Intermediate-level data acquisition interfaces

A. Harris, 12 May 2006

1 Overview

Each correlator chassis contains a microcontroller that handles all real-time interfaces with the correlator hardware. This note describes the intermediate-level software tasks that comprise the routine data acquisition interface between a host and the correlator micro.

The two basic tasks are:

1. A task that measures the cross-correlation coefficient for each lag, storing the data in a single buffer with length equal to the number of lags. Depending on the input signal, this task is appropriate for measuring correlator offsets, the set of monochromatic signals needed to establish the correlator's transfer function, and the difference spectrum of calibration blackbody loads. This task is a total power difference measurement.
2. A task that measures the cross-correlation coefficient for each lag with and without the amplitude calibration noise diode noise power. The data are stored in two buffers with lengths equal to the number of lags, one buffer for data with the noise diode on, the other for data with the noise diode off. This task is meant for astronomical observations.

Important auxiliary tasks are:

3. Returning data from the micros.
4. Setting the correlator input power levels.
5. Setting system synchronization parameters, such as which micro is the timing master.
6. Returning data from system monitor points.

2 Tasks in detail

2.1 Total power difference

The correlation receiver architecture produces the power difference between the two receiver inputs. This task looks at that difference at its simplest level. There are three tasks that use this difference in distinct ways:

2.1.1 zero

Correlator offsets, while small, are needed to make the most accurate total power measurements. This task establishes correlator offsets by measuring the cross-correlation with no microwave input power. The power can be reduced to suitable levels with the attenuators and switches in the microwave amplifier module; these are under the micro's control. The switches are just before the last gain stage, leaving a residual amount of uncorrelated microwave power from the amplifier noise. *zero* returns lag data, one datum per lag. All Zpectrometer measurements are differential, and *zero* provides data for any total power measurement.

2.1.2 cwsigs

A matrix converts data from the lag to the spectral domain, an analog to a discrete Fourier transform. It is necessary to establish the matrix elements by measurement for the Zpectrometer; this accounts for variations in delays to individual lags as a function of frequency. The *cwsigs* task records the cross-correlator response to a family of continuous wave (monochromatic) signals. The measurement pattern is to set the frequency of a microwave frequency synthesizer, allow some settling time, read out and record correlator lag data, then step to the next frequency. The loop breaks periodically to establish correlator zeros.

This task produces a relatively large amount of data: one full lag readout per frequency, plus zero measurements. Sampling the correlator transfer function requires frequency steps a few times the channel width. We generally make this measurement occasionally and store these data in a file separate from astronomical data. Converting from the lag to spectral domains requires the inverse of the transfer function matrix accumulated by *cwsigs*; this is the purpose of *cwsigsinvert*.

2.1.3 totpwr

This task measures the total power difference spectrum between the receiver's two inputs. At the low level, it integrates cross-correlation lag data and returns a single buffer with length equal to the number of lags.

Subsequent processing subtracts correlator offsets using data from the appropriate *zero* measurement and multiplies this true lag data by the inverse of the transfer function matrix to produce a spectrum.

An example of *totpwr*'s use is finding the passband gain calibration file by measuring two blackbody loads with a known temperature difference.

2.2 Differential measurements with switched noise diode power

This is the standard observing mode: integration with the source in one of the receiver inputs and with the noise diode switching under the micro's control. Each integration produces two buffers, one of the power in Beam 1 minus that in Beam 2 (the usual correlation receiver output) with the noise diode on and one with the power in Beam 1 minus that in Beam 2 with the noise diode off. Only one noise diode should fire at a time, injecting an amplitude calibration signal into one of the receiver inputs. These data make sense only as one half of a nod pair, so no zero measurement or correction is needed; correlator offsets cancel in the nod side differencing. Since software drives the nod cycle it is simplest to initiate a new observation for each half of the nod cycle, for a total of four buffers in two files.

The buffers are:

Buffer	**Signal**	**Nod side and file #**
B0	Source in Beam 1 + sky1_0 – sky2_0 + offset_0	1
B1	Source in Beam 1 + sky1_1 – sky2_1 + offset_1 + noise diode_1	1
B2	Source in Beam 2 + sky1_2 – sky2_2 + offset_2	2
B3	Source in Beam 2 + sky1_3 – sky2_3 + offset_3 + noise diode_2	2

Here we assume that there is only one source, with the receiver's internal phase switching phase set so power in Beam 1 has a positive correlator output and a source in Beam 2 has a negative output. The subscripts represent the values for fluctuating terms averaged across the integration time.

The useful sums and differences are:
source spectrum from (B0 + B1 – B2 – B3)/4
noise diode spectrum from (B1 – B0 + B3 – B2)/2 .
This is exact in the case that the differential sky, offset, and noise diode powers are constant over the nod cycle.

Nonideal terms should be small but can be included explicitly in the expressions above for full modeling. If the optics are well balanced and the beams have good overlap through the atmosphere the receiver architecture reduces the differential sky term, sky1 – sky2, to a low level. Phase switching the signal in one "arm" of the receiver reduces other errors. This phase switching is rather confusingly called beamswitching at the GBT. It is not beamswitching: the receiver continuously differences the power at the two inputs under all circumstances, and the source remains in one beam, whatever the phase, until the telescope nods. Phase switching changes the polarity of the source signal at the correlator output, with two main purposes. First, it shifts the power difference frequency away from zero to the switching frequency, preferably into a regime where the amplifier 1/*f* gain fluctuations are small. Second, synchronously detecting the modulated signal discriminates against offsets at the correlator output, leaving only a residual differential offset and pickup from signals synchronous with the phase inversion signal. The Zpectrometer's phase switch will run at a frequency somewhere between 3 kHz and 10 kHz to get past semiconductor 1/*f* noise knees. For reference, the Spectrometer's phase switch frequency is about 100 Hz.

Modulating the noise diode power improves the accuracy of the noise diode spectrum by reducing errors from fluctuating differential sky and offset power contributions within each nod side observation. This modulation does not improve the astronomical spectrum, which relies on differential sky, offset, and multiplier gain stability across an entire nod cycle. With a correlator readout rate of 86.8 Hz (11.52 ms) the noise diode can switch at a maximum frequency of 43.4 Hz (23.04 ms). Switching more slowly is also practical since, as for the astronomical data, receiver architecture and phase switching should reduce the error terms to low small levels. The Zpectrometer's internal micro will manage the noise diode switching since it is responsible for synchronous data acquisition.

As a note, this data format is equivalent to the WASP2 *nodobs* format, although the Zpectrometer requires a pair of files for four buffers.

2.3 Returning data from the micros

The micro is a TCP/IP data server for the accumulated data. Data are buffered to allow readout any time between the end of integration and before the following integration finishes. This time will be minutes under most circumstances.

2.4 Setting correlator power levels

The micro can set the step attenuators in the internal microwave gain modules either to an absolute value or by running an auto-adjust routine that sets the power to a given level as given by the zero lag. This is necessary when the input power changes by more than a factor of two or so, which can occur during amplitude calibration cycles.

2.5 Setting synchronization parameters

One correlator chassis is the master for all timing, including the phase switch signal for the receiver and, consequently, the correlator ADC readout timing. The other correlators are slaves that must be synchronous with the data acquisition from the master. The master sends out a high-frequency master clock (2 MHz) and a sync pulse in local hardware for this purpose. A software task is to set one and only one chassis as a master and set the others as slave unit. All chassis are in the slave mode on power-up.

2.6 Returning data from monitor points

The micro is a TCP/IP data server for the latest monitor point data. Monitor point data could include

- Component and chassis temperatures
- Step attenuator values
- Supply voltages
- Supply currents