JIVE Uniboard Correlator Review Memo: The Delay Model

Des Small

February 1, 2013

1 Introduction

This document outlines the calculation and use of the current proposed delay and phase-delay models supplied to the Uniboard by the JIVE Uniboard Control System (JUCS).

2 Summary of recommendations

According to the correlator design document (p.13), the correlator will operate with 16 MHz subbands for the current personality. This implies a Nyqvist sample rate of 32 MHz. This means that the sample time t_s is $\frac{1}{32\times10^6}$. The sample time, or *tick*, is the unit used for delay calculations.

The current recommendations for delay calculations are:

- Linear polynomials over an integration (an interval of up to one second);
- Units of ticks, i.e., $\frac{1}{32} \times 10^{-6}$ s;
- 48-bit fixed-point coefficients;
- 8 bits after binary point for delay;
- 32 bits after binary point for delay-rate;
- May be necessary to use quadratic polynomials for RadioAstron correlations

The recommendations for phase calculations are:

- Quadratic polynomials over an integration (an interval of up to one second);
- 64-bit coefficients in units of cycles;
- All 64 bits after a notional binary point;
- Units of cycles; i.e., 1 cycle corresponds to 2π radians.

2.1 A Note on the current implementation

The current implementation in the Uniboard firmware does not match what is described in this section. It was decided to leave changes to the model until other parts of the Uniboard implementation of correlation had been thoroughly debugged.

The current – transitional – implementation is described in Section 5 below.

2.2 From SCHED to Erlang

Delay model calculations are done using the SCHED program. Delays are calculated at a one-second interval. Parameters (scan times, station and source positions) are taken from the experiment database, which is filled from VEX files.

The version of SCHED used in the JUC CCS uses a slightly different output format from that used by SFXC, but that is the only change, and it only affects the top-level C wrapper, not the FORTRAN core of the program.

The calculation of the SCHED delays is intended to be done offline as part of the preparation for correlation; the results are stored in a directory structure for the JUC CCS that is modelled on that used in the existing Mk IV CCS.

When a correlation is started, an Erlang model-reader component reads the CALC output file and computes the coefficients of interpolating polynomials for delay and phase for a one second interval. The delay polynomial is linear, so it needs two delay points; the phase polynomials are quadratic, so they use three consecutive delay points, and a separate polynomial is calculated for each subband.

In the case that the integration time is less than a second, the polynomials sent to the Uniboard are derived from the one-second polynomials by selecting the appropriate "parent" polynomial and adjusting the coefficients such that they represent the same polynomial over the integration time, effectively shifting the origin. In the rationales given in Section 3 we consider only the most demanding case for accuracy, namely the case where the integration time is a full second.

2.3 Subband frequencies

Frequencies are ultimately taken from the VEX file from the experiment (via the experiment database). The `$FREQ` block stores the sky frequency which maps to 0 Hz in the BBC output for upper and lower subbands and the bandwidth of the channel. For the upper sideband, this is the frequency used for phase; for the lower sideband we subtract the bandwidth to calculate the frequency. In other words, we use the left-hand edge of the channel as the reference frequency for phase calculations.

For each subband, the delays computed by CALC are multiplied by the subband's reference frequency (using double-precision floating-point arithmetic) to calculate phase delays (for each second), which are used to calculate the coefficients of an interpolating polynomial.

2.4 Discretization and packetization

All polynomials are initially calculated using floating point arithmetic and then rescaled to the appropriate units before discretization: ticks, in the case of delay. The phase polynomial coefficients are simply reduced to their fractional part, since they are calculated in cycles.

The only subtlety in discretization is rescaling with binary points. The delay rate coefficient is to be stored in a 48-bit register with 32 bits after the (notional) binary point. This simply means that we multiply the floating point value by 2^{32} and convert the result to an integer. The delay coefficient has eight bits after the binary point so it is multiplied by 2^8 before conversion to an integer. Note that the position of the fixed point is not stored anywhere in the data: it is simply a convention that the Uniboard and the control software share.

Since the coefficients for phase polynomials only store fractional values in 64-bit registers, we multiple the values by 2^{64} and convert them to integers. The conversion to integers is done by Erlang's built in rounding function which converts to the nearest integer, as opposed to truncating.

The delay coefficients for each station are collected into a packet and written to the correlator; likewise, for each station and subband a packet of phase delay coefficients is written to the correlator. The format of the packets is given in the correlator definition document; it is transparent from the Erlang code and is therefore not reproduced here.

2.5 On the Uniboard

The quantum of data processed by the Uniboard is the FFT segment, made up of 2048 samples, for a given station and subband. For each FFT segment the Uniboard calculates an integer delay for the whole segment, using the delay model coefficients provided. The calculation uses the time of the central point of an FFT block to calculate the delay. The delay is calculated using fixed point arithmetic, and the integer and fractional parts are used separately by the uniboard.

The integer delay is used to retrieve the data from an appropriate location in the DDR memory, effectively shifting it in time in the appropriate way.

As the correlator design document states (p.20), "the four most significant bits of the fractional part represent the fine delay to an accuracy of 1/16th of a subband sample. This is used as the input to a look up table containing phase corrections to be applied to each frequency bin output from the poly-phase filter bank".

The choice of eight bits after the binary point for the constant term in the delay polynomial was made in order to make possible an accuracy of 2^{-8} ticks $= 1.2 \times 10^{-10}$ s in delay calculations, which is safely more than is actually needed.

The nine most significant bits of the phase delay for each subband are input to the appropriate subband mixer at the input to the polyphase filter bank. For 9-bit accuracy to be maintained using the same phase delay polynomial coefficients over a one-second interval we show below that the coefficients must be stored to at least 59 bit accuracy, which it is convenient to round up to 64 bits.

3 Justifications

3.1 Fixed point

Fixed point arithmetic is much cheaper to implent in VHDL than floating point arithmetic. If we *had* the luxury of floating point arithmetic on the Uniboard there would be no need for any decisions about scaling, and we would not need the control system to multiply delays by frequencies to calculate phases: that could all be done on the Uniboard itself.

3.2 A priori estimates of delay coefficients

We are going to use quadratic polynomials to approximate station delays relative to the centre of the earth. In order to get bounds for the worst case for coefficients of the polynomials we consider a Taylor's series expansion of the diurnal delay due to rotation of the earth. Again in the interest of studying the worst case we assume a station on the equator.

The delay is approximated by the polynomial

$$\text{delay}(t + \Delta t) = c_0 + c_1 \Delta t + c_2 \Delta t^2, \tag{1}$$

with coefficients c_i given by

$$c_0 = \text{delay} \quad = \frac{R}{c} \cos \Omega t \tag{2a}$$

$$c_1 = \frac{d}{dt} \text{delay} \; = -\frac{R}{c} \Omega \sin \Omega t \tag{2b}$$

$$c_2 = \frac{1}{2} \frac{d^2}{dt^2} \text{delay} = -\frac{1}{2} \frac{R}{c} \Omega^2 \cos \Omega t \tag{2c}$$

Plugging in the numbers ($\Omega = 2\pi/(24 \times 60 \times 60) = 7.3 \times 10^{-5}$ radians/s, $R = 6.4 \times 10^6$ m) we get maximum values for the coefficients (in second-based units) of

$$\max(c_0) \approx 0.02 \text{ s} \tag{3a}$$

$$\max(c_1) \approx 1.6 \times 10^{-6} \text{ s/s} \tag{3b}$$

$$\max(c_2) \approx 5.6 \times 10^{-11} \text{ s/s}^2. \tag{3c}$$

In order to rescale this to new variables Delay and T, with scaling $\text{Delay} = L\text{delay}$ and $T = Lt$. We get

$$\text{Delay} = L \, \text{delay}(t_0) + \frac{d}{dt} \text{delay}(t_0) \delta T + \frac{1}{2} \frac{1}{L} \frac{d^2}{dt^2} \text{delay}(t_0) \delta T^2 \tag{4}$$

In particular, scaling to the JUC's internal unit of ticks for time gives $L = 3.2 \times 10^7$ ticks/s, giving new coefficients C_i, with values

$$C_0 = \quad L \, \text{delay} \approx 6.4 \times 10^5 \text{ ticks} \tag{5a}$$

$$C_1 = \quad \frac{d}{dt} \text{delay} \approx 1.6 \times 10^{-6} \text{ ticks/tick} \tag{5b}$$

$$C_2 = \frac{1}{2} \frac{1}{L} \frac{d^2}{dt^2} \text{delay} \approx 1.8 \times 10^{-18} \text{ ticks/tick}^2 \tag{5c}$$

The quadratic coefficient C_2 is very small; it corresponds to $2^{-58.9}$ ticks/ticks^2, so that at least 59 places after the fixed binary point would be needed to store it. And the JUC uses only the four most significant bits of the fractional delay (in ticks) to calculate the fractional delay correction, so that the quadratic term becomes relevant at a time t_{quad} given by

$$1.8 \times 10^{-18} \text{ ticks} \cdot T_{\text{quad}}^2 = 2^{-4} \text{ ticks}, \tag{6}$$

which gives $t_{\text{quad}} \approx 5.8$ s.

The quadratic term, then, is impractical to store and of no practical use. There is no reason to include it in our time delay model.

3.3 Shifting and sizing delay-rate coefficient

As shown in Equation 5b, the expected maximum delay rate for terrestrial observation is 1.6×10^{-6} s/s $= 1.6 \times 10^{-6}$ ticks/tick. Since $\log_2(1.6 \times 10^{-6}) = -19.25$, the first 19 bits after the binary point in the register will be empty in this case.

For space observations with RadioAstron, the delay rate could be 50×10^{-6} ticks/tick, corresponding to 14 empty bits after the binary point.

This suggests that we should left-shift the linear coefficient, so as not to waste space. Exactly how far it should be shifted will be discussed in the next section, which is concerned with controlling errors.

3.4 Controlling errors due to linear delay coefficient

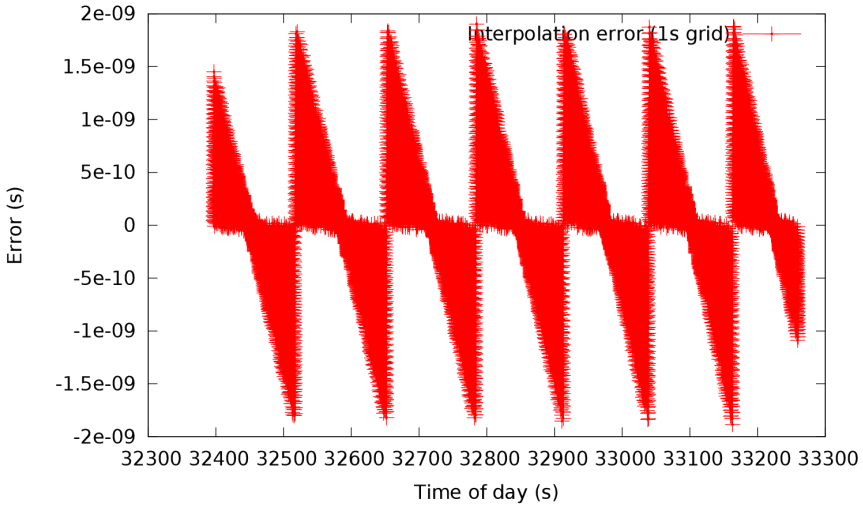

Figure 1: Delay errors with 28-bit shifted linear coefficient

We now turn to estimating the errors that will result from the fixed-point representations of the constant and linear coefficients.

We introduce the *time resolution*, t_{res} of the delay calculation, defined as the smallest unit of time stored in the constant coefficient of the delay polynomial. Since we have proposed eight bits after the binary point for this coefficient, $t_{\text{res}} = 2^{-8}$ ticks $= 1.22 \times 10^{-10} s$.

The maximum error in the constant coefficient as a result of quantization is thus

$$\text{Err}_0 = \pm \frac{1}{2} t_{\text{res}}. \tag{7}$$

We would like to have comparable accuracy for the linear term as well. The quantization error in the linear coefficient represents an error in representing the *gradient* of the delay function; the resulting error in the linear term will grow linearly with time, up to the maximum time for which the polynomial is valid.

If we use a linear coefficient with 28 bits after the binary point, the quantisation error due to the linear term over an interval of one second (32×10^6 ticks) is given by

$$\text{Err}_1 = \pm 0.5 \times 2^{-28} \cdot 32 \times 10^6 \text{ ticks} \approx 1.86 \times 10^{-9} \text{ s}. \tag{8}$$

To validate this prediction, we have evaluated the difference between the "true" delay[1] and the interpolation using the proposed model representation for the first scan

[1] Actually, we used Akima splines between CALC delays spaced one-second apart to represent "truth".

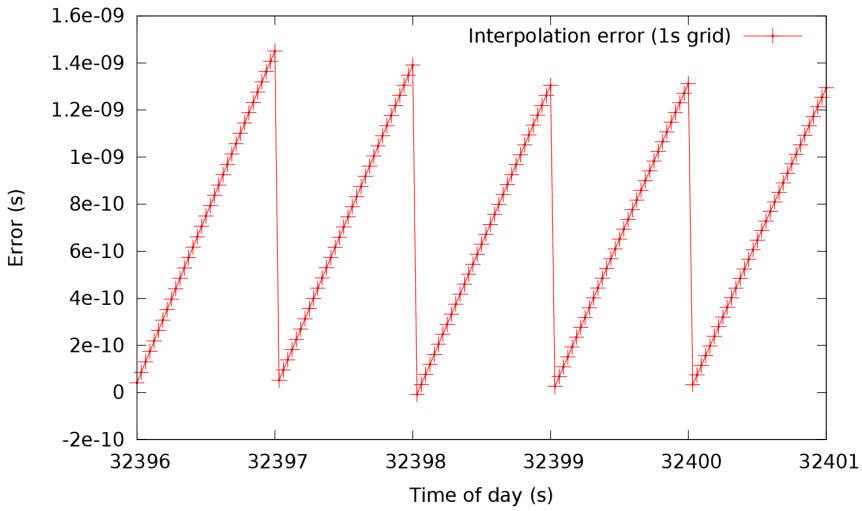

Figure 2: Delay errors with 28-bit shifted linear coefficient (detail)

of the EVN experiment EG065A. The results are shown in Figure 1, with a close-up in Figure 2. As expected, the error is dominated by the quantization error in the linear coefficient and grows linearly over each one-second interval.

The errors can be seen to peak at $\pm 2 \times 10^{-9}$, in good agreement with our prediction. This error is much larger than the time resolution of 1.2×10^{-10} s defined above. This suggests that we should reconsider the representation of the linear coefficient to achieve greater accuracy.

Figure 3 below shows the errors for a 32-bit shifted linear coefficient. The errors for this case are less than 1.5×10^{-10} s, which is comparable to the time resolution of the JUC's delay model.

The use of a 48-bit register in the constant delay term results from a need to accommodate delays of up to 2 s for experiments using the space telescope RadioAstron. Now, $2\,\text{s} = 64 \times 10^6$ ticks $= 1.9 \times 2^{25}$ ticks, so this requires an integer part of at least 26 bits. With 8 bits after the binary point, this won't fit in 32 bits but will fit in 48 bits, which is the next convenient size.

3.5 Phase coefficients

We also wish to assess the implications of quantization error for phase coefficients, using a similar methodology to that of Section 3.4. Clearly, the quadratic term in the phase polynomial will dominate quantization effects – it must accommodate a value of T^2 from zero to $(32 \times 10^6 \text{ ticks})^2 = 1.024 \times 10^{12}$ ticks^2.

It is proposed to add the nine most significant bits of the phase value to the inputs of the polyphase filterbank, so our polynomial must have this level of precision over an interval of one second.

Denoting the quantization error in the second-order coefficient for the phase polynomial by $\text{Err}(c_2)$, it follows that

$$\text{Err}(c_2) T_{\text{max}}^2 \leq 0.5 \times 2^{-9} = 9.8 \times 10^{-4}, \tag{9}$$

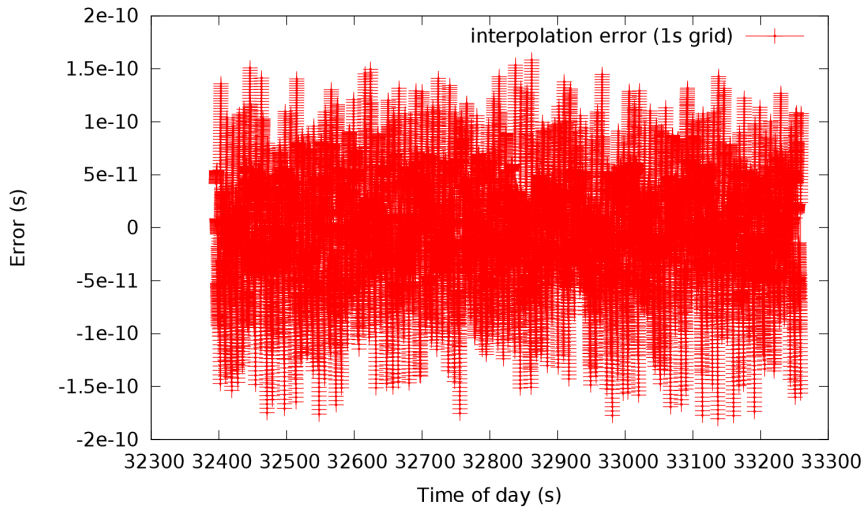

Figure 3: Delay errors with 32-bit shifted linear coefficient

so that with $T_{\max} = 32 \times 10^6$ ticks we require

$$\mathrm{Err}(c_2) \leq 9.54 \times 10^{-19} = 0.55 \times 2^{-59}, \tag{10}$$

which is to say that the second-order coefficient needs 60 bits.

It is also appropriate to reconsider the *a priori* estimates of Section 3.2 in the context of phase delays. For this purpose, we use a maximum frequency of $20\,GHz$. The constant term is not very useful, since given that we use units of cycles we effectively discard the integer part as a result of periodicity.

$$\max(c_1) \approx 3.6 \times 10^4 \,\text{cycles/s} \approx 0.011 \,\text{cycles/tick} \tag{11a}$$

$$\max(c_2) \approx 1.12 \,\text{cycles/s}^2 \quad \approx 3.5 \times 10^{-8} \,\text{cycles/tick}. \tag{11b}$$

This shows that there isn't any scope for shifting the phase rate register left. The phase acceleration register, on the other hand, will have zeros in its 24 most-significant bits for terrestrial experiments; it might be possible to exploit this fact to reduce the register size back down to 48-bits by shifting the register left as is done with the first-order coefficient of time delay, but it remains unclear what the phase acceleration might be for space experiments.

4 Implications of space experiments for delay coefficients

If it wasn't for space experiments with RadioAstron we could make do with linear time delay coefficients of 32 bits; with RadioAstron we need 48 bit coefficients.

The maximum delay acceleration coefficient for space experiments is unclear; it may require a second-order term for delay and it may put a bound on how far the phase-acceleration term can be shifted. The use of 64-bit unshifted coefficients for phase can be accommodated within the resources available on the Uniboard, though, so it does not seem important to reconsider this.

5 Transitional implementation

The current Uniboard firmware expects delay and phase-delay coefficients 32 times a second, with 32-bit coefficients for delay and 48-bit coefficients for phase.

This reflects the original design, and we have been waiting until the existing correlation firmware is mature and debugged before changing it. The existing control system code calculates delays over a one-second interval and then copies and adjusts the coefficients to make new coefficients (representing the same polynomial) over 32 smaller intervals.