k(N - 2N_d) + j - N_d \quad l = 0 \dots (M(N - N_d) - 1) \quad (19)$$

For example, in full bandwidth mode, 32 spectra of 64 points each are computed with an overlap of 4 points. 2 points are dropped from each sub-channel edge, and the spectral points j of spectrum k correspond to points $60k + j - 2$ in the final spectrum ($k = 0 \dots 31, j = 2 \dots 61$). A total of 1920 spectral points are computed.

The 4/64 overlap has been chosen considering the phase performance near the band edges of a 64-point correlator (see [2]). TFB filter shape has been designed accordingly, so this is the optimum value for the current hardware implementation. It is possible to use different values, with the constrain of a even number of spectral points in the overlap, but this is not currently supported by the firmware correlator.

At the end of this process, the composite spectrum corresponds to a portion (likely with holes, at least at the two edges) of the spectrum at the sampler output. It is thus equivalent to a portion of the spectrum observed with a conventional high speed 3-bit correlator on the same digitized signal.

7 Quantization correction for 3 bit sampler

The spectrum $S_8(\nu)$ is still affected by quantization losses and distortions due to the 3-bit quantization in the analog 4 GHz sampler.

In general for a frequency division scheme it is not possible to completely correct these effects, as the input spectrum is not completely observed. By dropping part of the spectrum, or in narrow-band modes by observing only a small portion of it, we lose information necessary to correct for quantization effects. To reconstruct the original spectrum it would be necessary to have at least a rough estimate of the spectrum, e.g. by a short and coarse observation in time division mode.

The relation between the 3-bit, 8 levels correlation function $R_8(\rho)$ and the original correlation ρ is much more linear than in the 4-level case, and in most cases the correction could be approximated with a linear function in the spectral domain. The sensitivity of the correction to the signal level is also less severe than for the 4 bit case, so a fixed relation could be used, assuming a nominal signal level at the sampler input.

Therefore we have the following possibilities to correct for 3-bit quantization, in order of decreasing complexity (and accuracy):

- Perform a short TDM observation, to estimate the spectrum in the unobserved portions of the IF band. The TFB itself can be used as a 32-point filterbank, to obtain a coarse spectrum of the whole IF band. From that, the algorithm described in chapter 7.1 can be used
- Estimate the spectrum in the unobserved portions of the IF band using a digital total power measurement to constrain the integrated spectrum
- Measure the digital total power at the TFB input and use a linear correction for the auto and cross spectra, as described in chapter 7.3
- Use the linear correction with the nominal value for the digital total power

7.1 Direct approach

The usual approach in FFT spectrometers (the hybrid design can be considered equivalent to a $N = 32$ Fourier instrument) is to derive the correlation function from the uncorrected spectrum, apply the quantization correction in the time (delay) domain, and re-transform the correlation function to the spectral domain. In our case this is not possible, as the spectrum is not known on the whole spectral domain (2 GHz bandwidth). Different methods to estimate the spectrum in the unknown regions are thus necessary.

The direct approach is to measure, with a coarse and quick time-division mode observation, the complete IF band. The derived spectrum is then used to fill the unobserved portions of the band. The method works both for ACF and CCF.

The coarse spectrum can be observed in time multiplexed mode, with a very short integration time. A 16 ms observation determines the spectrum with an accuracy of a fraction of a percent, much better than the absolute calibration of the individual IF channels. The observation must be performed in the same instrumental configuration, in particular the amplitude of the signal at the sampler input should be the same used for 3 bit quantization, even if 2 bit correlation is performed.

Let call this *reference* spectrum S^r , while the spectrum to be corrected is S^h . The corresponding correlation functions are R^r and R^h . The same symbols with a suffix, e.g. R_8^r indicate the corresponding quantities computed with a quantized representation with the indicated number of levels.

The coarse spectrum $S^r(\nu)$ must be observed in general with two bit quantization, as the 3 bit TDM mode is available only in some polarization modes, and has a different resolution than the spectrum to be corrected S^h . Therefore it must be modified to be compatible to the latter.

The simplest way to do this is (see fig. 2):

- Correct the observed correlation function (CF) $R_4^h(\tau)$ for 2 bit quantization, to $R^h(\tau)$
- Correct back $R^h(\tau)$ for 8 bit quantization, applying the correction routine backward. In this way a CF with a simulated 8 bit quantization error is obtained
- Expand the CF R_8^r with zeros to the appropriate length for the hybrid spectrum (e.g. to 8192 lags for a 32x256 lags hybrid configuration)
- Fourier transform the resulting $R_8^r(\tau)$ to obtain S_8^r
- Fill the unobserved regions of the spectrum S_8^h with S_8^r
- Transform back the resulting S_8^h to R_8^h
- Correct it for 8 bit quantization, obtaining $R^h(\tau)$
- Re-transform to $S^h(\nu)$

All direct and inverse FFTs are performed using the modified algorithm described in chapter 4. All quantization corrections must use the total power values derived from the zero-lag full-bandwidth ACFs $R^H(0)$.

A coarse estimate of the ACF spectrum can be derived using the TFB as a 32-point filterbank. The 32 subchannels must be positioned at equispaced points in the 2 GHz IF band, in order to completely cover it, and a digital total power measurement is performed with a short integration time. This has the advantage of using only the control computer, and the total power data can be easily stored together with other auxiliary informations.

The resulting spectrum is quantized to 3 bit, 8 levels (is S_8^r apart from a constant scale factor), so it can be directly used, after interpolation, to fill the unobserved portions of the ACF spectrum. For CCF, the integrated spectrum provides information on the digital total power, from which the linear approximation in chapter 7.3 can be used.

This method is usually quite time consuming, so alternative approaches are proposed.

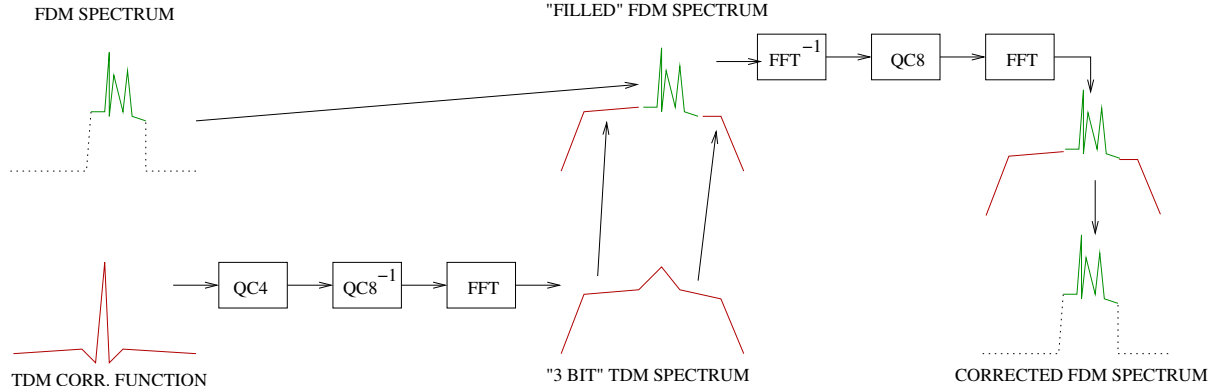


Figure 2: Quantization correction scheme using a TDM correlation function. The function is corrected for 2 bit quantization (QC4) and de-corrected for 3 bit quantization (QC8), to obtain an equivalent 3 bit TDM spectrum with the same resolution of the FDM spectrum. The spectra are merged together, and corrected for 3 bit quantization in the time domain

7.2 Interpolation of the unknown spectral regions

If a separate observation in TDM is not available, one may try to reconstruct the unobserved part of the spectrum with different methods.

Of course it is impossible with this approach to consider the effect of strong spectral features in the unobserved portions of the spectrum, but the consequent errors affect the observed portions only for unrealistically large spectral features.

If the digital total power at the TFB input is available, we may exploit the known relation between the total power and the integral of the spectrum over the whole 2 GHz bandwidth. It is thus possible to constrain the shape of the spectrum outside the observed region.

For example, one may assume that the unobserved edges at the two sides of the spectrum can be approximated with a linear slope. Constraining the total area of the spectrum to be equal to $NR_8(0)$, and assuming it to be continuous at the edges of the observed region, and to have equal amplitude at the two edges of the IF bandwidth, one can fill the unobserved regions with *reasonable* values. If more disjointed spectral regions are observed, the spectrum between them can be simply interpolated.

The resulting spectrum can then be transformed back to the time domain, corrected and re-transformed to the frequency domain.

The signal amplitude can be measured using the TFB total power meter. All the 32 TFB sub-channels must be used in parallel to measure the 32-fold time multiplexed input signal. This can be done in parallel with the normal use of the TFB filters, as the total power circuitry is independent from the filter. The requirements on the control system are however the same cited in chapter 3. As the gain error is much less sensitive to amplitude variations than in the 4 level case, the total power measurement can be much coarser, e.g. it can be performed only once at the beginning of an integration or of a group of integrations.

Simulation results with this approach give results only marginally better than with the linear algorithm described in the following chapter.

7.3 Linear approximation

For cross correlation, where $\rho \ll 1$, the quantization distortion $R_8(R)$ is with good approximation linear, and can be derived from the linear term of equation 9. As the FFT is a linear operation, the quantization correction can be expressed as a simple multiplication in the spectral domain:

$$S(\nu) = aS_8(\nu) \tag{20}$$

The main effect of quantization is an amplification of the input spectrum by a factor $(1/a)$, where a

can be derived from expression 10

$$a = \frac{\pi\sigma_1\sigma_2}{2} \left(1 + 2 \sum_{i=1}^3 \exp\left(-\frac{i^2}{2\sigma_1^2}\right) \right)^{-1} \left(1 + 2 \sum_{i=1}^3 \exp\left(-\frac{i^2}{2\sigma_2^2}\right) \right)^{-1} \quad (21)$$

where σ_1, σ_2 are the signal RMS amplitudes for the two antennas, with respect to the corresponding threshold spacing. These values can be computed from the digital total power measurement, $R_8(0) = \sigma_8^2$, and inverting relation 4 for the $N = 8$ case. This *quantization gain* depends on the signal level, and for a 1dB variation of the sampler input, it varies by $\simeq 0.3$ dB. This relation is quite accurate for $|\rho| \leq 0.3$, and introduce only modest nonlinearities even for $|\rho| \leq 0.8$.

For the autocorrelation, at least for a few points near $\tau = 0$ where $\rho \approx 1$, the linear relation 21 is incorrect. These points usually affect only the total power and in lesser way the overall bandshape.

The result of the 3 bit quantization under these assumptions is a compression of the spectral features with respect to the receiver noise of the order of 4-5%, or in an overestimate of the ACF continuum level by the same amount. As the ACF continuum is used in the calibration process, this correction is important in removing a systematic calibration bias.

If we don't care for the 2 GHz bandshape, assuming that it is reasonably smooth, we can assume that the only point where the linear approximation does not apply in the (unknown) 3-bit autocorrelation R_8 is $R_8(0)$.

The digital total power measured at the TFB input is by definition equal to the zero-lag ACF $R_8(0)$. From that, the zero-lag corrected ACF, $R(0) = \sigma^2$, can be computed. If we approximate the correction $R(\tau) = aR_8(\tau)$ for $\tau \neq 0$ and use the known relation (eq. 4) between $R_8(0)$ and $R(0)$ for $\tau = 0$, we can approximate the quantization correction in the frequency domain with a linear relation

$$S(\nu) = aS_8(\nu) - b \quad (22)$$

The minus sign has been chosen to have b positive.

Integrating the two sides of this relation, and considering that the spectra are normalized so that the average power is the zero-lag ACF, one obtains

$$b = aR_8(0) - R(0) \quad (23)$$

where $R_8(0)$ is the measured digital total power and $R(0)$ is obtained inverting eq. 4.

Numeric values for a (red, left scale) and b (green, right scale) are plotted in fig. 3 for input level at the 3-bit sampler around the nominal level of $\sigma = 1.706$ times the threshold spacing.

The above relation is independent of the actual shape of the spectrum outside the measured region, as long as no lags of the (unmeasured) ACF have a correlation greater than 0.2–0.3. This is not true in general, so nonlinear errors of less than 1% could be expected in the level and shape of the spectral baseline. This has been tested with simulations of realistic signals. These assumptions can be tested in post-processing if a TDM observation is available. In this case, it is always possible to “uncorrect” the spectrum, and apply the full procedure described in chapter 7.1.

7.4 Fixed correction

If no measure at all is available for the digital total power, the nominal values for a and b can be used in eq. 22 (for ACF) and 21 (for CCF). For the ALMA quantization scheme and for the optimal input level we have $a = 0.2698$ and $b = 0.1134$. If the fixed correction is used with a signal of different amplitude, using the wrong value for the factor a introduces a gain error, that is of the order of 3% for a input level variation of 1 dB at the 3-bit sampler input. Most of this error is removed when the spectrum is normalized, but a residual error, of the order of 0.5%, cannot be calibrated, due to the presence of the factor b in the formula 22.

8 Normalization and subsequent processing

All the subsequent processing in the ALMA software assume that the spectra are normalized to a total average spectral density of unity, for ACF at the sampler input:

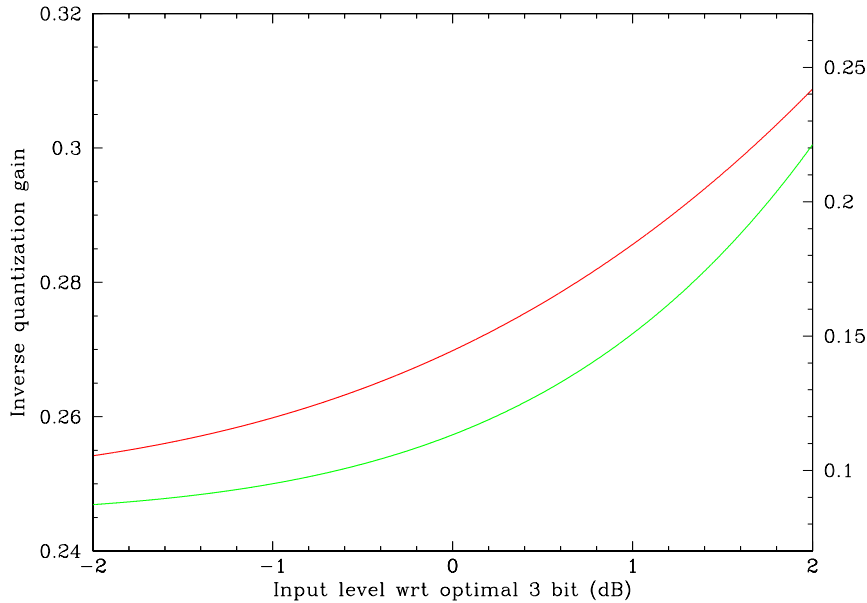


Figure 3: Inverse quantization gain, a (red, left scale) and autocorrelation offset b (green, right scale) as a function of the sampler input level. Input level is normalized to the optimum value for 3-bit quantization

$$\frac{1}{N} \sum_{i=0}^{N-1} S(\nu_i) = 1$$

The hybrid spectrum $S^h(\nu)$ must then be divided by the (digital) total power $R_i(0)$ for autocorrelation spectra of antenna i , and by $\sqrt{R_i(0)R_j(0)}$ for cross spectra between antennas i and j . $R_i(0)$ is derived either from:

- the zero delay ACF of the observed integration, corrected for quantization, in time division mode
- the zero delay ACF of the test TDM observation, when this is available as part of the procedure described in chapter 7.1
- a digital total power measurement, using the TFB digital total power counters
- the result of the analog total power meter, after appropriate calibration against the digital total power meter in the TFB
- the nominal value of $(1.706)^2 = 2.91$, if no measurements are available

$R(0)$ refers to the whole 2 GHz band, as the astronomical calibration is performed on this band, irrespective of the portion observed in the TFB. It should be noted that, while it is always $R(0) = \langle S(\nu) \rangle$ in time division mode, this is not true in general in frequency division mode, as $R(0)$ refers to the whole 2 GHz bandwidth that in general is not completely observed. Therefore it is no more true, in general, that the average of a normalized ACF spectrum is unity.

The resulting ACF and CCF spectra are at this point consistent with the spectra $acfSpec_{antI}(freq)$ and $ccfSpec_{antI,antJ}(freq)$ defined in equations (7) and (8) in Scott[11].

9 Summary

Summarizing the data processing, it is composed of the following steps:

1. Normalization of the raw lags $L(\tau)$ to obtain the uncorrected :

$$R_4(\tau) = N_m(L(\tau) - V_s)/V_s$$

where $V_s = N_m V_{s16ms} N_p N_d$ is the correlator bias, $N_m = 9$ for 2 bit quantization and 225 for 3 and 4 bit quantizations, N_p is the number of correlator planes co-added, $V_{s16ms} = 17539.65625$ is the bias for a single correlator dump, and N_d is the number of dumps in an integration cycle.

2. Total power measurement: for each autocorrelation, $R_4(0)$ is used to derive σ for each antenna and each sub-channel
3. 2 (or 4) bit quantization correction: a table of R_4 vs. ρ is calculated using the values of σ of the two antennas computed in step 2. The table may be computed once for repeated integrations with the same physical conditions. The table is computed using relation 8 for $\rho < 0.2$ and relation 5 for $\rho > 0.2$. For $\rho < 0$ symmetry is exploited.
4. $R(\tau)$ is computed from $R_4(\tau)$ using either eq. 8 or spline interpolation.
5. $R(\tau)$ is Fourier transformed to $S(\nu)$, using half channel shift and with the normalization convention that $\langle S(\nu) \rangle = R(0)$
6. Bandshape is corrected using the procedure described in chapter 5
7. Sub-channels are stitched together, copying them on a larger composite (hybrid) spectrum, $S_8^h(\nu)$
8. The spectrum is corrected using either the algorithm of chapter 7.1, or the linear relation $S^h(\nu) = aS_8^h(\nu) - b$. a is given by equation 21. b is zero for CCF, and $b = R(0) - aR_8(0)$ for ACF. $R_8(0)$ is measured using the digital total power detector in the TFB.
9. Each ACF is normalized by dividing it by $R(0)$. The CCF between antennas i and j are normalized by dividing them by $\sqrt{R_i(0)R_j(0)}$. The resulting spectra are consistent with quantities $acfSpec_{antI}$ and $ccfSpec_{antI,antJ}$ defined in equations (7) and (8) in [11].

Steps 6–9 are taken only for frequency division mode. For time division mode, step cw910 immediately follows step 5.

The quantities $R(0)$ (not the uncorrected $R_4(0)$, that depend on the quantization scheme adopted) are stored with the data.

All the subsequent processing follows the procedures in [11], from equations (9) on.

A FFT routine for modified frequency sampling

From [2], Appendix A

Many routines have been proposed in the literature to efficiently Fourier transform a real sequence. Given a sequence $g(\tau)$, of length $2N$, it is possible to obtain its spectrum in $N + 1$ frequency points $\nu_j = j\Delta\nu, j = 0 \dots N$, using a discrete Fourier transform of length N . A similar approach can be used to obtain the spectrum at the N points $\nu_j = (j + 1/2)\Delta\nu, j = 0 \dots (N - 1)$.

We will follow the algorithm described in *Numerical Recipes*[10], with the appropriate modifications.

Let the real function to be transform be:

$$g_k = g(k\Delta\tau) \quad k = -N \dots (N - 1) \quad (24)$$

In our specific case, g_k is the cross correlation product, computed for negative and positive lags. Due to the cyclic nature of the DFT, it is assumed that $g_N = g_{-N}$, that is not true. The effects of this misassumption are not considered here.

We follow the convention of denoting the Fourier transform of a function with the corresponding uppercase letter. In this appendix, however, we always use the modified Fourier transform computed on the set of points ν_j as specified above.

We want to compute the function G_j :

$$G_j = G((j + 1/2)\Delta\nu) \quad j = 0 \dots (N - 1)$$

We have $\Delta\nu = 1/(2N\Delta\tau)$. The function G_j is hermitian, $G_{-j-1} = G_j^*$. Using the function $\expi(x) = \exp(2\pi ix)$, we have

$$G_j = \sum_{k=-N}^{N-1} g_k \expi\left(\frac{k(j+1/2)}{2N}\right) = \sum_{k=-N}^{N-1} \left[g_k \expi\left(\frac{k}{4N}\right) \right] \expi\left(\frac{jk}{2N}\right) \quad (25)$$

Now let $h_{2k} = g_{2k} + ig_{2k+1}$, that is, we pack two successive samples as the real and imaginary part of a complex sample. In this way we obtain a complex sequence of length N (instead of $2N$). Transforming this sequence we obtain the function:

$$\begin{aligned} H_j &= \sum_{k=-N/2}^{N/2-1} g_{2k} \expi\left(\frac{k}{2N}\right) \expi\left(\frac{jk}{N}\right) \\ &\quad + i \sum_{k=-N/2}^{N/2-1} g_{2k+1} \expi\left(\frac{k}{2N}\right) \expi\left(\frac{jk}{N}\right) \\ &= H_j^1 + H_j^2 \end{aligned}$$

The first member is hermitian, since it is the Fourier transform of a real function, while the second is anti-hermitian, due to the coefficient i :

$$H_j^1 = H_{N-j-1}^{*1} \quad H_j^2 = -H_{N-j-1}^{*2}$$

Using these properties, one can separate the two parts, and combine them to recover G_j :

$$G_j = \frac{1}{2} \left[(H_j + H_{N-j-1}^{*1}) - i \expi\left(\frac{2j+1}{4N}\right) (H_j - H_{N-j-1}^{*2}) \right] \quad (26)$$

Using existing routines for DFT, the algorithm to be used to transform the real sequence g_k of length $2N$ is therefore:

1. Group consecutive real samples as the real and imaginary part of complex samples, obtaining a complex sequence h_k of length N
2. Multiply the sequence h_k by the complex factor $\expi(k/2N)$
3. Apply a standard DFT algorithm, obtaining the function H_j
4. Use relation 26 to obtain G_j

A computer routine to implement this algorithm is listed below. It is heavily based on the corresponding *Numerical Recipes* routine, and written in C++ (to use complex data type). The routine `four1()` performs a complex FFT in place.

```
// Real input (two sided, size 2n),
// complex output (size n)
// Modified version, with N output points equally spaced
// across bandwidth. s1[i] represents frequency (2i+1)/2N
// If sign==-1, s1 is input and s is output
```

```

void realft1(double s[], int n, complex s1[], int sign)
{
    double theta=M_PI/(double)n, c1=0.5,c2, wtemp;
    complex h1,h2,w,wp;
    const complex i(0.,1.);
    int i1,i2,j,n2=n/2;

    wtemp=sin(0.5*theta);
    wp=complex<double>(-2.0*wtemp*wtemp, sin(theta)); // exp(2*pi*i/2n) - 1
    w=complex(1.0, 0.0);
    if (sign == 1) {
        c2 = -0.5;
        for(j=i1=0; j<n2; ++j) {
            s1[j]=complex(s[i1], s[i1+1])*w*0.5; // phase slope to shift
            i1+=2; w+=w*wp; // 1/2 channel
        }
        w=-w;
        for(j=n2; j<n; ++j) { // the same for neg. times
            s1[j]=complex(s[i1], s[i1+1])*w*0.5;
            i1+=2; w+=w*wp;
        }
        four1(s1,n,1);
    } else {
        c2=0.5;
        theta = -theta;
        wp=conj(wp);
    }
    w=exp(complex(0., 0.5*theta));
    for (i1=0; i1<n/2; ++i1) {
        i2=n-1-i1;
        h1=c1*(s1[i1]+conj(s1[i2]));
        h2=c2*i*w*(s1[i1]-conj(s1[i2]));
        s1[i1]=h1+h2;
        s1[i2]=conj(h1-h2);
        w+=w*wp;
    }
    if (sign == -1) {
        four1(s1,n,-1);
        w=2.0;
        for (j=i1=0; j<n2; ++j) { // Correct phase offset
            s1[j] *= w;
            s[i1++]=real(s1[j]); s[i1++]=imag(s1[j]);
            w+=w*wp;
        }
        w=-w; // The same for neg. times
        for (j=n2; j<n; ++j) {
            s1[j] *= w;
            s[i1++]=real(s1[j]); s[i1++]=imag(s1[j]);
            w+=w*wp;
        }
    }
}

```


B Calculation of the correction table from the filter tap coefficients

The correction table used for bandshape correction (chapter 5) depends on the spectral response of each individual spectral point, i.e. on the output of each spectral point to a tone at a particular frequency at the TFB input. For simplicity, we neglected the effect due to the frequency conversion in the mixer, and considered just the response of the two stage filter, the decimation process, and the Fourier transform.

The overall scale factor of the signal processed by the tunable filter depends on details in the hardware implementation, as the number of bits retained after each processing stage. These are discussed in detail in [4].

The filter spectral response depends on:

- the filter tap coefficients used in the two FIR filters
- the number of lags in the correlator
- the tapering function used in the FFT routine

The sensitivity of the response to the adopted tapering function is modest, but not negligible. The error involved in choosing a different one is of the order of a few % in amplitude, increasing near the sub-channel edges.

For an ideal, non decimated, spectrometer, the response of the spectral point with nominal frequency ν_0 would be just the product of the individual responses, that can be obtained by Fourier transform the tap coefficients of the filters and the tapering function. The response of the filters must be squared, as they act on the electric field representation. The transform of the tapering function, $T(\nu)$ must not be squared, as it is applied on power data, and must be centered on ν_0 .

The decimation process, and the fact that the Fourier transform is computed on discrete lags spaced $\Delta\tau = 1/2B$, introduces two effects:

- Signals at frequency $\nu + 2kB$ and ν are aliased at frequency ν . Those referring to the reversed bands have conjugate phase.
- The discrete Fourier transform is cyclic, with period $2B$. This causes also the tapering function to be aliased. In particular, the a channel at frequency ν_0 can be described by a *direct* component, $T(\nu - \nu_0)$, and a *reversed* one, $T(\nu - 2B + \nu_0)$. The response to the real part of the Fourier transform is given by the sum of the two parts, while the response to the imaginary part is given by their difference.

The two effects combine together in producing a *direct* and *reverse* responses, given by:

$$A_+(\nu, \nu_0) = F(\nu - B/2) \sum_{k=-\infty}^{\infty} T(\nu - \nu_0 + 2kB) \quad (27)$$

$$A_-(\nu, \nu_0) = F(\nu - B/2) \sum_{k=-\infty}^{\infty} T(\nu + \nu_0 + 2kB) \quad (28)$$

where A_+ has direct phase, and A_- conjugated phase. The total response to the real and imaginary component of the signal are given by

$$A_r(\nu, \nu_0) = A_+(\nu, \nu_0) + A_-(\nu, \nu_0) \quad (29)$$

$$A_i(\nu, \nu_0) = A_+(\nu, \nu_0) - A_-(\nu, \nu_0) \quad (30)$$

The individual tapering response, $T(\nu)$, is computed on the discrete representation of the sampling function, evaluated for $\tau = n\Delta\tau$. T is identically zero for $|\nu| > B$, and its cyclic nature (with period $2B$) is represented by the infinite sum: only one term of the sum is nonzero for any given frequency ν . If it has to be computed analytically, instead that using a discrete Fourier transform, it is defined for any ν , and the sum must be carried on explicitly.

The filter response $F(\nu)$ can be computed by direct FFT of the filter tap coefficients of the two filters in the TFB. The two responses must be multiplied together, considering that, due to down-sampling, the frequency scale for the first filter must be 32 times that for the second filter, and the response for this latter must be repeated 32 times to fill the whole 2 GHz band. The filter response has to be shifted by $B/2$ due to the processing in the TFB: there the complex signal is filtered by a filter of band $B/2$, and converted to real by upconverting it by $B/2$. Both $F(\nu)$ and $T(\nu)$ are real, and even, so they can be computed only for $\nu \geq 0$.

For practical purposes, $F(\nu)$ is zero for $|\nu| > B$, even considering a very large transition band. In the TFB it is suppressed by 50 dB or more, so it gives a negligible contribution to the overall signal, even if in principle it is not exactly zero. This means that the two sums in 27 and 28 can be performed only for $k = [-1, 0, +1]$ and $k = [0, 1]$ respectively.

For increased resolution, the FFT can be computed with a much larger frequency grid. Interpolation of $F(\nu)$ and $T(\nu)$ can be performed by zero-padding the corresponding functions in the time domain. The ratio of the frequency grid to the spectral point spacing is denoted as (*oversampling factor*), and is in the range 4–16.

Integrating these quantities, and assuming that the tapering response has been normalized in order to have unitary integral, the response of the spectral point centered on ν_0 , and its first moment (frequency offset) are given by:

$$a(\nu_0) = \int_{-B/2}^{3B/2} A(\nu, \nu_0) d\nu \quad (31)$$

$$m(\nu_0) = \frac{1}{a_{\pm}} \int_{-B/2}^{3B/2} (\nu - \nu_0) A(\nu, \nu_0) d\nu \quad (32)$$

Both quantities must be computed separately for the real and imaginary parts of the signal, using A_r and A_i respectively.

To prevent rounding errors accumulation due to the large number of computations involved, it is essential to compute the above integrals using double precision quantities. The resulting response can be rounded and stored at single precision, if this is sufficient for the calibration accuracy.

B.1 Program implementation

A program has been written to numerically compute these quantities, given the filter tap coefficients, number of correlator lags, and tapering function.

The program runs from command line and accepts the following parameters:

- Total decimation factor after the second filter (32 or 64), corresponding to a final bandwidth of 62.5 and 31.25 MHz, respectively. The decimation between the first and the second filter is hardcoded as 32
- Number of lags in the correlation function (single sided, i.e. positive lags only)
- Oversampling factor, i.e. the number of frequency points in each correlator resolution element used to compute the integrals.
- Tapering function, identified by a numeric code. Codes are listed in table 2
- File names for file containing filter tap coefficients, resp. for the TFB first and second stage filters

The program produces a file name `filt_<nl>_<dd>_<taper>.cal`, where `np` is the number of lags, `dd` is the decimation, and the tapering is identified by a 5 letter acronym, listed in table 2. The file format is binary, with the first 3 words representing 32 bit integers, respectively for the decimation factor, the number of channels and the tapering code. Following words, in groups of `np` 32 bit floating point quantities each, specify the parameters a_r , a_i , m_r and m_i for each spectral point.

The oversampling factor determines the numerical accuracy of the integrals. It has been determined empirically, by increasing it until the results stop changing. For single precision results, a factor of 8 is adequate for all and 4 is sufficient for `np` greater than 256.

The tapering functions implemented in ALMA are summarized in table 2. They correspond to the tapering functions defined in chapter 4.

Code	Acronym	Tapering function
0	welch	Welch
1	bartl	Bartlett
2	black	Blackmann
3	bl_ha	Blackmann-Harris
4	hanni	Hanning
5	hammi	Hamming
6	unifo	Uniform

Table 2: Tapering functions, and associated numeric codes and acronyms

C Related documents and Drawings

C.1 References

References

- [1] Comoretto, G.: *A digital BBC for ALMA*, ALMA memo 305 (2000).
- [2] Comoretto, G.: *Phase effects in a hybrid correlator*, Arcetri Observatory Technical Report 4/2003 (2003) <http://www.arcetri.astro.it/pubblicazioni/Reports/03/3.03.html>
- [3] G. Comoretto: *Sub-channel stitching in a hybrid correlator*, ALMA document CORL-60.01.07.00-013-C-DOC
- [4] G. Comoretto, B. Quertier: *Data processing in the hybrid tunable filter*, ALMA document CORL-60.01.07.00-012-C-DOC
- [5] Cooper, B.F.C.: *Correlation with Two-bit Quantization*, Aust. J. Phys, **23**, 521 (1970).
- [6] R. Escoffier, G. Comoretto, C. Broadwell, R. Lacasse, J. Webber, A. Baudry “Observational Modes Supported by the ALMA Correlator”, ALMA memo 556
- [7] Hagen, J.B., Farley, D.T., Radio Science 8, 775 (1973)
- [8] Kogan, EVLA Science Memo 5 (1993)
- [9] Kogan, Radio Science 33(5), 1289 (1998)
- [10] W.H Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling: ”Numerical Recipes in C”, Cambridge University Press. Chap. 12.3: ”FFT Of Real functions”
- [11] Scott, S., “Specifications and Clarifications of ALMA Correlator”, <http://almasw.hq.eso.org/almasw/pub/CORR/CorrelatorDocuments/AlmaCorrel.pdf>
- [12] Schwab, F.R, ”Van Vleck Correction for the GBT Correlator” (2002), <http://www.naic.edu/~jeffh/fschwab-vanvleck.ps>

C.2 Glossary

channel The frequency interval chosen by one IF processor

sub-channel The frequency interval chosen by each FIR filter in the tunable filterboard

Correlation function The time averaged product of two signals x_i and x_j : $R_{ij} = \langle x_i x_j^* \rangle$. The signals may be continuous, or quantized to a limited number of levels

Correlation coefficient ρ : The correlation product normalized by the RMS amplitude of the two signals x_i and x_j : $\rho = \langle x_i x_j^* \rangle / \sigma_i \sigma_j$ By definition, $\rho = 1$ for the zero-lag autocorrelation function

Correlation raw lags $L(\tau)$: The raw counts provided by the correlator

Quantization correction The function that translates the quantized correlation product to the correlation product (or correlation coefficient) of the two original unquantized signals.

C.3 Acronyms

- ACF** Autocorrelation function
CCF Cross-correlation function
CF Correlation function (either ACF or CCF)
FDM Frequency division mode
LTA Long term accumulator
STA Short term accumulator (inside correlator chip)
TDM Time division mode
TFB Tunable filterbank

C.4 Mathematical symbols and definitions

The following symbols are used in the text (in order of use):

- $L(\tau)$ Raw correlation lags
 V_s Bias in $L(\tau)$
 t_{int} Integration time in milliseconds
 t_{dump} Time between correlator chip dumps, in milliseconds
 N_p Number of co-added planes
 $N_c = t_{int}/t_{dump}$ Number of correlator dump cycles
 $R_4(\tau)$ Correlation function (either ACF or CCF) quantized to 4 levels
 $R(\tau)$ Correlation function corrected for quantization
 $\rho(\tau) = R(\tau)/\sigma_1\sigma_2$ Correlation coefficient
 σ RMS of unquantized signal expressed in units of the quantization step.
 σ_4 RMS of 2-bit, 4-level quantized signal
 $l = 1/\sigma$ Quantization step in units of the RMS signal
 $P(x_1, x_2, \rho)$ Normal bivariate distribution
 $L(l_1, l_2, \rho) = \int_{l_1}^{\infty} \int_{l_2}^{\infty} P(x_1, x_2, \rho) dx_1 dx_2$ Cumulative bivariate distribution
 $S(\nu)$ Power spectrum. Is the FT of $R(\tau)$, with $\langle S(\nu) \rangle = R(0)$
 For the Tunable Filterbank:
 $S_i(\nu)$ Power spectrum of sub-channel i , corrected for bandshape and gain
 $S_8^h(\nu)$ Hybrid spectrum, quantized to 8 bit
 $S^h(\nu)$ Hybrid spectrum, after quantization correction

Contents

1	Introduction	1
1.1	Scope	2
1.2	Description of the problem	2
1.3	Processing steps	2
2	Normalization of raw lags	3
3	Quantization correction	4
3.1	Zero delay autocorrelations	4
3.2	Quantization correction	5
3.2.1	Polynomial approximation	5
3.2.2	3 and 4 bit quantization correction	6
4	Fourier Transform	6
4.1	Tapering functions	7
5	Bandshape correction	7
6	Sub-channel stitching	8
7	Quantization correction for 3 bit sampler	9
7.1	Direct approach	10
7.2	Interpolation of the unknown spectral regions	11
7.3	Linear approximation	11
7.4	Fixed correction	12
8	Normalization and subsequent processing	12
9	Summary	14
A	FFT routine for modified frequency sampling	14
B	Calculation of the correction table from the filter tap coefficients	17
B.1	Program implementation	18
C	Related documents and Drawings	20
C.1	References	20
C.2	Glossary	20
C.3	Acronyms	21
C.4	Mathematical symbols and definitions	21