Digital lag (XF) correlator theory.

David W. Hawkins, Ph.D.
California Institute of Technology,
Owens Valley Radio Observatory,
100 Leighton Lane, Big Pine, CA 93513-0968.
dwh@ovro.caltech.edu
January 4, 2002

Contents

1 Introduction .................................................. 3
2 The theory of cross- and auto-correlators .................. 3
  2.1 Cross correlators ............................................. 3
  2.2 Autocorrelators ................................................. 5
  2.3 Construction of a cross correlator using the prompt and delay elements of an auto-
      correlator .................................................. 6
  2.4 Construction of a correlator using counter-shifting input signals ..................... 8
3 Architectural options ........................................... 10
  3.1 Time demultiplexed correlator systems ....................... 10
  3.2 Deep memory correlator systems ................................ 14
  3.3 Cross correlator design equations ............................ 15
  3.4 The deep memory model versus time demultiplexing—examples ....................... 16
4 Digital correlator statistics .................................... 18
  4.1 Full precision sampling ........................................ 18
  4.2 1-bit sampling with 2-level quantization ........................ 23
  4.3 2-bit sampling ................................................ 25
      4.3.1 2-bit sampling with 3- and 4-level quantization .............. 26
  4.4 Allen deviation ................................................. 32
  4.5 Calculating the size of the accumulator in a correlator chip ...................... 33
5 FPGA implementation of a wide bandwidth correlator ........ 35
  5.1 *** to be completed *** ...................................... 35
List of Tables

1  Correlator design parameters ................................ 15
2  Correlator design examples–2MHz spectral sample spacing .... 17
3  1-bit quantization scheme ...................................... 24
4  1-bit multiplication scheme .................................... 24
5  2-bit multiplication scheme .................................... 28
6  Characteristics of the 2-bit digitizer multiplication schemes. .. 30

List of Figures

1  Positive and negative lags in cross-correlations .................. 4
2  Internal lag structure of an autocorrelator ....................... 6
3  Construction of a cross correlator ............................... 7
4  Internal architecture of an $N = 8$ lag cross correlator based on counter-shifting input signals ................................................. 9
5  Calculation of the correlation function via parallel processing methods ................................................. 11
6  The internal architecture for the example cross correlator chip. .... 12
7  A time demultiplex by 8 correlator ................................ 13
8  An ensemble of sample autocorrelation functions .................. 19
9  Block diagram of a 2-bit digitizer .................................. 25
10 Quantization and multiplication states and weights for 2-bit digitization schemes . . 26
11 Generic multiplication weights for a 2-bit digitizer ................. 27
12 Relative signal-to-noise ratio (SNR) of the 2-bit correlation schemes to the full precision correlation SNR ........................................ 30
13 The 2-bit biased multiplication scheme .............................. 31
1 Introduction

This document details the theory and operation of digital auto- and cross-correlators. Section 2 covers digital correlator theory, the notation used in this document, and digital correlator architectures. Section 3 contains the commonly implemented correlator architectures. The architectural options covered are; Nyquist-rate correlators (the correlator and digitizer clock frequencies are twice the input signal bandwidth), time demultiplexed correlators (the correlator clock frequency is the Nyquist rate divided by the time demultiplexing factor), and the deep memory model (the depth of the memory influences the relationship between the digitizer and correlator clock frequencies). Section 4 evaluates the signal-to-noise ratio (SNR) of sampled digital correlators relative to the SNR expected from a full precision (analog) correlator. This section covers the commonly used cases of; 2-level, 1-bit sampling, and 3- and 4-level 2-bit sampling. Section 5 details the application of the previous three sections to realize a flexible digital correlator in an FPGA using the hardware description language VHDL. This correlator is able to correlate in multiple sampling modes (2-level, 3-level, and 4-level) with multiple input bandwidths (Nyquist rate sampling from 31MHz to 500MHz).

Applications of digital correlators can be found in [1–5,7–12].

2 The theory of cross- and auto-correlators

2.1 Cross correlators

Cross correlators are used in interferometric arrays to determine the cross-power spectrum between telescope pairs. These cross-power samples correspond to samples in the 3-D (data cube) velocity-spatial frequency space of an astronomical object.

The digital cross correlation of a waveform \( x(t_k) = x(k\Delta_t) = x[k] \) with another waveform \( y(t_k) = y(k\Delta_t) = y[k] \) is

\[
    r_{xy}^+(\tau_n) = r_{xy}^+[n] = \sum_k x[k]y[k + n]
\]

where lag time \( \tau_n = n\Delta_t \) has the same sense as time \( t_k \), the lag indice \( n \in [-N/2,N/2-1] \), \( N \) is even and typically a power of 2, and the superscripted ‘+’ indicates the sense of the lags. Fig. 1(a) demonstrates the relationship between the \( N \) lags and the two waveforms at a particular instant of time \( t_k = k\Delta_t \). Note that negative lags occur at times earlier than time instant \( t_k \) and positive lags occur after \( t_k \).

The following form of cross correlation is also used:

\[
    r_{xy}^-[n] = \sum_k x[k]y[k - n]
\]

Fig. 1(b) shows how this redefinition affects the location at which the lags are calculated. The relationship between the two cross correlation functions in the time domain is

\[
    r_{xy}^+[n] = r_{xy}^-[\pm n]
\]

for \( n \in [-N/2-1,N/2-1] \). The two correlation functions do not overlap at the extremes, i.e, at \( n = -N/2 \) or \( n = N/2 \).

The relationship in (3) transforms to the frequency domain as

\[
    R_{xy}^+[m] = R_{xy}^-[\pm m] = (R_{xy}^-[m])^*,
\]

for \( m \in [-N/2-1,N/2-1] \). The two frequency functions do not overlap at the extremes, i.e, at \( m = -N/2 \) or \( m = N/2 \).
Figure 1: Positive and negative lags in cross-correlations. This figure shows how the definition of the cross correlation function affects the location of the positive and negative lags. (a) Shows $x(t_k) y(t_{k+n}) = x[k] y[k+n]$ for $n = -N/2, \ldots, N/2 - 1$ at a particular instant in time, $t_k$. (b) Shows $x(t_k) y(t_{k-n}) = x[k] y[k-n]$. Redefining the $n$ term in the cross correlation product reverses the location (and hence the asymmetry) of the positive and negative lags.

where $m$ is the frequency indice with the same range as $N$, and the superscripted asterix implies complex conjugation. (This relationship is exact only if the $N-1$ overlapping samples in $r_{xy}^+[n]$ and $r_{xy}^-[-n]$ are used. An $N$ point FFT can still be used, provided a single zero is used to pad the sample
sequences back to $N$ points.

The correlation of $x(t_k)$ with $y(t_k)$ is not the same as the correlation of $y(t_k)$ with $x(t_k)$. Switching the order in which the input functions are specified inverts the lag ordinate, i.e.,

$$r_{yx}[n] = \sum_k y[k]x[k+n] = r_{xy}[-n] = r_{xy}[n].$$  \hfill (5)

This derivation assumes that $k \in [-\infty, \infty]$. In practice $K$ samples are summed, where $k \in [0, K-1]$. If the non-existent samples at the extremes are considered to equal zero, a slight taper is introduced into the final correlation function, i.e., $K$ samples will contribute to lag zero, $K-1$ samples will contribute to lags 1 and -1, etc. When $K \gg N$, this effect is minor, since the extreme lag $N/2$ is tapered by $K-N/2$.

Typically the cross correlation function is calculated as a real valued function. A real valued function, $h(t)$, has the property that its spectrum, $H(f)$, is Hermitian, i.e., the real part of the spectrum is even and the imaginary part is odd. This property is mathematically described by

$$H(f) = H^*(-f).$$  \hfill (6)

Because of this property, it is necessary only to calculate positive frequencies of the cross correlation function.

Throughout this document, lags are indicated as positive or negative as per (2) and Fig. 1(b) and the superscripted ‘-’ is dropped for convenience. With this convention, an autocorrelator (discussed in the next section) produces positive lags.

### 2.2 Autocorrelators

Autocorrelators are used with single dish telescopes as a method of determining the spectral content or power spectral density of a particular signal. The power spectral density of a sampled signal $x(k\Delta t)$ with a spectrum $X(m\Delta f)$ is given by $|X(m\Delta f)|^2$. The power spectral density is typically calculated by FFTing an $N$-point estimate of the autocorrelation of $x(k\Delta t)$. Because the power spectral density is an even and real valued frequency function, the autocorrelation is also an even and real valued temporal function, this means that only the positive lags of the autocorrelation need to be calculated to fully determine the power spectral density. Once the positive lags of the autocorrelation have been determined, then the power spectral density is fully specified by the real valued positive spectral samples of the Fourier transform of the positive lags of the autocorrelation. Because only real valued positive spectral samples have to be calculated, no complex butterflies are necessary, so algorithms an order of magnitude faster than a full complex FFT can be used.

Fig. 2 shows the basic prompt and delay element in an autocorrelator. Given the definition of positive and negative lags as per (2) and Fig. 1(b), then Fig. 2 inherently generates positive lags only, i.e., if a signal $x(t_k)$ is connected to both the prompt and delay inputs of Fig. 2, then Fig. 2 generates the autocorrelation function

$$r_{xx}[n] = \sum_k x[k]x[k-n]$$  \hfill (7)

for lags $n \in [0, N-1]$.

The real valued Fourier transform of the autocorrelation function gives the power spectral density (for positive spectral samples only), i.e.,

$$R_{xx}[m] = |X[m]|^2 = r_{xx}[0] + 2 \sum_{n=1}^{N-1} r_{xx}[n] \cdot \cos\left(\frac{2\pi mn}{2N}\right)$$  \hfill (8)
where \( m \in [0, N - 1] \). Lag zero is pulled out of the sum, as it should only contribute once to the Fourier transform, the factor of 2 outside the sum is due to the fact that lags \( n \in [1, N - 1] \) are identical to lags \( n \in [-N - 1, -1] \), the denominator in the cosine argument is \( 2N \), as symmetry implies that there are \( 2N \) lags, and lag \(-N\) is implied a value of zero.

An FFT can be used to obtain identical results to (8) by extracting the real part of the \( N \) positive frequency components of the FFT of the length \( 2N \) vector given by

\[
\begin{align*}
\hat{r}'_{xx}[m] &= \left\{ \begin{array}{ll}
    r_{xx}[n] & \text{for } m \in [0, N - 1], \\
    0 & \text{for } m = N, \\
    r_{xx}[N - 1 - n] & \text{for } m \in [N + 1, 2N - 1],
\end{array} \right. \\
\end{align*}
\tag{9}
\]

where \( m \in [0, 2N - 1] \) and \( n \in [0, N - 1] \).

2.3 Construction of a cross correlator using the prompt and delay elements of an autocorrelator

If signals \( x(t_k) \) and \( y(t_k) \) are connected to the prompt and delay inputs of an autocorrelator like the one in Fig. 2, then lags \( n \in [0, N - 1] \) are calculated \([8,9,12]\). If the signals \( x(t_k) \) and \( y(t_k) \) are also connected to another autocorrelator, with the inputs reversed, then lags \( n \in [-N - 1, 0] \) are calculated. With this scheme, lag 0 would be calculated twice and lag \(-N\) would never calculated. To produce \( 2N \) independent lags, i.e., \( n \in [-N, N - 1] \), the signal \( x(t_k) \) is delayed by a single sample before entering the delay chain on the autocorrelator calculating the negative lags (lags \( n \in [-N, -1] \) are then calculated). Figure 3 shows the setup of the two autocorrelators and the lags calculated. Note that an autocorrelator with a switchable delay element has been used to calculate both positive and negative lags.

If such a switchable delay element does not exist in the autocorrelator cell, then the delay can be added externally into the \( x(t_k) \) path. Alternatively, delay 0 can be calculated twice and delay \(-N\) not at all. The double calculation of the zeroth lag can be used as an error checking measure—if lag zero calculated by each autocorrelator does not match, then there must be an error in the system. The removal of the \(-N\)th lag has little effect on the final resolution, but it is important that the \(-N\)th lag is given a zero value before performing an FFT on the data. The reintroduction of the \(-N\)th lag ensures correct phase estimates in the complex spectrum produced using the FFT (an
Figure 3: Construction of a cross correlator using two autocorrelators (note the switchable delay element). Shown either side of the cross correlator are the lags calculated by each path at the time instant $t_k$, i.e., for negative lags $x[k-n-1]y[k]$ and for positive lags $x[k]y[k-n]$ where $n \in [0, N-1]$ and $2N$ is the total number of lags in the cross correlator.
2N-point FFT references phase in the Fourier domain to the \( N + 1 \) temporal sample, i.e., the zeroth lag.

## 2.4 Construction of a correlator using counter-shifting input signals

Figure 4 shows the internal architecture of an \( N = 8 \) lag cross correlator based on counter-shifting delay lines (registers) \[1\]. If the \( N \) multiplier-accumulators in the cross correlator are numbered \( n = -N/2 \) to \( N/2 - 1 \), then the multiplier-accumulator numbering reflects the lag number that a particular multiplier-accumulator calculates. That is, the even numbered multiplier-accumulators calculate the \textit{positive and negative} even lags given by

\[
\begin{align*}
    r_{xy}[2m - N/2] & = \sum_k x[k + m - N/2]y[k - m] \\
    \text{and the odd numbered multiplier-accumulators calculate the \textit{positive and negative} odd lags given by} & \\
    r_{xy}[2m + 1 - N/2] & = \sum_k x[k + m + 1 - N/2]y[k - m],
\end{align*}
\]

where \( m \in [0, N/2 - 1] \). Cascading of this form of chip simply relocates the location of the lag origin, i.e., if two of these hypothetical 8 lag cross correlators were cascaded, then the pair would calculate the 16 lags -8 to 7.

The counter-shifting cross correlator can be made into an autocorrelator by connecting the \( x \) delay line output to the \( y \) delay line input. With this configuration, the \( N \) multiplier-accumulators from left to right in Fig. 4 calculate lags \( n \in [0, N - 1] \).
Figure 4: Internal architecture of an $N = 8$ lag cross correlator based on counter-shifting input signals. Shown below the cross correlator are the odd and even lags calculated by the odd and even multiplier-accumulators at time instant $t_k$, i.e., for odd lags $x[k + m + 1 - N/2]y[k - m]$ and for even lags $x[k + m - N/2]y[k - m]$ where $m \in [0, N/2 - 1]$. 
3 Architectural options

In a typical auto- or cross correlator system, the bandwidth available from the telescope receiver(s) is larger than a digitizer can sample directly. For example, the IF bandwidth at OVRO is 4GHz and will soon be 8GHz, a digitizer would need to sample at 8GHz or 16GHz respectively to adequately sample this IF signal. To sample the full IF bandwidth using digitizers with more modest sampling frequencies requires the use of analog filters and downconverters. The analog filter bandpass filters the IF to a bandwidth that is half the digitizer sample frequency, while the downconverter translates the filtered IF signal down to baseband. To calculate the power spectral density across the full IF bandwidth, multiple non-overlapping (or slightly overlapping) filters are used to filter the IF into sub-bands. Each sub-band is downconverted to baseband, digitized, and correlated. A digital correlator system that uses this form of analog-digital parallel processing is referred to as a hybrid digital correlator.

The digitized data can be processed by a digital correlator running at the same clock frequency as the digitizer, or further digital parallel processing schemes can be employed. The two digital parallel processing schemes commonly used are time demultiplexing and the use of deep memory to store digitizer samples. In a time demultiplexed system, the digitizer output data is time demultiplexed into a number of parallel streams and the correlation lags are calculated by a square array of correlator chips. In the deep memory model, the digitized data is stored in RAM for multiple correlator chips to access. Each correlator accesses a different segment of memory at a slower clock frequency than the digitizer feeding the RAM. Both models are shown graphically in Fig. 5. The following sections develop the two models and compare their operation.

A hybrid spectral line correlator is currently used by OVRO, time demultiplexing is currently used by BIMA, IRAM, and the SMA, while a correlator based on the deep memory model is being designed for the GBT, and is being considered for the MMA.

3.1 Time demultiplexed correlator systems

When a digitizer is available that has a clock frequency that exceeds the correlator chip clock frequency by a factor of two or more, then instead of slowing the digitizer clock to that of the correlator, a time demultiplexed scheme can be used [1, 10, 12]. In the time demultiplexed scheme, the digitizer and correlator clock frequencies are adjusted so that the digitizer clock is an integer multiple of the correlator clock. If the maximum operating frequency of the digitizer is $f_D$ and the maximum operating frequency of the correlator is $f_C$, then the ratio of these parameters determines the time demultiplexing factor; that is,

$$D = \frac{f_D}{f_C}.$$  \hspace{1cm} (12)

If $D$ is not an integer, then one can either; (a) decrease the digitizer clock frequency until $D$ is an integer, or (b) decrease the correlator clock frequency until $D$ is an integer. Since decreasing the digitizer clock frequency reduces the processed band size $B$, we chose to decrease the correlator clock frequency, hence $D$ is calculated via

$$D = \text{round}_\text{up} \left( \frac{f_D}{f_C} \right).$$  \hspace{1cm} (13)

For example, if a 1GHz clock frequency digitizer and a 130MHz clock frequency correlator chip are available, then $D = 8$ and the correlator clock frequency is reduced to 125MHz.

A correlator integrates the output of a digitizer (two digitizers for a cross correlator) for a integration period $T$. This period corresponds to $K = f_D T$ samples or clock pulses to the digitizer and $M = f_C T$ clock pulses to the correlator (where $f_C$ has been adjusted to produce an integer value for $D$). Since the correlator must process all of the samples from the digitizer, clearly $K = DM$, i.e.,
Figure 5: Calculation of the correlation function via the parallel processing methods of (a) time demultiplexing and (b) deep memory. In (a) the digitized data is time demultiplexed into \( D \) parallel streams of data at a lower clock frequency. The cross correlation between the two input signals is then calculated by a square array of \( D^2 \) correlator chips. In (b) the digitized data is stored in double buffered memory banks and \( D \) correlator chips process the data simultaneously. (Note that to obtain equivalent spectral resolution, each correlator chip (CC) in option (b) must have \( D \) times the number of lags as the chips in (a), i.e., the same total number of lags are used in either design). The internal architecture assumed for the cross correlator chip is shown in Fig. 6.
Figure 6: The internal architecture for the example cross correlator chip (the CC blocks) in Fig. 5. The chip calculates the function 
\[ r_{ab}[n] = \sum_{k} a[k] b[k - n] \] for \( n \in [-N_C/2, N_C/2 - 1] \), where \( N_C \) is the number of lags per chip, and \( \Delta t \) is the inverse of the correlator clock frequency. (Fig. 3 shows the instantaneous lag calculated by each multiplier-accumulator).
Figure 7: A time demultiplex by 8 correlator. During every correlator clock period, \( D \) digitizer samples enter the shift registers at the digitizer clock frequency \( f_D \). These \( D \) samples then exit the shift register at the correlator clock frequency \( f_C \). The sample indices are all shown relative to the digitizer sample period. The relationship between the digitizer sample indice \( k \) and the correlator sample indice \( m \) is shown at the bottom of the figure. Each correlator in the \( 8 \times 8 \) array has \( N \) lags. For an autocorrelator these lags are \( n \in [0, N-1] \), and for a cross correlator \( n \in [-N/2, N/2-1] \). Each correlator shows the indices of the lags it produces. For an autocorrelator, each bold line along the diagonal indicates an additional delay element. For a cross correlator, the bold lines indicate that the inputs to all the correlators above the diagonal are reversed.

During each clock pulse of the correlator, \( D \) samples from the digitizer(s) are processed. Figure 7 shows an example of an autocorrelator and a cross correlator built using 1GHz clock frequency digitizers and 125MHz clock frequency correlators.

The lags calculated by each correlator in the \( 8 \times 8 \) array of correlators in Fig. 7 are determined
as follows. Repeating (2) with explicit dependence on the correlator clock period $\Delta_{tc} = 1/f_C$:

$$r_{xy}(n\Delta_{tc}) = \sum_k x(k\Delta_{tc})y([k - n]\Delta_{tc}).$$ \hspace{1cm} (14)

The indice $k \in [0, K - 1]$ in this equation assumes a digitizer operating at the same frequency as the correlator. Taking account of the faster digitizer by replacing the correlator period with the digitizer period and replacing the indice $k$ by the correlator indice $m \in [0, M - 1]$ gives

$$r_{xy}(8n\Delta_{td}) = \sum_m x(8m\Delta_{td})y(8[m - n]\Delta_{td}),$$ \hspace{1cm} (15)

or equivalently,

$$r_{xy}[8n] = \sum_m x[8m]y[8m - 8n].$$ \hspace{1cm} (16)

where sample indices are now assumed relative to the digitizer. Equation (16) gives the lags produced by the correlator that is located in the bottom right of the $8 \times 8$ array shown in Fig. 7. Delays introduced by the time demultiplexing shift register alter the lags produced by each correlator in the array. The lags calculated by each correlator in the array are determined according to

$$\text{lag number} = \begin{cases} 
8n + x \text{ index} - y \text{ index} & \text{for correlators on and below the diagonal} \\
8n + 8 + x \text{ index} - y \text{ index} & \text{for correlators above the diagonal}
\end{cases}.$$ \hspace{1cm} (17)

For example, the bottom left correlator in Fig. 7 correlates $x[8m + 7]$ with $y[8m]$ and hence produces lags $r_{xy}[8n + 7]$. Fig. 7 clearly shows that there are only 8 independent lag sets calculated in the correlator array.

If each correlator in Fig. 7 has $N = 8$ lags, then because 8 independent lag sets are calculated by the array, a total of 64 independent lags are calculated. For an autocorrelator, $n \in [0, N - 1]$, so the diagonals in the correlator array calculate lags $\{8n = 0, 8, \ldots, 56\}, \{8n + 1 = 1, 9, \ldots, 57\}, \ldots, \{8n + 7 = 7, 15, \ldots, 63\}$. For a cross correlator $n \in [-N/2, N/2 - 1]$, so the diagonals calculate lags $\{8n = -32, -24, \ldots, 16, 24\}, \{8n + 1 = -31, -23, \ldots, 17, 25\}, \ldots, \{8n + 7 = -25, -17, \ldots, 23, 31\}$.

To make a cross correlator using two time demultiplexed autocorrelators, the positive lags are calculated using the time demultiplexed correlator just discussed and the negative lags are calculated with a slightly modified autocorrelator. If the delay elements shown after the diagonal in Fig. 7 are moved so that they precede the diagonal, then the diagonal elements produce lags $8n + 8$. If such an autocorrelator is used with the inputs reversed, then this modified autocorrelator calculates the negative lags without repeating the zeroth lag.

### 3.2 Deep memory correlator systems

In a deep memory correlator with a basic integration time $T$, $K = f_cT$ samples are processed per integration. These $K$ samples are piped into a RAM before the correlator processes them (double or more buffering of the RAM is required). Once the $K$ samples in RAM are valid, $D$ non-overlapping segments of $M = f_cT$ samples are shifted out to $D$ parallel correlators. Each correlator produces the same lag result from different segments of the original digitized waveform. Hence, if each of the $D$ correlators used in the deep memory system have $N$ lags each, then only $N$ lags are produced in total.

The application of the deep memory model to autocorrelators and cross correlators is fairly obvious.
Table 1: Correlator design parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B$</td>
<td>Hz</td>
<td>bandwidth of IF sub-band</td>
</tr>
<tr>
<td>$f_D = 2B$</td>
<td>Hz</td>
<td>digitizer clock frequency (Nyquist rate sampling)</td>
</tr>
<tr>
<td>$f_C$</td>
<td>Hz</td>
<td>correlator clock frequency</td>
</tr>
<tr>
<td>$N_B$</td>
<td></td>
<td>basic number of correlator chips required</td>
</tr>
<tr>
<td>$N_{CC}$</td>
<td></td>
<td>number of cascaded chips</td>
</tr>
<tr>
<td>$N_T$</td>
<td></td>
<td>total number of correlator chips used</td>
</tr>
<tr>
<td>$N_C$</td>
<td></td>
<td>number of lags per correlator chip</td>
</tr>
<tr>
<td>$N_R$</td>
<td></td>
<td>number of lags required</td>
</tr>
<tr>
<td>$N_I$</td>
<td></td>
<td>number of independent lags obtained</td>
</tr>
<tr>
<td>$\Delta f = 2B/N_I$</td>
<td>Hz</td>
<td>spectral sample spacing</td>
</tr>
<tr>
<td>$D = \text{round}_\text{up}(f_D/f_C)$</td>
<td>-</td>
<td>digitizer to correlator chip clock ratio</td>
</tr>
<tr>
<td>$\gamma = N_I/N_R$</td>
<td>-</td>
<td>overkill ratio</td>
</tr>
</tbody>
</table>

### 3.3 Cross correlator design equations

This section details the steps required to evaluate whether time demultiplexing or the deep memory model will meet the spectral resolution requirements of a cross correlator design (autocorrelator designs use similar equations, but since they do not need to calculate negative lags, their spectral resolving capability, for the same number of lags, is twice as good as a cross correlator). Some of the relevant system parameters are shown in Table 1.

The basic requirement of any cross correlator is usually specified in terms of the spectral resolution it can obtain across the IF bandwidth, i.e., the basic requirement is to produce a cross power spectrum with a spectral sample spacing $\Delta f$ (depending on the windowing function used in the lag domain, the spectral resolution is typically on the order of twice $\Delta f$). If we consider that the IF bandwidth is filtered into sub-bands of bandwidth $B$, then each cross correlator must produce a spectral sample spacing of $\Delta f$ across bandwidth $B$. This spectral sample spacing is achieved using a digitizer with a clock frequency $f_D = 1/(2B)$ and a cross correlator chip with a clock frequency $f_C$ and $N_C$ lags.

If the digitizer clock frequency and the cross correlator clock frequency are the same, then the design is trivial—data from the digitizer is sent directly to one or more cross correlator chips. In this case, the spectral sample spacing in the Fourier transform of the cross correlation function is

$$\Delta f = \frac{2B}{N_I}$$

(18)

where $N_I = N_{CC}N_C$ and $N_{CC}$ is the number of cascaded correlator chips used to achieve the desired spectral resolution.

In cases where the digitizer clock frequency exceeds the cross correlator chip clock frequency by a factor of two or more, a digital parallel processing scheme is employed. The design of a cross correlator then follows the following steps:

1. Determine the number of lags required to obtain the desired spectral sample spacing; that is,

$$N_R = \frac{2B}{\Delta f}$$

(19)

2. Calculate $D$, the ratio of the digitizer clock frequency to the correlator clock frequency.
3. (a) For a time demultiplexed design, the basic number of correlator chips required is \( N_B = D^2 \), and the number of independent lags is \( N_1 = D N_C \).

(b) For a deep memory model based design, the basic number of correlator chips required is \( N_B = D \), and the number of independent lags available from this design is \( N_1 = N_C \) (each of the \( D \) chips calculate the same lags from different temporal samples).

4. Calculate the ratio of the number of independent lags to the number of lags required, i.e.,

\[
\gamma = \frac{N_1}{N_R}
\]  

(20)

For reasons that will become clear, this ratio is referred to as the \textit{overkill} ratio. If \( \gamma \approx 1 \), then the design produces the required spectral sample spacing, if \( \gamma < 1 \) then the next integer above \( (1/\gamma) \) gives the number of cascaded chips required to produce the desired spectral sample spacing, i.e., \( N_{CC} = \text{round}_{\uparrow}(1/\gamma) \), and if \( \gamma > 1 \), then the design exceeds the desired spectral sample spacing. If \( \gamma \) is greater than 1, by a small margin, then the design meets the specified spectral sample spacing. However, if \( \gamma \) is significantly larger than 1, then the design has overkilled on the spectral sample spacing and redesign may be necessary.

5. If \( \gamma \) was less than or equal to 1, then the total number of correlator chips required in each design is

\[
N_T = \begin{cases} 
D^2 N_{CC} & \text{time demultiplexing} \\
D N_{CC} & \text{deep memory}
\end{cases}
\]  

(21)

and the number of independent lags available in each design is

\[
N_1 = \begin{cases} 
D N_C N_{CC} & \text{time demultiplexing} \\
N_C N_{CC} & \text{deep memory}
\end{cases}
\]  

(22)

and the final spectral sample spacing is then,

\[
\Delta_f = \frac{2B}{N_1}.
\]  

(23)

To achieve equivalent spectral sample spacings between a time demultiplexed correlator and a deep memory correlator, the number of independent lags in each design must be equivalent. Equation (22) indicates that chips used in a deep memory model need to have \( D \) times the number of lags used in a time demultiplexed model (or that you need to cascade \( D \) chips in the deep memory model).

### 3.4 The deep memory model versus time demultiplexing—examples

The operation of the two models is compared in this section by way of several examples. Consider a system that has an IF bandwidth of 4GHz that is to be filtered into \( B = 500\text{MHz} \) bandwidth sub-bands, digitized at \( f_D = 1\text{GHz} \), and processed to a spectral sample spacing of \( \Delta_f \approx 2\text{MHz} \), with a basic integration time of \( T = 1\text{ms} \). There are two correlator chips available for the correlator design; the first has \( N_C = 512 \) lags and operates at \( f_C = 125\text{MHz} \), the other has \( N_C = 64 \) lags and operates at \( f_C = 500\text{MHz} \).

Table 2 summarizes the results of the design steps. The following text also briefly describes the design process:

**Example 1(a): time demultiplexing with the 512-lag/125MHz correlator**
Table 2: Correlator design examples–2MHz spectral sample spacing

<table>
<thead>
<tr>
<th>Step</th>
<th>Parameter</th>
<th>512 lags/125MHz</th>
<th>64 lags/500MHz</th>
<th>512 lags/125MHz</th>
<th>64 lags/500MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(N_R)</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>2</td>
<td>(D)</td>
<td>8</td>
<td>2</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>(N_B)</td>
<td>64</td>
<td>4</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>(N_I)</td>
<td>4096</td>
<td>128</td>
<td>512</td>
<td>64</td>
</tr>
<tr>
<td>4</td>
<td>(\gamma)</td>
<td>∼8</td>
<td>∼1/4</td>
<td>∼1</td>
<td>∼1/8</td>
</tr>
<tr>
<td></td>
<td>status</td>
<td>overkill</td>
<td>underkill</td>
<td>pass</td>
<td>underkill</td>
</tr>
<tr>
<td>5</td>
<td>(N_{CC})</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>(N_I)</td>
<td>64</td>
<td>16</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>(\Delta_f)</td>
<td>244kHz</td>
<td>2.0MHz</td>
<td>2.0MHz</td>
<td>2.0MHz</td>
</tr>
<tr>
<td></td>
<td>(\Delta_f)</td>
<td>244kHz</td>
<td>2.0MHz</td>
<td>2.0MHz</td>
<td>2.0MHz</td>
</tr>
</tbody>
</table>

The 1GHz 2-bit sample streams from each telescope need to be slowed to 125MHz, so they must each be time demultiplexed by \(D = 8\). The two 16-bit/125MHz data streams are processed by an array of 8×8 correlator chips. The correlated result is \(8 \times 512 = 4096\) independent lags across 1GHz, which gives a spectral sample spacing of \(\Delta_f = 244kHz\)—8 times finer than the required 2MHz sample spacing.

**Example 1(b): time demultiplexing with the 64-lag/500MHz correlator**

The 1GHz 2-bit sample streams from each telescope need to be slowed to 500MHz, so they must each be time demultiplexed by \(D = 2\). The two 4-bit/500MHz data streams are processed by an array of 2×2 correlator chips. The correlated result is \(2 \times 64 = 128\) independent lags across 1GHz. This gives a spectral sample spacing of \(\Delta_f = 7.8MHz\), which is 4 times coarser than the required value. Cascading 4 correlator chips behind each chip in the 2×2 array increases the number of independent lags to 512 and decreases the spectral sample spacing to 2.0MHz.

**Example 2(a): deep memory with the 512-lag/125MHz correlator**

In the basic integration time of 1ms, at a sample frequency of 1GHz, there are \(K = 1M\) samples. Given that each correlator chip operates at 125MHz, in 1ms it can process \(M = 125k\) samples. Thus, 8 correlator chips working on 8 125k non-overlapping samples can calculate 512 lags for 1ms worth of data (the computer reading the 512 lags from each of the 8 chips will have to accumulate the data to give the final 512 independent lags). The spectral sample spacing is then 2.0MHz.

**Example 2(b): deep memory with the 64-lag/500MHz correlator**

In the basic integration time of 1ms, at a sample frequency of 1GHz, there are \(K = 1M\) samples. Given that each correlator chip operates at 500MHz, in 1ms it can process \(M = 500k\) samples. Thus, 2 correlator chips working on 2 500k non-overlapping samples can calculate 64 lags for 1ms worth of data. The spectral sample spacing is then 15.6MHz, which is just under 8 times coarser than required. If you cascade 8 of these chips to work on each of the 2 500k sample blocks (512 independent lags, 16 chips total), then the final spectrum sample spacing becomes 2.0MHz.

There are several observations that can be made using these examples. In Example 1(b) and 2(b), the *same number of correlator chips* are used to obtain the *same spectral sample spacing*. There is not a quadratic scaling of correlator chips as has been indicated by designers preferring the deep memory model. In fact, because the deep memory model requires the memory segment and associated data resorting logic, it is the more expensive option! The analysis in the previous section gives the correct relationship between the two designs: if the chips in a time demultiplexed model have \(N_C\) lags, then an equivalent deep memory model implementation needs to have correlator chips...
with $D N_C$ lags per chip (or needs to cascade $D$ chips with $N_C$ lags).

Example 1(a) stands out as being the only scheme not able to be manipulated to produce the desired output spectral sample spacing. This is because the relatively slow correlator chip clock frequency combined with the high number of lags per chip leads to a high number of independent lags—giving a spectral sampling finer than desired. To make Example 1(a) be the time demultiplexed implementation of Example 2(a), then the number of lags on the slower chip needs to be dropped by $D = 8$ to $512/8 = 64$ lags per chip. A time demultiplexed design with a 64 lag, 125MHz clock frequency correlator chip would give the desired spectral sample spacing of about 2MHz (a 64 lag, 125MHz clock frequency correlator is consistent with the capabilities of current FPGA technology).

The choice of which scheme to use is dictated by the ratio of the digitizer clock speed to the correlator clock speed and the number of lags per chip. For example, the Canaris chip used by the SMA has a 125MHz clock speed and 512 lags. A low resolution system would have to be built with the chip using the deep memory model (because of the results of Example 1(a)). Higher resolution can then be obtained by cascading chips (increasing their lag depth). This is the design chosen by both the GBT and SMA (the GBT Canaris chip has 1024 lags).

Since time demultiplexing and deep memory use the same number of correlator chips to obtain the same spectral resolution, then in cases where the result of Example 1(b) does not occur, time demultiplexing is the obvious choice for parallel digital processing. The choice of the time demultiplexing method removes the cost associated with the deep memory and the associated data resorting logic. If higher resolution is required, then each chip at each intersection in the array of chips in Fig. 5(a) can have its lag length increased by locating cascaded chips at each intersection.

An important point that should be noted before leaving this section is that the deep memory and time demultiplexing architectures just described are typically implemented using ASICs (custom integrated circuits) that already exist. Section 5 details the implementation of a time demultiplexed correlator in an FPGA. Due to the redundancy present in time demultiplexed scheme, a better resource utilization can be achieved in an FPGA than can be achieved in a full custom ASIC. For example, the diagonal elements shown in Fig. 7 all calculate the same lag. If this correlator was implemented using existing correlator chips, then the control system has to read 8 separate counters and sum the results to obtain the correct count for each lag. In an FPGA, the multiplier-adders calculating the same lags can have their carry outputs cascaded together with the resulting single carry output driving a single counter. The logic cells saved by performing this carry cascade can be used to implement more lags.

### 4 Digital correlator statistics

This section discusses the signal statistics of band-limited, sampled digital correlator systems. Systems utilizing full precision (infinite number of bits), 1-bit, and 2-bit sampling are covered. For this section, it is assumed that the correlator operates at the same clock frequency as the digitizer(s) (this assumption does not result in any loss of generality).

#### 4.1 Full precision sampling

Consider an autocorrelator estimating $N$ lags of the autocorrelation function of a wide sense stationary process $X(t)$ that is modeled as a zero mean white noise process with standard deviation $\sigma$. It is assumed that the process has been low-pass filtered to bandwidth $B$ and is sampled at $f_D = 2B$ by a digitizer with a high number of bits (so that quantization noise need not be considered). Figure 8 shows an ensemble of the autocorrelation functions generated in the autocorrelator. The top plot in the figure represents a sample function of the autocorrelation of the random process $X(t)$ that

---

1 RAM chips that can be written/read at 500MHz may not be available, so a deep memory application may not even be possible with high clock frequency correlator chips
Figure 8: An ensemble of sample autocorrelation functions. The top plot in the figure represents a sample function of the autocorrelation of the random process \( X(t) \) that is generated at time \( t = t_j \). This sample function is generated in the autocorrelator by multiplying the sample \( x[j] \) of the sample function \( x(t) \) first by itself, then by the previous samples \( x[j-n] \) for \( n \in [1,N-1] \), where \( N \) is the number of lags in the autocorrelator. The other \( K-1 \) sample autocorrelation functions are generated in a similar manner. Ensemble averaging of the \( K \) sample functions gives an improved estimate of the autocorrelation of the random process \( X(t) \).

is generated at time \( t = t_j \). This sample function is generated in the autocorrelator by multiplying the sample \( x[j] \) of the sample function \( x(t) \) first by itself, then by the previous samples \( x[j-n] \) for \( n \in [1,N-1] \), where \( N \) is the number of lags in the autocorrelator. These \( N \) samples are located in the delay line of the autocorrelator.

If we begin accumulating estimates of the autocorrelation function at time \( t_j \), and finish when \( K \) sample functions have been accumulated (i.e., samples \( t_j \) to \( t_j+K-1 \) have entered the autocorrelator delay line), an unbiased estimate of the autocorrelation function of the stationary random process \( X(t) \) is then

\[
\mu_{r_{XX}}[n] = E[X(t_j+k)X(t_j+n)] \\
= \langle x[j+k]x[j+k-n] \rangle \\
= \frac{1}{K} \sum_k x[j+k]x[j+k-n]
\]

(24)

where \( K = f_DT \), \( k \in [0,K-1] \), \( T = t_{j+K-1} - t_j \) is the integration or ensembling averaging time, \( T_N = N/f_D \) is the temporal length of the autocorrelator, and \( T \gg T_N \) (so that a reasonable estimate is obtained). If the integration starts at time \( t = 0 \), then the index \( j = 0 \) and (24) takes on the familiar form of (2) (with a scale factor to account for the ensemble averaging).

Each measurement of the zeroth lag of the autocorrelation function corresponds to a measurement from a probability distribution \( Y(t) = X^2(t) \), where \( X(t) \) is the zero-mean Gaussian random process generating the power signal \( x(t) \). Since each sample \( y(t) \) from the probability distribution \( Y(t) \) corresponds to the square of each sample \( x(t) \) from the distribution \( X(t) \), it is clear that the distribution function for \( Y(t) \) is non-zero only for \( y \geq 0 \). The distribution function for \( Y(t) \) is derived on p432 of [6] and is repeated here:

\[
f_Y(y) = \begin{cases} \\
\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y}{2\sigma^2}\right), & y \geq 0 \\
0, & \text{otherwise.}
\end{cases}
\]

(25)
This probability function is a \textit{chi-squared density function} with a mean \( \mu_Y = \sigma^2 \) and standard deviation \( \sigma_Y = \sqrt{2} \sigma^2 \).

In the zeroth lag of the estimate of the autocorrelation function, \( K \) samples from \( Y_k \) (where \( k \in [0, K - 1] \)) independent chi-squared distributions have been ensemble averaged. This produces a new distribution function

\[
Z = \frac{1}{K} \sum_{k=0}^{K-1} Y_k
\]  

(26)

The central limit theorem [6, p428] then tells us that the distribution \( Z \) is Gaussian, with mean \( \mu_Z = \mu_Y = \sigma^2 \) and standard deviation \( \sigma_Z = \sigma_Y / \sqrt{K} = \sqrt{2/K} \sigma^2 \).

Each measurement of any lag other than the zeroth lag of the autocorrelation function corresponds to a measurement of the probability distribution \( Z = X Y \), where \( X \) and \( Y \) are independent zero-mean Gaussian random processes with standard deviations \( \sigma_X = \sigma_Y = \sigma \). Since the probability distributions are independent, the expected value of \( Z \), namely \( \mu_Z \), is zero [6, p424]. The expected value of \( Z^2 \) is \( E[(XY)^2] \), and since \( X \) and \( Y \) are independent, \( E[Z^2] = E[X^2] \cdot E[Y^2] \), so \( \sigma_Z = \sigma_X \sigma_Y = \sigma^2 \). Ensemble averaging of \( K \) results, results in a distribution function \( W \), with zero-mean and standard deviation \( \sigma_W = \sigma^2 / \sqrt{K} \).

The autocorrelation estimate, calculated as an ensemble average of \( K \) temporal samples of the waveform \( x(t) \) produced by the zero-mean Gaussian random process \( X(t) \) with standard deviation \( \sigma \) is

\[
r_{XX}[n] = \mu_{r_{XX}}[n] \pm \sigma_{r_{XX}}[n] \\
= \begin{cases} \sigma^2 \pm \sigma^2 \cdot \frac{2}{\sqrt{K}} & n = 0 \\ 0 \pm \sigma^2 \cdot \frac{1}{\sqrt{K}} & n \neq 0 \end{cases}
\]  

(27)

The autocorrelation estimate converts to correlation coefficients via [6, p424]

\[
\rho_{XX}[n] = \frac{r_{XX}[n]}{\sigma^2}.
\]

(28)

Hence, the expected values of the correlation coefficients are

\[
\rho_{XX}[n] = \mu_{r_{XX}}[n] \pm \sigma_{r_{XX}}[n] \\
= \begin{cases} 1 \pm \frac{2}{\sqrt{K}} & n = 0 \\ 0 \pm \frac{1}{\sqrt{K}} & n \neq 0 \end{cases}
\]  

(29)

Now let us consider the case of a cross correlator. If \( X \) and \( Y \) are joint Gaussian random variables with zero mean and variance \( \sigma^2 \), their probabilities of occurrence are governed by the Bivariate Gaussian probability distribution [9][p211]; that is,

\[
f_{XY}(x, y) = \frac{1}{2\pi \sigma_x \sigma_y \sqrt{1 - \rho^2}} \cdot \exp \left[ -\frac{(x^2 + y^2 - 2\rho xy)}{2\sigma^2(1 - \rho^2)} \right]
\]

(30)

where the cross correlation coefficient [9][p211,p215]

\[
\rho = \frac{E[XY]}{\sqrt{E[X^2]E[Y^2]}} = \frac{\langle xy \rangle}{\sqrt{\langle x^2 \rangle \langle y^2 \rangle}} = \frac{\langle xy \rangle}{\sigma^2}
\]

(31)
and $-1 < \rho < 1$. For $\rho \ll 1$, the exponent can be expanded giving

$$f_{XY}(x, y) \approx \left[ \frac{1}{\sigma \sqrt{2\pi}} \cdot \exp \left( -\frac{x^2}{2\sigma^2} \right) \right] \left[ \frac{1}{\sigma \sqrt{2\pi}} \cdot \exp \left( -\frac{y^2}{2\sigma^2} \right) \right] \left( 1 + \frac{\rho_{X,Y}}{\sigma^2} \right).$$  \hspace{1cm} (32)

For $\rho = 0$, the expression is simply the product of two Gaussian functions.

The cross correlation coefficient in (31) is the expected value of the maximum of the correlation function. When using test noise sources, or imaging a point source with the appropriate delay correction, this maximum will occur at the zeroth lag. However, when imaging a distributed source using an interferometer, this maximum is not necessarily at the zeroth lag. If full precision sampling is used and we assume that the maximum is at the zeroth lag, then the zeroth lag of the $K$-ensemble averaged correlator is directly proportional to $\rho$; that is,

$$r_{XY} = \mu_{r_{XY}} \pm \sigma_{r_{XY}}$$

$$= \rho \sigma^2 \pm \sigma^2 \cdot \sqrt{\frac{\rho^2 + 1}{K}} \hspace{1cm} (33)$$

where

$$\mu_{r_{XY}} = E[XY]$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} xy f_{XY}(x, y) \, dx \, dy$$

$$= \rho \sigma^2, \hspace{1cm} (34)$$

$$\sigma_{r_{XY}}^2 = \frac{1}{K} \left\{ E[(XY)^2] - (E[XY])^2 \right\}, \hspace{1cm} (35)$$

and ([9, p215])

$$E[(XY)^2] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^2 y^2 f_{XY}(x, y) \, dx \, dy$$

$$= \sigma^4 (1 + 2\rho^2) \hspace{1cm} (36)$$

Normalizing (33) to a correlation coefficient gives

$$\rho_{XY} = \rho \pm \sqrt{\frac{\rho^2 + 1}{K}} \hspace{1cm} (37)$$

where it is not necessary to indicate $n$, as when analyzing the autocorrelator $n = 0$ implied maximum correlation ($\rho = 1$) and $n \neq 0$ implied a lag other than the zeroth (where $\rho = 0$), i.e., (29) contains the two extremes of (37).

The signal-to-noise ratio (SNR) of the measured correlation coefficient is defined as

$$\text{SNR} = \frac{\mu_{r_{XY}}}{\sigma_{r_{XY}}}. \hspace{1cm} (38)$$

For full precision sampling, the SNR is then

$$\text{SNR} = \rho \cdot \sqrt{\frac{K}{\rho^2 + 1}} \hspace{1cm} \approx \rho \sqrt{K} \text{ for } \rho \ll 1. \hspace{1cm} (39)$$
For an $N$ point autocorrelation function ($n \in [0, N-1]$), the Fourier transform of the measured correlation coefficients in (29) is calculated via (8); that is,

$$P_{XX}[m] = \mu_{P_{XX}}[m] \pm \sigma_{P_{XX}}[m]$$

$$= \begin{cases} 
1 \pm \sqrt{\frac{4N-2}{K}} & m = 0 \\
1 \pm \sqrt{\frac{2N-2}{K}} & m \neq 0,
\end{cases}$$

$$\approx \begin{cases} 
1 \pm \sqrt{\frac{4N}{K}} & m = 0 \\
1 \pm \sqrt{\frac{2N}{K}} & m \neq 0,
\end{cases}$$

where $m \in [0, N-1]$,

$$\mu_{P_{XX}}[m] = \mu_{\rho_{XX}}[0] + 2 \cdot \sum_{n=1}^{N-1} \mu_{\rho_{XX}}[n] \cos \left(2\pi \frac{mn}{2N}\right),$$

and since the correlation coefficients are independent, their variances add, i.e.,

$$\sigma_{P_{XX}}^2[m] = \sigma_{\rho_{XX}}^2[0] + 4 \cdot \sum_{n=1}^{N-1} \sigma_{\rho_{XX}}^2[n] \cos^2 \left(2\pi \frac{mn}{2N}\right)$$

where

$$\sum_{n=1}^{N-1} \cos^2 \left(2\pi \frac{mn}{2N}\right) = \begin{cases} 
N-1 & m = 0 \\
\frac{N-1}{2} & m \neq 0.
\end{cases}$$

For an $N$-point cross correlation function ($n \in [-N/2, N/2-1]$) with cross correlation coefficients

$$\rho_{XY}[n] = \mu_{\rho_{XY}}[n] \pm \sigma_{\rho_{XY}}[n]$$

$$= \begin{cases} 
\rho \pm \sqrt{\rho^2 + 1} & n = 0 \\
0 \pm \frac{1}{\sqrt{K}} & n \neq 0,
\end{cases}$$

the Fourier transform of the correlation coefficients is

$$P_{XY}[m] = \mu_{P_{XY}}[m] \pm \sigma_{P_{XY}}[m]$$

$$= \begin{cases} 
\rho \pm \sqrt{\rho^2 + 2N - 2} & m = 0 \\
\rho \pm \sqrt{\rho^2 + N - 3 + (-1)^m} & m \neq 0,
\end{cases}$$

$$\approx \begin{cases} 
\rho \pm \sqrt{\frac{2N}{K}} & m = 0 \\
\rho \pm \sqrt{\frac{N}{K}} & m \neq 0,
\end{cases}$$

where

$$\mu_{P_{XY}}[m] = \sum_{n=0}^{N-1} \mu_{\rho_{XY}}[n] \cdot \exp \left(-j2\pi \frac{mn}{N}\right)$$

$$= \mu_{\rho_{XY}}[0] + \sum_{n=1}^{N-1} \mu_{\rho_{XY}}[n] \cdot \exp \left(-j2\pi \frac{mn}{N}\right),$$

(41)
and since we have assumed that the correlation peak lies at \( n = 0 \), lags \( n \in [1, N/2 - 1] \) have the same statistics as lags \( n \in [N/2 + 1, N - 1] \) (although they are time reversed) and the correlation coefficient spectrum is real, with variance

\[
\sigma_{\rho_{XY}}^2[m] = \sigma_{\rho_{XY}}^2[0] + (-1)^m \sigma_{\rho_{XY}}^2[N/2] + 4 \cdot \sum_{n=1}^{N/2-1} \sigma_{\rho_{XY}}^2[n] \cdot \cos^2 \left( 2\pi \frac{mn}{N} \right),
\]

(47)

where

\[
\sum_{n=1}^{N/2-1} \cos^2 \left( 2\pi \frac{mn}{2N} \right) = \begin{cases} 
\frac{N}{4} - 1 & m = 0 \\
\frac{N}{2} - 1 & m \neq 0.
\end{cases}
\]

(48)

For a general cross correlation, the peak will not occur at the zeroth lag. If for example, the lag peak moves to positive lag \( n_1 \), then the statistically dependent lags become statistically independent lags, leaving \( N/2 - n_1 \) repeat lags either side of the lag peak. Movement of the lag peak to the extreme of \( n_1 = N/2 - 1 \), makes all other lags statistically independent. This statistical independence reduces the variance of the cross correlation spectrum by \( \sim \sqrt{2} \). When the correlation lag is not located at \( n = 0 \), the mean and variance of the magnitude of the Fourier transform of the correlation coefficients is given approximately by (45).

Calibration of a correlator is usually performed using two separate noise sources that contain a small component of correlated noise. The delay from each noise source to the correlator is usually equivalent, so the lag maximum occurs at the origin.

### 4.2 1-bit sampling with 2-level quantization

A 1-bit correlator quantizes the input waveforms to two levels, i.e., the input voltage is either positive or negative. These two values are given the weights -1 and 1 [11]. Table 3 shows this quantization scheme for an input voltage \( x \). Table 4 shows the multiplication scheme for two inputs \( x \) and \( y \) both digitized to 1-bit. The probabilities shown in Table 4 are obtained by integrating (30) over limits implied by the input voltage ranges shown in Table 4 (see p245 [9]). The 2-level cross correlation of the two signals \( x \) and \( y \) is

\[
r_2 = \mu_{r_2} \pm \sigma_{r_2} = \frac{2}{\pi} \cdot \sin^{-1} \rho \pm \sqrt{1 - \left( \frac{2}{\pi} \cdot \sin^{-1} \rho \right)^2},
\]

(49)

where

\[
\mu_{r_2} = E\{r_2\} = \sum_{ij} w_{ij} P_{ij}
\]

(50)

and

\[
\sigma_{r_2}^2 = \text{Var}\{r_2[n]\} = \sum_{ij} (w_{ij} - \mu_{r_2})^2 P_{ij} = \sum_{ij} w_{ij}^2 P_{ij} - \mu_{r_2}^2.
\]

(51)

For an \( N \)-lag autocorrelator that ensemble averages \( K \) autocorrelations, (49) evaluates to

\[
\begin{align*}
r_2[n] &= \mu_{r_2} + \sigma_{r_2}[n] \\
&= \begin{cases} 
1 & n = 0 \quad \rho = 1 \\
0 & n \neq 0 \quad \rho = 0.
\end{cases}
\end{align*}
\]

(52)
Table 3: 1-bit quantization scheme

<table>
<thead>
<tr>
<th>Input voltage range</th>
<th>State, $i$</th>
<th>Weight, $w_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x &lt; 0$</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>$x &gt; 0$</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4: 1-bit multiplication scheme

<table>
<thead>
<tr>
<th>Input voltage range</th>
<th>State, $i,j$</th>
<th>Weight, $w_{ij}$</th>
<th>Probability, $P_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x &lt; 0, y &lt; 0$</td>
<td>00</td>
<td>1</td>
<td>$\frac{1}{2} + \frac{1}{2\pi} \sin^{-1} \rho$</td>
</tr>
<tr>
<td>$x &lt; 0, y &gt; 0$</td>
<td>01</td>
<td>$-1$</td>
<td>$\frac{1}{2} - \frac{1}{2\pi} \sin^{-1} \rho$</td>
</tr>
<tr>
<td>$x &gt; 0, y &lt; 0$</td>
<td>10</td>
<td>$-1$</td>
<td>$P_{01}$</td>
</tr>
<tr>
<td>$x &gt; 0, y &gt; 0$</td>
<td>11</td>
<td>1</td>
<td>$P_{00}$</td>
</tr>
</tbody>
</table>

where we have used $\sin^{-1}(1) = \pi/2$ and $\sin^{-1}(0) = 0$. Similarly, an $N$ lag cross correlator with $\rho \ll 1$ obtains

$$r_2[n] \approx \begin{cases} 
\frac{2}{\pi} \cdot \rho \pm \frac{1}{\sqrt{K}} & n = 0 \quad \rho \ll 1 \\
0 \pm \frac{1}{\sqrt{K}} & n \neq 0 \quad \rho = 0.
\end{cases} \quad (53)$$

To convert the measured correlations in (52) and (53) to correlation coefficients the amplitudes of the 1-bit correlation functions need to be transformed according to

$$\rho_2[n] = \sin \left( \frac{\pi}{2} r_2[n] \right) \approx \begin{cases} 
1 \pm 0 & n = 0 \quad \rho = 1 \\
\rho \pm \frac{\pi}{2} \frac{1}{\sqrt{K}} & n = 0 \quad \rho \ll 1 \\
0 \pm \frac{\pi}{2} \frac{1}{\sqrt{K}} & n \neq 0 \quad \rho = 0.
\end{cases} \quad (54)$$

For cross correlations in which $\rho$ is not small, but $K \gg 1/\rho^2$ (i.e., the variance is small), the increase in the variance of the measured correlation after the amplitude transform to correlation coefficients in (54) is given approximately by the slope of the transform function; that is,

$$\frac{d\rho}{dr_2} = \frac{\pi}{2} \cdot \sqrt{1 - \rho^2}. \quad (55)$$

The equation shows that the increase in variance is greatest for small correlations.

For cross correlations where $\rho \ll 1$, the signal-to-noise (of the lag maximum) of a 1-bit cross correlator is,

$$\text{SNR} = \rho \cdot \frac{2}{\pi} \cdot \sqrt{K}. \quad (56)$$

This SNR is $2/\pi = 0.64$ that of a full precision correlator (see (39)). This means that for a 1-bit correlator to achieve the same SNR as the full precision correlator, the integration time of the 1-bit
Figure 9: Block diagram of a typical 2-bit digitizer. The quantization counters count the number of 2-bit samples in each of the four quantization states over an integration period $T$. Optimal threshold voltages can be set using the relative counts of these four counters.

The Fourier transform of the 1-bit cross correlation coefficients in (54) for $N$ lags can be calculated in a similar manner to the Fourier transform of the full precision correlation coefficients calculated in (40) and (45); that is,

$$P_2[m] \approx \begin{cases} 
\rho \pm \frac{\pi}{2} \cdot \sqrt{\frac{2N}{K}} & m = 0 \\
\rho \pm \frac{\pi}{2} \cdot \sqrt{\frac{N}{K}} & m \neq 0.
\end{cases}$$

(57)

For the Fourier transform of an autocorrelation ($\rho = 1$) replace $N$ with $2N$.

4.3 2-bit sampling

Figure 9 shows a block diagram of a typical 2-bit digitizer. The quantization counters count the number of 2-bit samples in each of the four quantization states over an integration period $T$. For example, in an integration time $T$, there are $K = f_0 T = K_{00} + K_{01} + K_{10} + K_{11}$ samples, where $K_{ij}$ are the number of counts for the quantization states $ij$ ($i = 0, 1$, $j = 0, 1$). For random Gaussian input noise, the optimal relative count of these states is set by the multiplication scheme, and if the measured quantization counts are incorrect, the threshold voltages can be adjusted until the correct count is obtained. This feedback loop can be used as an automatic gain compensation (AGC) circuit.

Figure 10 shows three possible multiplication schemes that can be used with a 2-bit digitizer. Figure 10(a) and (d) show the quantization states and weights and multiplication states and weights...
for a 4-level digitization scheme. The efficiency of this scheme, and the efficiency of a slightly modified version of it, is discussed in the next section. Figure 10(b) and (e) show that a 2-bit correlator can be used as a 1-bit correlator simply by changing the quantization and multiplication weights. This option would be exercised in cases where efficiency (relative SNR) is traded off for more lags (in an FPGA approach, the multiplication table for a 1-bit correlation requires fewer resources than a 2-bit multiplication table). Figure 10(c) and (f) show a 3-level digitization scheme. The 3-level scheme has an efficiency and a resource usage intermediate to that of 2-level and 4-level correlation (a few extra lags can be obtained for a small drop in efficiency). The 3-level correlator is also analyzed in the next section.

4.3.1 2-bit sampling with 3- and 4-level quantization

Figure 11 and Table 5 show the multiplication scheme for two inputs $x$ and $y$ both digitized to 2-bits. Figure 11 and Table 5 contain generic multiplication weights $(l, m, n)$, so that the efficiency of various 2-bit multiplication schemes can be evaluated. Figure 11 shows the weights in the multiplication table relative to the weights in Fig. 10. The joint probabilities in Table 5, $P_{ijkl}$, are obtained by integrating (30) over limits implied by the input voltage ranges shown in Table 5. In general, these joint probabilities are evaluated numerically (see p221 [9], and [2,3,5]). Table 5 contains the
probabilities for the three cases $\rho = 1$, $\rho = 0$, and $\rho \ll 1$. The parameters $\phi$ and $E$ are

$$\phi(v_0) = \operatorname{erf}\left(\frac{v_0}{\sqrt{2}}\right)$$

and

$$E(v_0) = \exp\left(-\frac{1}{2}v_0^2\right)$$

where the threshold voltages on the digitizer are set according to $V_{\text{zero}} = 0$ and $V_{\text{high}} = -V_{\text{low}} = v_0\sigma$ and $\sigma$ is the RMS voltage of the input.

The mean and variance of the generically weighted 2-bit cross correlation is

$$\mu r_g = E\{r_g\} = \sum_{ijkl} w_{ijkl} P_{ijkl}$$

and

$$\sigma^2 r_g = \text{Var}\{r_g[n]\} = \sum_{ijkl} (w_{ijkl} - \mu r_g)^2 P_{ijkl} = \sum_{ijkl} w^2_{ijkl} P_{ijkl} - \mu^2 r_g.$$  

Using Table 5 we get the following summation results:

$$\sum_{ijkl} w_{ijkl} P_{ijkl} = 2l(P_{1111} - P_{1101}) + 4n(P_{1110} - P_{1100}) + 2m(P_{1010} - P_{1000})$$

$$= \begin{cases} m\phi + l(1 - \phi) & \rho = 1 \\ 2\rho \pi \cdot [E^2(l - 2n + m) + 2E(n - m) + m] & \rho \ll 1 \end{cases}$$

and

$$\sum_{ijkl} w^2_{ijkl} P_{ijkl} = 2l^2(P_{1111} + P_{1101}) + 4n^2(P_{1110} + P_{1100}) + 2m^2(P_{1010} + P_{1000})$$

$$= \begin{cases} m^2\phi + l^2(1 - \phi) & \rho = 1 \\ \phi^2(l^2 - 2n^2 + m^2) + 2\phi(n^2 - l^2) + l^2 & \rho \ll 1 \end{cases}$$
Table 5: 2-bit multiplication scheme.

<table>
<thead>
<tr>
<th>Input voltage range</th>
<th>State, ijl</th>
<th>Weight, wijkl</th>
<th>Probabilities, Pijkl</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>l</td>
<td>$\rho = 1$</td>
</tr>
<tr>
<td>$x &lt; V_{\text{low}}$, $y &lt; V_{\text{low}}$</td>
<td>1111</td>
<td>$\frac{1}{2}(1-\phi) - \frac{1}{4}(1-\phi)^2 + \frac{\rho E^2}{2\pi}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>n</td>
<td>$\rho = 0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$-n$</td>
<td>$\frac{\phi^2}{4} + \frac{\phi^2(1-\phi)}{4} - \frac{\rho E^2(1-E)^2}{2\pi}$</td>
</tr>
<tr>
<td>$x &lt; V_{\text{low}}$, $V_{\text{low}} \leq y &lt; V_{\text{zero}}$</td>
<td>1110</td>
<td>$\phi$</td>
<td>$\rho \ll 1$</td>
</tr>
<tr>
<td>$x &lt; V_{\text{low}}$, $V_{\text{zero}} \leq y &lt; V_{\text{high}}$</td>
<td>1100</td>
<td>$0$</td>
<td>$\frac{\phi^2}{4} + \frac{\phi^2(1-\phi)}{4} - \frac{\rho E^2(1-E)^2}{2\pi}$</td>
</tr>
<tr>
<td>$x &lt; V_{\text{low}}$, $V_{\text{high}} \leq y$</td>
<td>1101</td>
<td>$0$</td>
<td>$\frac{\phi^2}{4} + \frac{\phi^2(1-\phi)}{4} - \frac{\rho E^2(1-E)^2}{2\pi}$</td>
</tr>
<tr>
<td>$V_{\text{low}} \leq x &lt; V_{\text{zero}}$, $y &lt; V_{\text{low}}$</td>
<td>1011</td>
<td>$\frac{\phi}{2}$</td>
<td>$P_{1110}$</td>
</tr>
<tr>
<td>$V_{\text{low}} \leq x &lt; V_{\text{zero}}$, $V_{\text{low}} \leq y &lt; V_{\text{zero}}$</td>
<td>1010</td>
<td>$\frac{\phi^2}{4}$</td>
<td>$P_{1100}$</td>
</tr>
<tr>
<td>$V_{\text{low}} \leq x &lt; V_{\text{zero}}$, $V_{\text{zero}} \leq y &lt; V_{\text{high}}$</td>
<td>1000</td>
<td>$\phi^2$</td>
<td>$P_{1010}$</td>
</tr>
<tr>
<td>$V_{\text{low}} \leq x &lt; V_{\text{zero}}$, $V_{\text{high}} \leq y$</td>
<td>1001</td>
<td>$-n$</td>
<td>$P_{1110}$</td>
</tr>
<tr>
<td>$V_{\text{zero}} \leq x &lt; V_{\text{high}}$, $y &lt; V_{\text{low}}$</td>
<td>0011</td>
<td>$-n$</td>
<td>$P_{1100}$</td>
</tr>
<tr>
<td>$V_{\text{zero}} \leq x &lt; V_{\text{high}}$, $V_{\text{low}} \leq y &lt; V_{\text{zero}}$</td>
<td>0010</td>
<td>$-m$</td>
<td>$P_{1100}$</td>
</tr>
<tr>
<td>$V_{\text{zero}} \leq x &lt; V_{\text{high}}$, $V_{\text{zero}} \leq y &lt; V_{\text{high}}$</td>
<td>0000</td>
<td>$m$</td>
<td>$P_{1010}$</td>
</tr>
<tr>
<td>$V_{\text{zero}} \leq x &lt; V_{\text{high}}$, $V_{\text{high}} \leq y$</td>
<td>0001</td>
<td>$n$</td>
<td>$P_{1110}$</td>
</tr>
<tr>
<td>$V_{\text{high}} \leq x$, $y &lt; V_{\text{low}}$</td>
<td>0111</td>
<td>$-l$</td>
<td>$P_{1100}$</td>
</tr>
<tr>
<td>$V_{\text{high}} \leq x$, $V_{\text{low}} \leq y &lt; V_{\text{zero}}$</td>
<td>0110</td>
<td>$-n$</td>
<td>$P_{1100}$</td>
</tr>
<tr>
<td>$V_{\text{high}} \leq x$, $V_{\text{zero}} \leq y &lt; V_{\text{high}}$</td>
<td>0100</td>
<td>$n$</td>
<td>$P_{1110}$</td>
</tr>
<tr>
<td>$V_{\text{high}} \leq x$, $V_{\text{high}} \leq y$</td>
<td>0101</td>
<td>$l$</td>
<td>$P_{1110}$</td>
</tr>
</tbody>
</table>
So the variances of the correlation are

\[
\sigma_{r_g}^2 = \begin{cases} 
\frac{(l-m)^2\phi(1-\phi)}{\sqrt{\phi^2(l^2-2n^2+m^2) + 2\phi(n^2-l^2) + l^2}} & \rho = 1 \\
\rho \ll 1
\end{cases}
\]

and

\[
r_g = \begin{cases} 
\frac{2\rho}{\pi} \cdot [E^2(l-2n+m) + 2E(n-m) + m] \pm \sqrt{\phi^2(l^2-2n^2+m^2) + 2\phi(n^2-l^2) + l^2} & \rho = 1 \\
\rho \ll 1
\end{cases}
\]

Normalizing both correlation estimates by the estimate for \(\rho = 1\) gives the relationship between the measured correlation and the correlation coefficients; that is,

\[
r_{g}^* = \begin{cases} 
\frac{1}{\rho} \cdot \frac{2\pi}{E^2(l-2n+m) + 2E(n-m) + m} \pm \frac{\sqrt{\phi^2(l^2-2n^2+m^2) + 2\phi(n^2-l^2) + l^2}}{\sqrt{\phi^2(l^2-2n^2+m^2) + 2\phi(n^2-l^2) + l^2}} & \rho = 1 \\
\rho \ll 1
\end{cases}
\]

This equation gives us the relationship between the measured correlation \(r_{g}^*\) and the actual correlation \(\rho\). The amplitude corrected \(K\)-ensemble 2-bit correlation estimate for \(\rho \ll 1\) is then

\[
\rho_g = r_{g}^* \cdot \frac{\pi}{2} \cdot \frac{m\phi + l(1-\phi)}{E^2(l-2n+m) + 2E(n-m) + m} \cdot \frac{1}{\sqrt{K}}
\]

\[
= \rho \pm \frac{\pi}{2} \cdot \frac{\sqrt{\phi^2(l^2-2n^2+m^2) + 2\phi(n^2-l^2) + l^2}}{E^2(l-2n+m) + 2E(n-m) + m} \cdot \frac{1}{\sqrt{K}}
\]

The SNR of the generically weighted 2-bit correlator relative to the full precision correlator SNR in (39) (for \(\rho \ll 1\)) is then

\[
\frac{\text{SNR}_g}{\text{SNR}} = \frac{2}{\pi} \cdot \frac{E^2(l-2n+m) + 2E(n-m) + m}{\sqrt{\phi^2(l^2-2n^2+m^2) + 2\phi(n^2-l^2) + l^2}}
\]

The relationship between the measured correlation coefficient and the actual correlation coefficient is non-linear for large values of \(\rho\). Large values of \(\rho\) occur for the lags close to lag zero in an autocorrelation estimate. The amplitude correction function for these large correlation coefficients can be found in [3,5,7,9,11].

Table 6 shows the generic weights, relative SNR, and optimal threshold voltages for the different 2-bit multiplication schemes. When implementing any of these multiplication schemes, the weights in the multiplication tables are translated and scaled so that only positive values exist. Positive weights mean that only an adder is required to determine the count for a particular lag. Table 6 shows the multiplication weight transforms required to obtain only positive weights, along with the number of bits required in the adder attached to the multiplier output. The decreasing relative SNR with reduced number of bits in the adder clearly shows the trade-off between SNR and resource usage (more bits in the adder uses more logic cells in the FPGA). See Section 5 for further implementation details.

Figure 12 shows the relative SNR of the 4-level multiplication scheme versus threshold voltage, and the relative SNR of the 4-level, with deleted inner products, multiplication scheme versus threshold voltage. The full 2-bit scheme has maximum of 0.881, so the variance in (67) equals 1.135/\(\sqrt{K}\). To obtain the same SNR as the full precision case, the integration time (and hence \(K\)) of the 2-bit correlator needs to be increased by 1.135^2 = 1.288. The deleted inner product scheme has maximum
Table 6: Characteristics of the 2-bit digitizer multiplication schemes.

<table>
<thead>
<tr>
<th>Description</th>
<th>Generic weights, ((l, m, n))</th>
<th>Threshold, (v_0)</th>
<th>relative SNR</th>
<th>Multiplication weight transform</th>
<th>Adder size</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-level</td>
<td>(9,1,3)</td>
<td>0.996</td>
<td>0.881</td>
<td>((w_{ijkl} + 9)/2)</td>
<td>4-bits</td>
</tr>
<tr>
<td>4-level, with deleted inner products</td>
<td>(9,0,3)</td>
<td>0.906</td>
<td>0.872</td>
<td>(w_{ijkl}/3 + 3)</td>
<td>3-bits</td>
</tr>
<tr>
<td>3-level</td>
<td>(1,0,0)</td>
<td>0.612</td>
<td>0.810</td>
<td>(w_{ijkl} + 1)</td>
<td>2-bits</td>
</tr>
<tr>
<td>2-level</td>
<td>(1,1,1)</td>
<td>-</td>
<td>0.637</td>
<td>((w_{ijkl} + 1)/2)</td>
<td>1-bit</td>
</tr>
</tbody>
</table>

Figure 12: Relative signal-to-noise ratio (SNR) of the 2-bit correlation schemes to the full precision correlation SNR (solid line = full 2-bit scheme, dashed line = 2-bit scheme with deleted inner products).

The current OVRO correlator uses a 2-bit multiplication scheme with deleted inner products. The rest of this section details the expected output of such a correlator system. Figure 13 shows the quantization probabilities and weights along with the multiplication weights for the biased 2-bit multiplication scheme with deleted inner-products. The quantization probabilities in this figure are determined with optimal threshold voltages, i.e., \(V_{\text{high}} = -V_{\text{low}} = 0.906\sigma\) and \(V_{\text{zero}} = 0\), where \(\sigma\) is the RMS voltage of the input noise voltage. These optimal threshold voltages can be set and monitored using the quantization counters shown in Fig. 9. In an integration period \(T\) with a total of \(K = f_D T\) samples, the expected counts are as follows: 

\[ K_{00} = K_{10} = \text{erf}(0.906/\sqrt{2})/2 \cdot K = 0.318K \]

and 

\[ K_{01} = K_{11} = [1 - \text{erf}(0.906/\sqrt{2})]/2 \cdot K = 0.182K. \]

The expected count out of a particular lag of a correlator for the accumulation of \(K\) correlation estimates is determined using (65) evaluated for \((l, m, n) = (9, 0, 3)\), \(v_0 = 0.906\), divided by 3, and...
offset by 3. The expected 4-level, deleted inner product, count is then,

\[ c_{4d} = \begin{cases} 
4.095 \cdot K \pm 1.444 \cdot \sqrt{K} & \rho = 1 \\
(1.125 \cdot \rho + 3) \cdot K \pm 1.289 \cdot \sqrt{K} & \rho \ll 1 
\end{cases} \quad (69) \]

After these counts are read from the correlator, they are converted back to the \( K \)-ensemble average version of (65) via

\[ r_{4d} = 3 \left( \frac{c_{2d}}{K} - 3 \right) \]

\[ \begin{cases} 
3.284 \pm 4.333 \cdot \frac{1}{\sqrt{K}} & \rho = 1 \\
3.374 \cdot \rho \pm 3.868 \cdot \frac{1}{\sqrt{K}} & \rho \ll 1 
\end{cases} \quad (70) \]

Normalizing this by the expected value for \( \rho = 1 \) gives

\[ r'_{4d} = \begin{cases} 
1 \pm 1.319 \cdot \frac{1}{\sqrt{K}} & \rho = 1 \\
1.027 \cdot \rho \pm 1.178 \cdot \frac{1}{\sqrt{K}} & \rho \ll 1 
\end{cases} \quad (71) \]

Finally, correction of the amplitudes for small correlations gives

\[ \rho_{4d} = \begin{cases} 
1 \pm 1.319 \cdot \frac{1}{\sqrt{K}} & \rho = 1 \\
\rho \pm 1.146 \cdot \frac{1}{\sqrt{K}} & \rho \ll 1 
\end{cases} \quad (72) \]

For an \( N \) lag correlator, the expected value of the correlation of an input signal of bandwidth \( B \), digitized to 2-bits at sample rate \( f_D = 2B \) and correlatated using a 4-level multiplication scheme with deleted inner products is then

\[ \rho_{4d}[n] = \begin{cases} 
1.000 \pm 1.319 \cdot \frac{1}{\sqrt{K}} & n = 0, \rho = 1 \\
\rho \pm 1.146 \cdot \frac{1}{\sqrt{K}} & n = 0, \rho \ll 1 \\
0.000 \pm 1.146 \cdot \frac{1}{\sqrt{K}} & n \neq 0 
\end{cases} \quad (73) \]
where the correlation counts read back from the correlator have been converted to correlation coefficients using the procedure just described.

The Fourier transform of the 2-bit cross correlation coefficients in (73) for \(N\) lags can be calculated in a similar manner to the Fourier transform of the full precision correlation coefficients calculated in (40) and (45); that is,

\[
P_{4d}[m] \approx \begin{cases} 
\rho \pm 1.146 \cdot \sqrt{\frac{2N}{K}} & m = 0 \\
\rho \pm 1.146 \cdot \sqrt{\frac{N}{K}} & m \neq 0.
\end{cases}
\]  

(74)

For the Fourier transform of an autocorrelation (\(\rho = 1\)) replace \(N\) with \(2N\).

### 4.4 Allen deviation

Given two correlation coefficient spectra (full precision, 1-bit, or 2-bit), \(P_A[m]\) and \(P_B[m]\) with means \(\mu_A[m] = \mu_B[m] = \mu[m]\), and standard deviations \(\sigma_A[m] = \sigma_B[m] = \sigma[m]\), the Allen deviation is determined from the random process

\[
P[m] = \frac{P_A[m] - P_B[m]}{(P_A[m] - P_B[m])/2}.
\]  

(75)

The mean and standard deviation of each sample in this process is derived as follows: for sample \(m\), the numerator \(P_A[m] - P_B[m]\) has zero-mean and standard deviation \(\sqrt{2} \sigma[m]\), while the denominator \((P_A[m] - P_B[m])/2\) has mean \(\mu[m]\) and standard deviation \(\sigma[m]/\sqrt{2}\). Since it is assumed that the standard deviation is significantly smaller than the mean, the division in (75) is dominated by the mean \(\mu[m]\), giving

\[
P[m] = 0 \pm \sqrt{2} \cdot \frac{\sigma[m]}{\mu[m]}.
\]  

(76)

The standard deviation of this difference process is the Allen deviation (the square of the Allen deviation is the Allen variance); that is,

\[
A[m] = \sqrt{2} \cdot \frac{\sigma[m]}{\mu[m]}.
\]  

(77)

From (45), (57), and (74), the Allen variance for a cross correlation function is

\[
A[m] \approx \begin{cases} 
\alpha \cdot \sqrt{\frac{4N}{K}} & m = 0 \\
\alpha \cdot \sqrt{\frac{2N}{K}} & m \neq 0.
\end{cases}
\]  

(78)

where the constant \(\alpha = 1\) for a full precision correlator, \(\alpha = \pi/2\) for a 1-bit correlator, and \(\alpha = 1.146\) for a 2-bit correlator with deleted inner products. For an autocorrelation, replace \(N\) with \(2N\).

When performing Allen deviation measurements on the correlation spectra of data filtered to bandwidth \(B\), sampled at \(f_D > 2B\), the Allen deviation should only be determined over the passband of the filter. Allen deviation measurements taken outside the bandpass region violate the assumption that the mean of the processes \(A\) and \(B\) is greater than the standard deviation.

An advantage of using the Allen deviation to determine that a correlator is “integrating down the noise” (according to (78)) is that the bandpass shape of the filter used with the noise source is normalized out and hence the variance of a particular spectral bin is not scaled by the bandpass level of the filter.
4.5 Calculating the size of the accumulator in a correlator chip

The expected count in any particular lag of a correlator employing a 2-bit multiplication scheme with deleted inner products is given by (69). The largest accumulated result is for $\rho = 1$ and is $4.095K$ (the smallest accumulated result is for $\rho = 0$ and is $3K$—not much smaller). This requires an accumulator with at least $Q$-bits, where $Q$ is the next integer above $\log_2(4.095K)$. For example, at $f_D = 1$GHz, an integration time of 1ms requires a $Q = 22$-bit accumulator, while an integration time of 6.25ms requires a $Q = 25$-bit accumulator. The number of bits required to represent the variance of (69) indicates the number of noise bits. For the examples just mentioned, the noise is 11-bits for a 1ms integration, and is 12-bits for a 6.25ms integration. Since these bits are noise, there is no need to read them all when downloading the correlation counts. This observation is usually summarized as: half the bits of a $Q$-bit correlation are not significant and can therefore be discarded.

In a correlator implementation, it is common to discard some of the noise bits, so that the count read back by the control system contains mainly significant bits. The multiplier-adder (MA) of a 2-bit correlator with deleted inner products requires 3-bits, and the carry output of this MA can be counted by an accumulator. For an integration time $T$, the 3 least significant bits (LSBs) of the $Q$ bit correlation count are in the adder so they are not conveniently available, and $Q/2 - 3$ of the bits that are available are noise. The $Q/2$ MSBs of the available count bits plus a few of the noise bits are usually read back by the control system (call this number of bits $C$), and the remaining LSBs of the count, i.e., $P = Q - C - 3$, are referred to as prescaler bits. For example, the OVRO correlators use a 3-bit adder, 3-bit prescaler, and 18-bit accumulator and operates at 256MHz. Hence, $Q = 24$-bits and the largest integration time possible is $T \leq 2^{24}/(4.095 \times 256 \times 10^8) = 16$ms for $P = 1$.

For a time demultiplexed correlator implemented in an FPGA, the carry outputs of the MAs calculating the same lags can be chained together through full-adders (FAs). If the number of bits lost in the FAs does not exceed the number of noise bits, then those bits are not missed, and they can be considered part of the prescaler. For example, in a time demultiplex-by-16 scheme, multiple sets of 16 MAs calculate the same lag result from different input samples. The 16 carry outputs of these MAs can be cascaded through 4 layers of FAs (with 8, 4, 2, 1 FAs in each layer) to produce a single carry output which can drive the accumulator for that lag. The carry output of the carry-cascade goes high for every 16 carries in, i.e., the carry-cascade carry output is like the carry output of a 4-bit counter, hence, the carry-cascade chain is considered to act as a 4-bit prescaler. Since there are 3-bits lost in the MAs and a further 4-bits lost in the FA chain, the minimum integration time is set by the fact that we want at least twice the number of bits lost in the adder and prescaler in our correlation count, i.e., we want at least 14-bits which gives $T \geq 2^{14}/(3f_D) = 5.5\mu s$ at 1GHz (for $\rho = 0$). This is a fairly short timescale relative to typical integration times, so more bits can be added to the prescaler. For example, the 1ms integration at 1GHz has $Q = 22$-bits of which 11 LSBs are noise, 7 of these bits are lost in the adder and prescaler, so there are 15MSBs available. Of these 15-bits, the 4LSBs are still noise, so 3-bits could be prescaled. If the integration time was increased to 6.25ms, then $Q = 25$ bits, there are 12-bits of noise, and 18 MSBs are available (the 5 LSBs of this 18-bit number are noise). If an 18-bit counter is used for the lag accumulator and the 16 MSBs of this counter are buffered for read back by the control system at the end of an integration ($C = 16$, $P = 6$), then the system can be used for both 1ms and 6.25ms integrations without corrupting the signal statistics (the integration time can be varied over $88\mu s < T \leq 8.2$ms).

The limits on the integration time are determined by

$$\frac{2^{2(3+P)}}{3f_D} < T \leq \frac{2^Q}{4.095f_D}$$

(79)

where the total number of bits $Q = 3 + P + C$, $P$ is the prescaler length, $C$ is the length of the counter read back by the control system (i.e., it is the $C$ MSBs of the $Q$-bit number). The minimum integration time is given by the amount of time it takes for the read back counter to accumulate a statistically significant number of samples for a $\rho = 0$ lag. For example, in the
current OVRO correlator, $P=3$ so the minimum integration time at a 256MHz clock frequency is $2^{12}/256\text{MHz}=5.4 \mu s$.

When dealing with time demultiplexed correlators, it is important to note that (79) depends on the digitizer clock frequency and not the correlator clock frequency. This is because the prescaler, which includes the carry-cascade effective bit length, accounts for the lower correlator clock frequency.
5 FPGA implementation of a wide bandwidth correlator

This section details the development of VHDL code that implements a correlator that can operate using any of the 2-bit multiplication schemes presented in the previous section. The 2-bit digitizers to be used with this cross correlator system will operate at 1GHz and will time demultiplex the 2-bit data by 16 to give a 32-bit output at 62.5MHz. The input bandwidth to the digitizer can be a maximum of 500MHz. Input bandwidths of 250MHz, 125MHz, 62.5MHz, and 31.25MHz, can be processed at an effective Nyquist rate by downloading new VHDL code that has a reduced demultiplexing factor. Alternatively, the reduced bandwidth signals can be oversampled to obtain an increase in SNR (the relative SNR for oversampled systems can be found in [2,5,9]). The FPGAs implementing the correlator will operate at 62.5MHz. Two correlator chips will process 1 baseline and each correlator card will contain 5 baselines (10 FPGAs per correlator card). Three cards are required to process each 500MHz band from the 15 baselines available from OVROs 6 element interferometer. With 4GHz of processable bandwidth, 24 correlator cards are necessary.

5.1 *** to be completed ***

DWH: 15-June-1998. This section is still being worked on. Updates of this document can be obtained from me via email dwh@ovro.caltech.edu or via ftp://ovro.caltech.edu/pub/dwh/correlators.ps
References


