Phasing Algorithm for the Phasing System

Geoff Crew[*] for the APP Team

February 20, 2013

1 Overview

This is a white paper on the algorithms and implementation options for the phasing component of the ALMA Phasing Project.

The phasing system is tasked with analyzing the correlated data from the individual antennas and constructing the optimal set of corrections to apply **in real time** so that the signals from the antennas may be added to produce a coherent sum for every channel being observed.

These corrections are applied via commands to the Tunable Filter Banks (TFBs). The phase-sum signal is reduced to 2-bits per channel and (a) fed back to the correlator (so that the efficiency of the summation process can be measured) and (b) passed to phasing interface cards which pass the result to recorders for eventual VLBI correlation.

The initial software implementation needs to lay out the interfaces among the various components of the phasing system; but the final choice of (optimal) algorithms and their tuning must await early further study and possibly commissioning tests.

It is expected that the detailed design of the phasing system implemention can be built on the formalism and options set down here.

2 Placement of Phasing Processing

The essential data available for the calculations are the channel-averaged data (see Section 7 for an explanation of this term) from all baselines as produced by the correlator and the WVR data available from each antenna. Both data types are available in the correlator data processing (CDP) as well as the telescope calibration (TelCal) computers. The actual phasing (*i.e.* the TFB) commands must be inserted into those already issued by the correlator control

[*] This (current) draft is based on notes from the first face-to-face session: R. Lucas on the algorithms, J. Kern on separation of the slow/fast phasing components; the work of R. Cappallo and L. Matthews who did the phasing study with 16 antennas; and various other discussions with many other members of the team over the past 2 years. Errors, incompletions, &c. are most likely my fault.

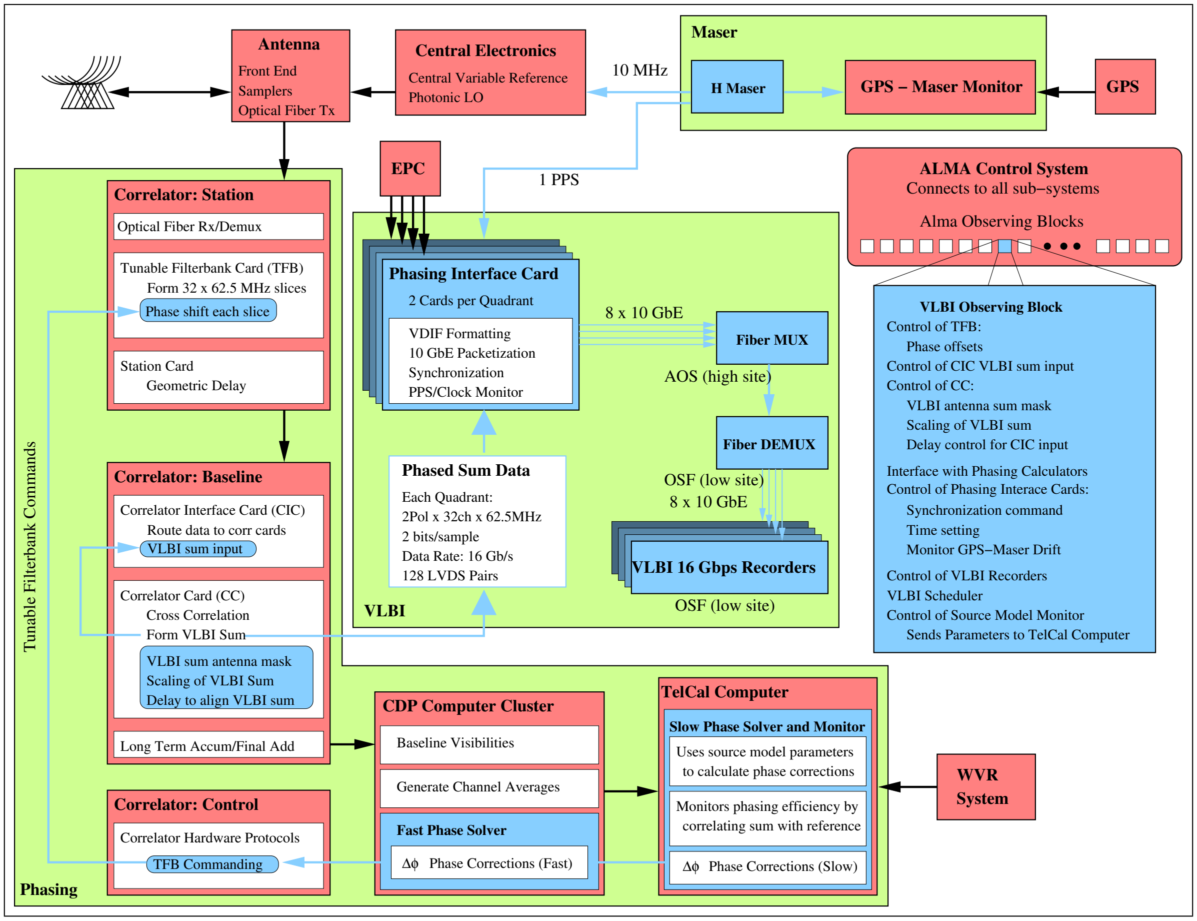

Figure 1: The APP Block Diagram. The phasing system activity is confined to the L-shaped green rectangle of correlator and TelCal computers. New components are in blue.

computer (CCC) to manage the channelized data. Note also, that the CDP computers are rather loaded with their normal correlation duties.

After considerable discussion of how best to implement the system without impacting the performance of the ALMA correlator it was deemed acceptable split the processing burden between the CDP and TelCal computers as shown in the overall project block diagram, Figure 1. This diagram is described in detail in the project plan, so we do not do so here. The phasing system is contained within the L-shaped green block; the pink objects refer to existing components, and the blue objects to additions for the APP.

This split is made such that most of the (hard) work is placed in TelCal which will, on a “slow” timescale provide instructions to a “fast” component operating in the CDP. The latter will simply decode its instructions, make simple adjustments based on the current time, and pass the necessary TFB command adjustments to the CCC for implementation. The former does all the hard thinking (to the extent that such is implemented).

Note that the latency of the required data to different parts of the system is not fully known at this time, and may be different for the different paths. The latest channel-average data is available within TelCal at the end of every subscan. Subscan durations are (typically) 2–30 s in duration, but require an overhead of (at least) 200–300 ms between them. Current WVR data is available from each station to both CDP and TelCal computers with a latency on the order of a second (**TBR,** but not necessarily the same latency to both).

We return to this issues in Section 9.

As an aside, it should be noted that providing channel-average data on shorter timescales where needed (*e.g.* on the integration [several dumps], or even dump timescales [512 ms]) is potentially possible with effort. The preliminary discussion of this option suggests that the effort level ("considerable") is rather beyond the current scope of the project.

3 Basic Algorithm

At a given time t, the correlator baseline data are channel averaged in each baseband. From these data, the baseline phases $\varphi_{OBS}(t)$ can be computed. Note that no reference here is made to the width of the band on which this phase is averaged—we return to this point later.

From the source model at large angular scales, the baseline model phases $\varphi_{MOD}(t)$ are also computed, using a table sent by TelCal. These are zero if the model is a point source.

The observed phases are assumed to be the model phases affected by (respectively) instrumental errors, atmospheric errors, the correction phases, and by noise:

$$\varphi_{OBS}(t) \;=\; \varphi_{MOD}(t) + \varphi_{INS}(t) + \varphi_{ATM}(t) - \varphi_C(t) + \varphi_N(t)$$

The goal is to phase the array for a point source in the field center, for which the model phases are zero; so the correction should as much as possible cancel the error terms, *i.e:*

$$\varphi_C(t) \;=\; \varphi_{INS}(t) + \varphi_{ATM}(t)$$

For completeness, we note that the atmospheric component includes a wet component for which WVR data is applicable as well as a dry component. Moreover, errors in the model phases or real source structure departing from the modelled structure can be considered subsumed in the unspecified instrumental error term. Likewise aside from thermal noise, the noise term might also include RFI and so forth. Without loss of generality, we could consider φ_{INS} to refer to systematic errors and φ_N to statistical errors and detail these later.

JA asks: can ALMA have the telescope pointed direction offset from the ALMA correlation phase center?

We have to estimate these various terms from the correlator baseline data and recent WVR data. However, the corrections we want to apply are per-antenna, so one does at regular intervals an antenna-based solution:

$$\psi_{OBS}(t) \;=\; L \cdot [\varphi_{OBS}(t) - \varphi_{MOD}(t)]$$

where L is some linear transformation and we have denoted antenna-based phases as ψ and baseline-based phases as φ. We may define a "reference" antenna which has zero phase, or more generally identify some other location which is assigned zero phase. We defer calculation of the transformation matrix L until later (in Section 5).

As with the baseline phases, we may express the per-antenna phases as:

$$\psi_{OBS}(t) \;=\; \psi_{INS}(t) + \psi_{ATM}(t) + \psi_N(t) - \psi_C(t)$$

The antenna-based noise $\psi_N(t)$ is actually reduced by a factor $\sqrt{N}$ from the baseline-based noise φ_N (where N is the number of antennas). It is further reduced by getting a longer integration time, but the atmospheric term may be fluctuating on a time-scale of about one second. The instrumental term however is slowly varying and can be assumed constant on a time scale of tens of seconds.

The adopted scheme is then to split the correction into fast and slow terms:

$$\psi_C(t) \;=\; \psi_{CF}(t) + \psi_{CS}(t)$$

and use two correction loops, of cycles t_f (~1 s) and t_s (~10 s). The timing of these cycles is set by external constraints: the availability of WVR data, subscan timing, and a requirement not to introduce new timescales into the analysis.

In the fast loop we use as a fast component the WVR correction calculated using the WVR data in the previous interval (*i.e.* data available at t_f), and a constant slow component (from Telcal, deemed applicable at time t_0):

$$\psi_C(t) = \psi_{WVR}(t - t_f) + \psi_{CS}(t_0)$$

This is applied every t_f interval, until t_s is elapsed. The WVR corrections are calculated as in normal interferometry observations. We defer until Section 4.3 a more precise description of the actual WVR correction.

In the slow loop (*i.e.* in TelCal) at the end of t_s, we use the time-averaged observed data (denoted by $< ... >$ as described above), to get the correction for the next interval $t_0 + t_s$.

$$< \psi_{OBS}(t) > \;=\; (< \psi_{INS}(t) > - \psi_{CS}(t_0)) + (< \psi_{ATM}(t) - \psi_{WVR}(t) >)$$

That is, we will use:

$$\psi_{CS}(t_0 + t_s) \;=\; < \psi_{OBS}(t) > - \psi_{CS}(t_0)$$

thus including in the next slow term the change in instrumental correction and a possible slow systematic error in the WVR estimate of the atmospheric phase. The antenna based noise is here assumed to average out.

The slow correction discussed here, is in general either some slowly varying predicted phase $\psi_{CS}(t_0, t)$ from TelCal's calculation on the subscan timescale, or else (assuming all the baseline data is available and computational resources are adequate) a simplified linear transform $L(t_0, t)$ which itself could be (slowly) linearly varying or constant.

If the solver retains some history of the transformation it applies, it can choose to switch from a constant transformation for the fast loop to a slowly varying one (*i.e.* as a low order polynominal in time with fixed coefficients).

4 Some Specific Cases to Consider

We consider (briefly) certain cases worthy of note. There may be others.

There are two measurable quantities that are relevant to our choices for dealing with weather conditions (good or bad): stability and opacity. Good weather is both stable and has low opacity, and to some degree the two are correlated. But one can have very stable atmosphere with poor opacity, good opacity but unstable air, and of course both can be poor.

Low opacity is somewhat degenerate with source strength; the line between sources strong enough to support phasing and those which are too weak will depend on the opacity.

An additional consideration is the spatial configuration of the ALMA array. It may be the case that, *e.g.* , the longest baseline is a further discriminant of what approaches may be applied to specific sources based on weather conditions.

4.1 Very Good Weather

In that case the atmosphere phase fluctuations are very small and the WVR correction would only add noise to the correction. In that case it is better to set $\psi_{CF}(t) = 0$ at all times, using only the slow correction loop. Or, more generally, use a weight w_f such that

$$\psi_C = w_f \psi_{CF}(t) + (1 - w_f)\psi_{CS}(t)$$

To enable good performance for this case, the fast loop would need to be told NOT to use the WVR data.

4.2 Weak Source

In that case the science target source is too dim be used to phase the array, and a nearby quasar would be used, observed in a dedicated scan of duration t_S. That scan would be used to estimate $\psi_{CS}(t_0 + t_s)$ which would then be used during the observing scans on the project source, until the next observation on the phasing calibrator.

Again, depending on the weather as discussed in the previous section (4.1), the WVR data could be used or not in the fast loop.

Note also that if the atmosphere is not stable, phasing-up on a bright source and then switching to the dim source is probably less effective than using longer integrations on the weak source.

4.3 Global WVR Correction

When using the WVR data for corrections, there are in fact two options:

1. correct all antennas with their WVR normally
2. correct each antenna with the difference of its WVR correction and that of the reference antenna.

The advantage of the former is that all antennas are treated equivalently.

The potential advantage of the second approach is that it allows an after the fact WVR correction of the entire array using the data from the reference antenna. (Of course, if you have the complete WVR history of that one antenna, you can undo the phase adjustment of (1) to get back to the equivalent of (2) before doing a more careful WVR adjustment.)

5 Antenna-based Solution

There are a number of possible algorithms to convert the baseline-based phases to antenna-based phases via some operator. Our baseline implementation is a least squares approach discussed in the next section (5.1). We mention other options in the following sections which might be useful for numerical studies, or commissioning testing.

5.1 Least Squares Solution, General Derivation

In the determination of the linear transformation L we can use a general linear least squares fit approach to a solution. For N antennas, there are $N(N-1)/2$ baseline measurements, and $(N-1)$ antenna phases. (The phase of the entire array may be set arbitrarily in a number of different ways.) Since a baseline phase measurement φ_{ij} between antennas of index i and j may be expressed in terms of the unknown per-antenna phases as $\psi_i - \psi_j$, we can form a χ^2 as

$$\chi^2 \;=\; \sum_{i \neq j} \frac{[\varphi_{ij} - (\psi_i - \psi_j)]^2}{\sigma_{ij}^2}$$

One can (following the standard treatments) require $\partial\chi^2/\partial\psi_i \;=\; 0$ for every i to find the minimum of χ^2 and arrive at a linear system of equations which may be solved for ψ_i. If the data is not too noisy, the minimum found will even be the correct one.

The equations are worked out below (Section 10.2) in the context of a specific example. A revised version of this draft should complete the derivation and include the equations here.

5.2 Least Squares Solution, using Reference Antenna

In the determination of the linear transformation L we use a least squares solution, and introduce the reference antenna as index r. [This is the phrasing from R. Lucas' original draft.... It presumes one can introduce the reference antenna and rearrange to solve for the antenna phases ψ_i in terms of φ_{ij} and φ_{rj} but the derivation is missing, and would be needed to code this up.]

If the baseline data have weights that can be split into antenna-based weights w_i, the antenna-based solution for phases ψ_i can be expressed as:

$$\psi_i \;=\; \frac{1}{W}\left[\sum_{j,j\neq i} w_j\varphi_{ij} - \sum_{j,j\neq r} w_j\varphi_{rj}\right]$$

where W is the sum of antenna weights, and r is the index of the reference antenna. So the L matrix is essentially determined by the antenna weights, that may be assumed constant over a slow cycle ot the algorithm.

Note that the phased sum must be constructed using the 2-bit data available for each channel, so the sum is going to be T_{sys} weighted by antenna, so those may be the starting weights to use here.

If we have to down-weigh long baselines where most of the source flux is resolved, then the weights cannot be split any more and the solution has to be derived numerically. But the weights should not change fast with time, and L should still be essentially constant over a slow cycle of the phasing algorithm.

5.3 Selected Baselines to Selected Reference Antennas

The SMA phased array system in current use in the Event Horizon Telescope does not have the option of solving for antenna phases using the data from all baselines. A simpler algorithm was adopted which consists of selecting a (central) reference antenna, assigning it zero phase, and attributing the entire baseline phase from each remote-reference pairing to the remote antenna. Results with this approach are shown in Figure 2 which shows a clear square-root increase in correlation amplitude with the number of antennas.

This could be directly implemented for ALMA, however for some of the longer baselines, the errors might be too great. One could use a "tree" of reference antennas, where a central antenna provides phases for a central group. Then a set of these act as references for a secondary group (and so on).

This is the computationally quickest option, but it would also be noisiest.

The source must be strong enough to be detected on all baselines to the reference antenna.

5.4 Singular Value Decomposition

The least-squares approach assumes that the normal equations are not singular and are also numerically stable. A general treatement for handling such case is available via singular value decompostion (SVD). This method has the advantage of automatically finding and excluding the noisiest baselines.

While this method is possibly not needed to achieve reasonable solutions, it may be of some use to the phase solver for identifying problematic antennas. *I.e.* if one or more antennas are noisier for some reason, their noise will pollute the otherwise good phase solutions on the other antennas. SVD may be useful to help identify them.

5.5 Hybrid and other schemes...

Obviously depending on the arrangement of the antennas, one could partition them into groups and, *e.g.* , perform least-squares solutions on each group with an additional fit on a set of baselines to tie them together.

JA suggests: Robust fitting, or non-least-squares fitting.

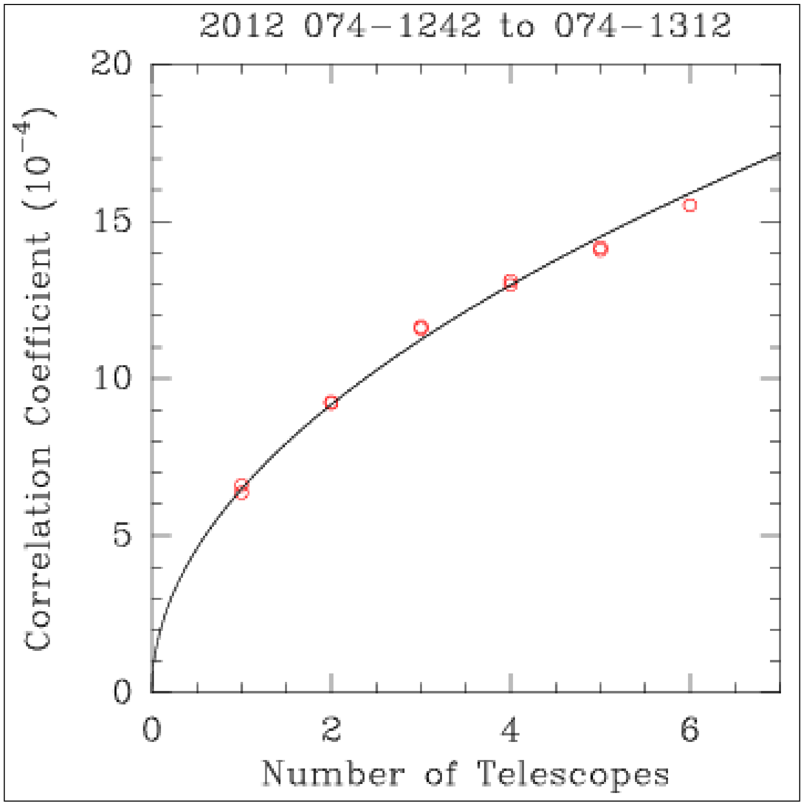

Figure 2: Dependence of correlation efficiency with the number of antennas using PHRINGES (J. Weintroub, private communication).

6 Delays and Other Complications

6.1 DelayServer

There is a global process, DelayServer which projects what the delays between antennas are likely to be in the near future (*i.e.* over the next few minutes, at least). Various agents within the ALMA system subscribe to this projection to make appropriate adjustments. From the correlator perspective, the delay is partitioned into multiples of samples (250 ps), fractions of the sampler clock (250 / 16 ps), and some residual (*i.e.* the piece less than 250 / 16 ps).

The multiple-sample correction is made by adjusting the relative alignment of the samples from the different antennas. The fractional part is made by adjusting the phase of the digitizer clock (DGCK).

A correction for the residual is applied in the CDP computers in the generation of the channel-averaged data. Either this piece can be ignored, or the PhaseSolver can subscribe to the DelayServer model to be aware of it, or potentially the CDP computers can be commanded not to apply the residual correction.

The resolution of adjustments to the ALMA delay model is 15.625 ps (250 / 16 ps), so that across a 2 GHz band this is 12° of phase. Across a 62.5 MHz band this phase slope is negligible (0.35°).

Note that the preceding sections assume that the residual delays between the antennas are negligible. (*I.e.* that the single field interferometry mode is has correctly modeled and removed the delays between antennas). Thus the algorithm discussed above is not required to find and remove a residual delay separately from the (random) phase adjustments.

The study described below in Section 10 suggests that the ALMA correlator does indeed to an adequate job with delays, but that it is not necessarily as good as the 12° of phase. Hypothetically, the phasing system can be aware of whether the projection is in fact accurate, and potentially it could be involved in providing real-time corrections (within the 250 / 16 ps commanding range).

A further consideration regarding delays is that the summed-antenna signal that is fed back through the correlator replaces one of the antennas in the array. If no action is taken, the CDP processing will treat the data from the sum as if it came from this antenna; hence the geometric delay will (in general) be incorrect. Note there is also a processing delay (10 clock cycles in the correlator) added due to the time required for the correlator logic to compute this sum.

The simplest solution is to dynamically inform the DelayServer of a new position for the "sum" antenna with each new target as the array phase center delayed by these 10 clock cycles.

6.2 Significant Residual Delay

The analysis of Section 3 presumed that there was no significant residual delay. If, however, if a significant delay is present, then it is necessary to solve for this on every baseline, in addition to solving for the phases.

It can be taken as a very good assumption that the variation of delay with frequency (channel) due to the atmosphere is linear. (*I.e.* there is no point allowing some curvature.)

On the other hand, instrumental effects may introduce other behavior.

6.3 Band Pass Correction

Presumably each antenna has its own (stable) bandpass imperfections which can be measured with a calibration scan on a bright source. This correction could be explicitly made, rather than leaving it to the fitting algorithm to remove.

6.4 Discarding Antennas

Presumably either the solver or the efficiency monitor (Section 8 below) may determine that the overall situation would be improved by removing pairs of antennas from the array. (Antennas must be removed pairwise so that the statistics of the phased sum when reduced to 2 bits remains the same as that of a single antenna.)

7 Channel Average Data

In the designated mode of operation (Mode 13) each quadrant will have 32 TFB sub-bands defined in such a way as to allow eventual VLBI correlation. The 2 GHz of bandwidth is decomposed into 2048 spectral points (with a spectral resolution of 976 kHz on each "channel"). Thus each TFB will have 64 spectral channels.

Apparently, the specification of how to average channels for the so-called "channel-average data" is not tied to the TFB specification. Thus, one might specify these to be the innermost 60 channels. (Remember that the filter is such that only the inner 15/16 of the sub-band is considered usable.)

Alternatively, for the purposes of solving the phase problem, one might presume that the phase is a slowly varying function of frequency (*i.e.* "delay-like", possibly with some dispersion); and group together larger numbers of channels to improve statistics.

This may also be important for computational reasons. (The computational effort is proportional to the number of sub-bands.)

Finally, if spectral lines are present, well, setting up the channel averages is even more complicated, but presumably the span of channels over which to average will be limited to the line (or lines). Of course if the source has multiple lines coming from different regions, it gets even more complicated....

Again, for analyzing the ALMA data, the channel average data together with the full spectral, or spectral averaged data should be available.

8 Efficiency Monitoring

Depending on the solution procedure adopted, it may provide some measure of the quality of the solution. (*E.g.* in the least-squares approach the likelihood of χ^2 could be evaluated and reported; but that's probably not too useful.)

Aside from that, there is a powerful diagnositic provided by the correlations of the sum-signal with that of all of the antennas available to the correlator. It is sometimes useful (here for this purpose, but also with the polarization calibration) to leave one antenna out of the Array. Such an antenna is termed the "comparison antenna". Alternatively one can simply consider one of the antennas participating in the sum. Either way, the correlation of the distinguished antenna with the sum should scale with the number of antennas in the array, N. See, *e.g,* Figure 2.

The simple argument is that the antenna sum is one of N random (real) variables s_i:

$$S \;=\; s_0 + s_1 + ... + s_{N-1}$$

so for any i

$$\begin{aligned} < Ss_i > \quad &= \quad < s_i^2 > + \textstyle\sum_{j \neq i} < s_i s_j > \\ &\equiv \quad [1 + \eta(N-1)] < s_i^2 > \\ \eta \quad &= \quad \tfrac{1}{N-1}\left[\tfrac{<Ss_i>}{<s_i^2>} - 1\right] \end{aligned}$$

where we've introduced and solved for an efficiency η, noting that $< s_i s_j >$ is at most 1 for perfectly correlated signals, 0 for uncorrelated signals, -1 for anticorrelated signals, and in general, somewhere in between.

Note that this only works if:

1. the autocorrelation $< s_i^2 >$ is available and
2. sufficient precision and integration time is available in order to remove the very strong autocorrelation signal from the receiver noise

so if a comparison antenna is available it gives a better measure of the efficiency. Similar algebra applies for the signals correlated with comparison antenna C:

$$\begin{array}{rcl} < SC > & = & \sum_j < C s_j > \\ & \approx & (\eta N) < C s_i > \\ \eta & = & \frac{1}{N}\left[\frac{<SC>}{<Cs_i>}\right] \end{array}$$

Note that as we've defined it η can go very negative and approach -1 (which means s_i is anti-phased relative to all the rest).

Note that we do not have access to the raw signals, but rather to the correlator channel-average data products. Similar logic applies to the visibilities, but there are probably minor corrections to be made here. . . .

9 Data Interface Considerations

For reference on the details, the supporting documentation for the current ALMA implementation is captured in these documents:

- `2011-12-20-ALMA-60.00.00.00-70.40.00.00-D-ICD.pdf` the ICD
- `CCC-SCC-CanProtocolVerB-64Ant.doc` with more implementation details
- `2011-10-19-CORL-60.01.07.00-002-H-MAN.pdf` the TFB manual

The phasing system involves a loop; one must start somewhere, and we'll begin with the commanding...the following sections probably can be reordered for clarity later.

9.1 Testing and Simulation

Since one half of the phase loop lives in the correlator and the other in TelCal, it is highly desirable that the interface between the two halves be constructed such that a complete TelCal or Corr system is not need for some of the testing.

9.2 Commanding Authority and Timing

The TFBs are a component of the station cards in each correlator, and the commands which actually adjust the phases are issued by the correlator control computer (CCC) to these cards. Each quadrant has a dedicated CAN bus (so bandwidth is limited to 1 Mb/s to each quadrant), and these commands can proceed more or less independently. (The only shared resource here is the CCC itself.)

These TFB adjustments are delivered in two phases: a `DOWNLOAD_TFB_PHASES` command with the phases to use, and a `APPLY_TFB_PHASES` which transfers the commanded phases from temporary storage to active use at a specified time. (See Fig. 3 for the underlying logic.) The former takes about 220 ms to deliver (12 bits per phase for 2 polarizations on 64 antennas, delivered all at once), while the latter is a short, quick message (about 1 μs). The phase is actually represented as a 16-bit datum, but the upper bits represent phase wrap (*i.e.* multiples of 360 degrees).

The `APPLY_TFB_PHASES` does not currently exist; but it will be defined to update ONLY the phase offset register (from the temporary loading register in to the active register) at some specifed TE (possibly the next one). The existing APPLY command is used at the start of each (sub)scan to fully program the station cards (and we definitely do not want to do that).

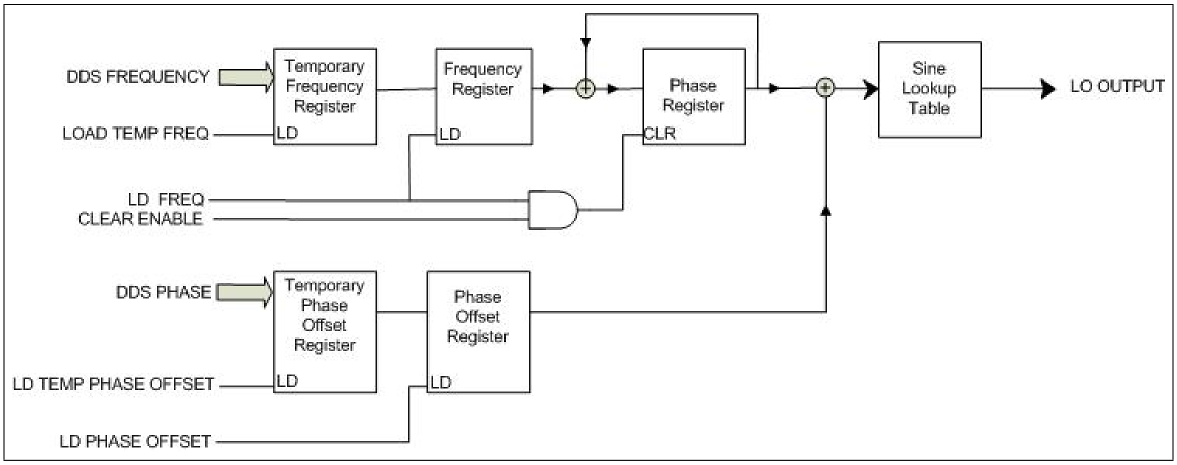

Figure 3: A simplified view of the TFB logic. The frequency and phase offset are downloaded to temporary registers, and brought into use by deliberate load commands at precisely determined times. Each clock cycle produces a phase-based lookup into the sine value for the LO output.

Note that the normal commanding procedure would use two additional commands to verify that both the phase download and apply commands succeeded. The CAN bus is generally reliable, but we should be prepared the verification to fail.

A complicating factor is that the temporary phase register is used in the delay management. An entity called the DelayServer propagates a model of delays to each antenna periodically (*e.g.* every few minutes) and asynchronously. This model delay is decomposed into separate portions for application in various parts of the array (correlator, DGCK &c.) The delay in force at the start of a subscan (to a resolution of 250/16 ps) is used to calculate phases to insert into the phase offset register of each TFB. During a subscan, the correlator will adjust the delay in multiples of 250 ps (sample delays) and the digitizer clock (DGCK) will adjust the delay in multiples of 250ps/16 (sub-sample delays). The residual delay (less than 250/16 ps) is cleaned-up by the CDP processors post-correlation (for the archive) and does not concern us here.

Each adjustment using this fractional delay will have placed some phase psi_f into the phase offset register at some time t_f. Thus, if the phasing system desires to apply some addition phase adjustment psi_a at time t_a it needs to know what this was and incorporate that into the planned update. This could be accomplished with the following pseudo-code adjustment to the existing phase update mechanism:

```
subscan_phase_adj = delay_server_adj
subscan_phase_adj += phase_update_adj(subscan_phase_adj, apply_time, ...)
```

where `delay_server_adj` is the phase adjustment as currently calculated, and the function `phase_update_adj()` returns 0 in current usage or in the future when the phasing system is not active. With an active phasing system, it would calculate and return the appropriate additional phase adjustment from the phasing system (and in the process be notified that the delay has changed).

Equivalently, `+=` could be replaced by `=` according to what `phase_update_adj()` chooses to return. This largely doesn't matter, except for how the phase update is done independently of the normal subscan phase updates.

The CAN bus is heavily utilized at the start of a scan (programming station cards, LTAs &c.); a process taking about 2 seconds. It is also heavily utilized at the start of each subscan (*e.g.* sending delay phases). Otherwise, there is CAN bus activity associated with delay updates which may interfere with the phasing system commands.

On the other hand, if dropping the occasional phasing system adjustment on the floor is an option, we appear to have the commanding capability for a few adjustments per second.

9.3 CDP Processing

Here are some Q & A notes from email with Rodrigo Amestica.

9.3.1 Latency

Q: One of the RFAs we got at PDR concerned the latency of data around the phasing loop. (We knew we needed to look into this anyway, so I'm glad it was asked.) So: are there any measurements on how long it takes data to move through the correlator, CDP and (utimately to TelCal)? I can imagine that the path from the inputs to the station card to the correlator card output are well-determined, but I've no idea what happens after that (other than that the CDP must keep up).

A1: On every sub-scan completion I log the sub-scan overhead. This parameter is computed as now (current wall time) minus the moment at which the last integration ended in the correlator hw. The overhead is a function of the correlator mode (for phasing you will probably always be thinking FDM) and number of antennas. My experience so far, with less than 40 antennas is that the overhead in FDM is not shorter than 2.5 seconds. (This is why I was wondering about observing efficiency, if the sub-scans are short like $\sim$10 seconds....)

(It will be Mode 13 (we need all cross products and 32 TFB channels on full resolution). IF you have this as a function of antenna number that is probably something someone will ask for eventually. I'm assuming that the correlator continues to process while the CDP is doing its work, so this should not affect our efficiency. But if we need to stop the correlator between subscans until TELCAL is caught up, that would be important to know.)

A2: The moment at which I log 'overhead' is right before TELCAL should had received the last bit and, therefore, the moment at which it has started processing the data for that sub-scan. I do not know what TELCAL's overhead looks like. My understanding is that TELCAL does not start processing until the last bit for the current sub-scan has been received, therefore, the total overhead would be CORR+TELCAL.

(Yes. The TELCAL part of the question has been enqueued...and it is relevant to the "slow" loop which allows TELCAL to think.

However, the "fast" loop potentially only involves the CCC and CDP. For example, a matrix multiply of a vector of baseline phases to produce a vector of antenna phases on one or more channels. IF the baseline phases are all available after each integration then the new antenna phases can be calculated and sent to the CCC. How short might that delay be in that case?)

A3: I do not have a clear cut answer, but for four base-bands (16 GHz) and 63 antennas the minimum dump duration is $\sim$8.1 seconds (see below). Assuming that the matrix multiple is not a big issue and also understanding that from cdp to ccc we need to send just a few integers per node (each one been the phase setting for a TFB phase register) then I do not think that the delay should be of importance compared to the $\sim$8.1 seconds per dump.

The minimum dump time in mode 13 is 0.512 seconds, but the computing infrastructure (software backend, ACS transport mechanism (aka bulk data) and networking hardware) only supports an average of 60 MB/s. There are plans to improve this in the future.

The data limit that must be respected comes from the following calculation:

```
       4 correlation products
    2048 spectral points (from 32 TFBs with ~ 1 MHz resolution)
 64*63/2 baselines (at most)
******** (multiply together to get)
16515072 data items /dump /quadrant
         (ignoring overhead bytes)
      8B /data item (Re,Im as 4B floats or equivalent)
= 132 MB /dump /quadrant (1024*1024B/MB)
= 528 MB /dump
= 8.064 sec (at 62.5 MB/s)
```

Slightly more time should be allowed for a safety margin.

Note that this data can be discarded to remove the constraint. As a better option, there is a spectral averaging function that reduces the data by 2^n for some integer $n < 11$.

9.3.2 Using WVR Data in CDP

Q: We've talked of making (or not making) WVR corrections in real time in the CDP computers. IS this currently implemented, or is this something that is in development for some future release? IF it is implemented, is it something that can be easily turned on or off? (In the big picture, we might choose to do it in Telcal or in the CDP computers, but doing it twice would be wrong.)

A: It is already implemented. Its currently under commissioning. The spectral specification commands the software what to deliver:

- only uncorrected
- only corrected
- both

so, you can freely command this setting from the VOM.

9.3.3 CDP Residual Delay Correction

Q: Is the residual delay calculation in the CDP something that can be disabled? (*I.e.* the data goes uncorrected for the residual delay $<$ 250/16 ps.)

A: Yes, it can be disabled, but only informally. That is to say, that it is not part of the spectral specification, but a "back door trick" unsed now and then for testing purposes (IDL interface method in ObservationControl). If this is required, there would be some work to formalize it.

9.3.4 Processing Limits

No quantitative answer yet. The CDP Master is data volume limited (60 MB/s) and the CDP nodes are processing limited.

9.4 Data Interfaces Available in CDP

- Need to track down how the baseline data is available in the CDP context.
- Need to track down how the WVR data is available within the CDP context.
- Need to track down precisely how it is being used now.
- Need to work out details of the PhaseUpdate component (lives in CCC and perhaps farms work out to the CDP nodes).

9.5 Data Interfaces Available in TelCal

Helge can summarize here what he learned in Grenoble...

- Need to unpack the SDM Tables and work out how the PhaseSolver will pull in the most recent subscan dumps for its “slow” calculation.
- Need to understand how the source model will become available. Will it be something already in the archive, or will it be something delivered by the VOM?

9.6 Interface between `PhaseCommander` and `PhaseUpdate`

As indicated previously, the “fast” and “slow” components communicate with each other such that on the slow timescale ($\sim$10 s) the `PhaseCommander` will need to send to the `PhaseUpdate` (in general) the following:

- whether to use WVR data on fast timescale (or not, or more general w_f)
- whether to use per-antenna phase ψ or operator L or both
- applicability time t_0
- constant slow phase ψ_0 at t_0
- (higher order polynomial coefficients for ψ with $(t - t_0)$)
- constant operator L_0 at t_0
- (higher order polynomial coefficients for L with $(t - t_0)$)
- delay correction δ_c (see Section 6)

Note that there are 32 channels which need to be handled, so a phase ψ_0 refers to 32 channels and 63 antennas. If the phase variation with TFB is asserted to be linear, a phase and slope per antenna suffices.

In general, one might want separate L operators for each TFB, but in practice, average per-antenna-band properties might suffice so that the same L could be used for all channels. With at most 63 antennas and $63 \times 62/2$ baselines, L is at most a 63×1953 matrix, we are talking $\sim$ 123039 floating point quantities per slow cycle ($\sim$98 kB/s) per quadrant per polarization.

Formally this matrix has size of $O(N^3)$ but the matrix discussed in Section 10.2 can be factored and decomposed with a factor that is sparse, so with careful packaging, only $O(N^2)$ quantities need to be sent. If one is willing to assume a smooth variation of phase across the channels (*e.g.* a linear delay-like variation) an additional order of magnitude reduction is possible.

Potentially, L need only be sent once, or could be constructed within the CDP computers, or need only be performed on the slow timescale.

ToDo: find out current link capacity limitations...

9.7 Polarization issues

The polarization study has recently conclude that the most efficient solution for coping with the linear to circular conversion required by conventional VLBI analysis codes is to perform this conversion as part of the VLBI correlation of the data. To do this, however, requires that the X-Y polarization delay be minimized. If this is to be enforced during the adjustment of the TFBs, then the `PhaseCommander` → `PhaseUpdate` conversation might be the best place to insert this adjustment.

ToDo: work out exactly what is required here.

JA asks: "for pol calibration ad for the TFB conversion to 2 bits, what do we do about sources that are time-variable in I, Q, U or V or here [sic] spacially variable?"

10 The Initial 16-Antenna study

To characterize the effects of the Earth's atmosphere on our ability to phase ALMA, we have investigated the coherence losses caused by atmospheric fluctuations in ALMA test observations of bright quasars taken in several ALMA bands. These data sets were kindly provided to us by V. Impellizzeri and were processed using standard tasks available in CASA and AIPS.

10.1 Atmospheric Characterization

We show here results from the analysis of Band 6 (230 GHz) observations of the quasar 1924-292 taken on 2012 May 10 at ∼50 degrees elevation (`uid___A002_X3ffc69_X85`). The mean precipitable water vapor was ∼0.9 mm and mean RMS path fluctuations on 300 m baselines were ∼125 microns (Figure 4), both of which reflect typical May conditions (`https://almascience.nrao.edu/about-alma/weather`). The 1 mm flux density of 1924-292 on this date was ∼4.0 Jy, similar to that of Sgr A*. Eighteen one-minute scans of the quasar were performed over the course of ∼90 minutes with ∼4 minutes separating each scan. The sampling rate was 0.962 seconds. Fifteen working antennas were present in the array, representing 23% of the collecting area of the full ALMA array. Baseline lengths ranged from 42.7 m to 2.29 km. In Figure 4 we illustrate the effectiveness of applying phase corrections to the data based on the water vapor radiometer (WVR) measurements collected at each antenna. The corrections were computed using the wvrgcal program as implemented in CASA. We found a significant decrease in the phase fluctuations (expressed as RMS path differences) for most baselines after application of the WVR corrections.

In Figure 5 we explore the time scales on which calibration solutions need to be computed in order to sample the residual atmospheric fluctuations not removed by WVR corrections and thus preserve adequate phase coherence. We find that for WVR-corrected data, we are able to tolerate delays of up to 10 s in our phasing solutions without loss of correlated amplitude $> 5\%$. We conclude that we will be able to correct our data to high accuracy under typical observing conditions.

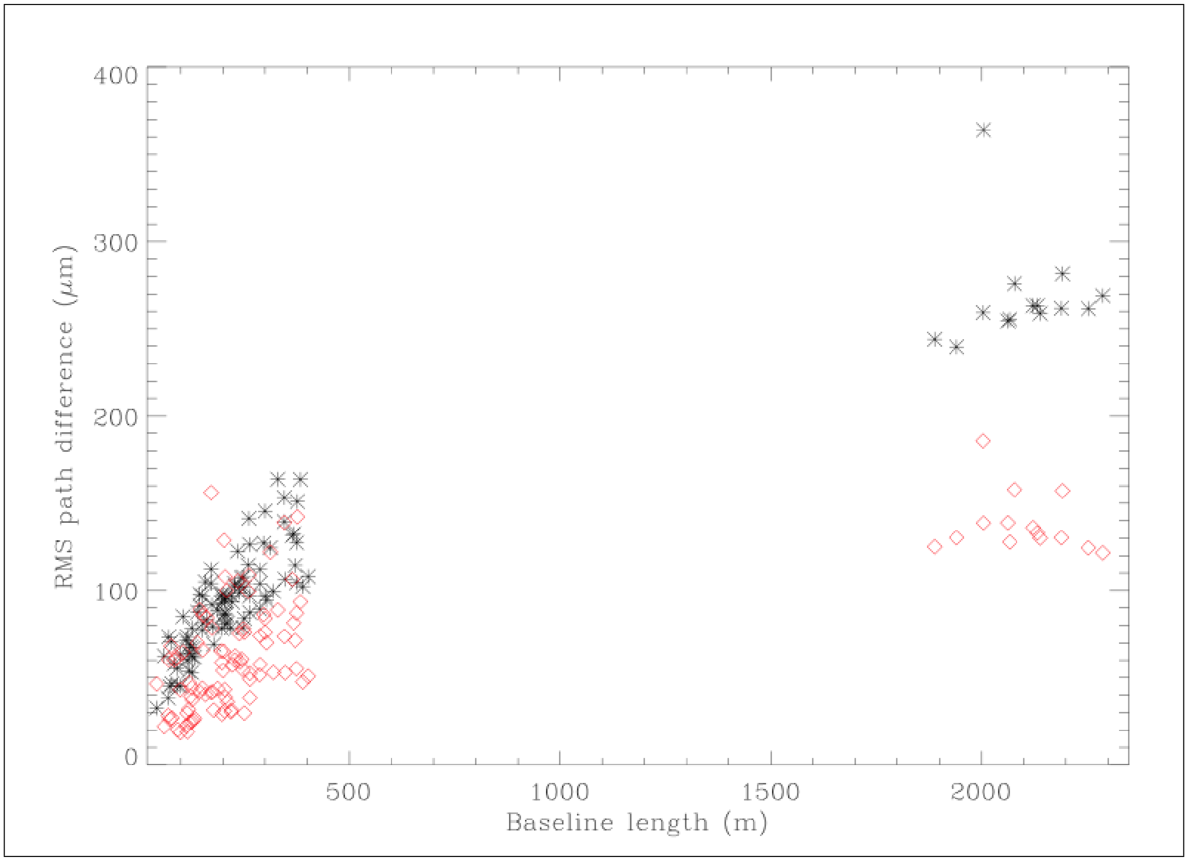

Figure 4: RMS path fluctuation as a function of baseline length during a 90 minute observation of 1924-292 at 229.6 GHz on 2012 May 10. Black symbols represent raw data values; red symbols are the same data after the application of WVR corrections. The data have been vector averaged over a 1.8 GHz band.

Finally, we plot in Figure 6 the signal to noise ratios for the 16 antennas which correspond to the self-calibration carried out for Figure 7. Using a 1-second solution interval, the SNR for all antennas was ∼250 as shown in Figure 7. Longer solution intervals produce higher SNR results, but with more dispersion due to atmospheric effects.

10.2 Simple Least Squares Solution

As a first step in developing an operational phase solver for ALMA we implemented some of the necessary algorithms in MATLAB, and applied them to the same 16-antenna data as used in our coherence studies above. We used non-weighted linear least squares to estimate a phase offset for $(n-1)$ antennas to a reference antenna, independently for each correlator dump (0.96 s). The normal equations and partial derivatives have a simple form due to the nature of the data, which are phase differences between antenna pairs: $Ax = B$ yielding

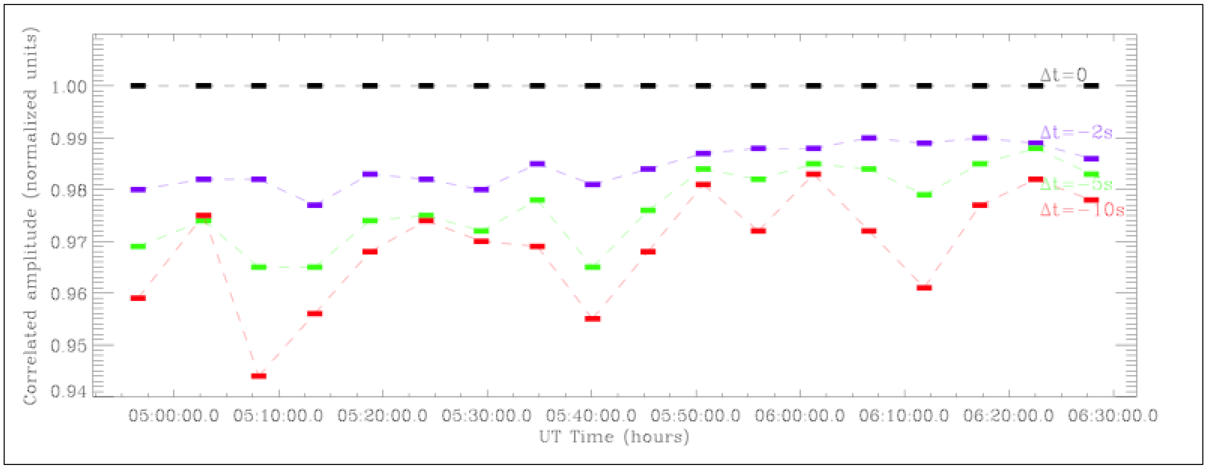

Figure 5: Correlated amplitude as a function of time for observations of 1924-292 on 2012 May 10 (XX polarization). The data were averaged over a 1.8 GHz band centered at 229.6 GHz. WVR corrections and self-calibration with a solution interval of 1 second have been applied. The data were then averaged over 1minute time intervals and vector averaged over all baselines for display purposes. Data plotted in black have optimal WVR and self-calibration corrections applied. The horizontal extent of the rectangular points indicates the duration of individual scans and the dashed lines indicate intervals during which there are no data. The purple, green, and red curves show the result of offsetting the selfcalibration solutions in time by 2, 5, and 10 seconds, respectively.

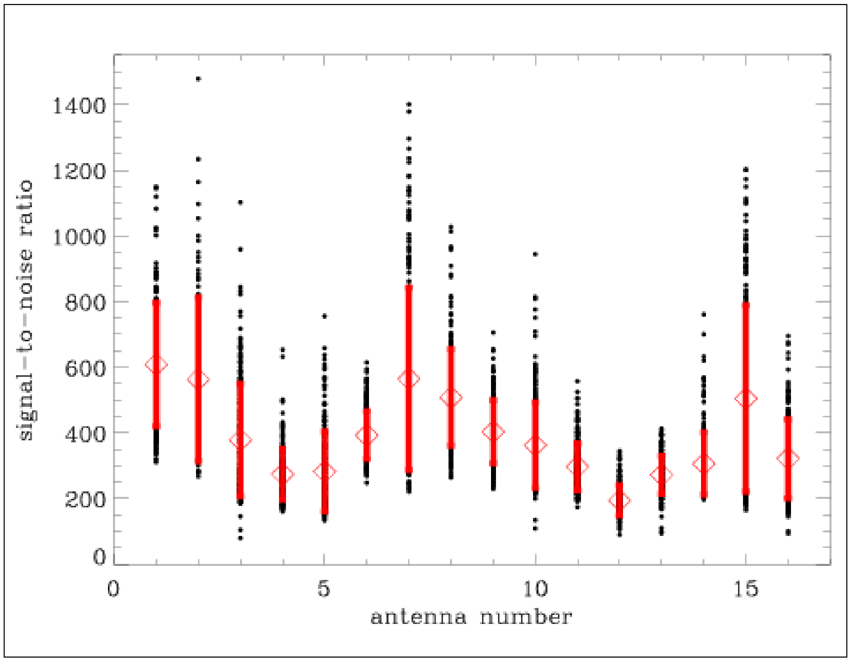

Figure 6: Signal to noise ratios of the antenna-based self-calibration solutions for the ALMA test data presented in Figure 4. Each point represents a 10-second solution; the diamonds indicate the average SNR and the red vertical bars indicate the one sigma dispersion. These solutions were obtained for a 1.8 GHz bandwidth on a 4 Jy source (approximately the flux density of Sgr A*). Using the full bandwidth of ALMA would increase these SNRs by a factor of ∼2.5.

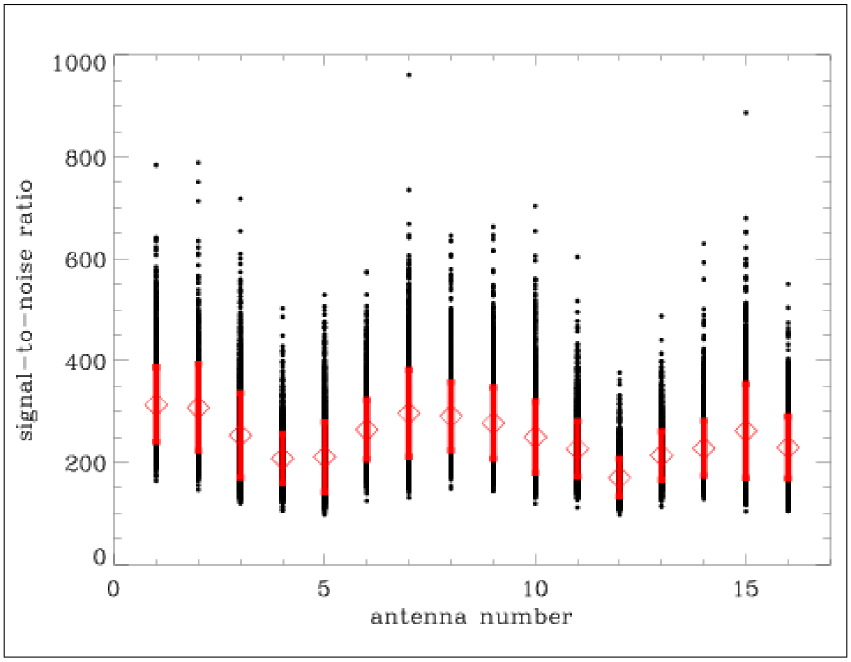

Figure 7: Same as Figure 6, but with 1-second solutions.

the solution $x = A^{-1}B$ with

$$\begin{aligned}
A_{ij} &= \textstyle\sum_{k,l} \frac{\partial C_{kl}}{\partial x_i} \cdot \frac{\partial C_{kl}}{\partial x_j} \\
B_j &= \textstyle\sum_{k,l} \frac{\partial C_{kl}}{\partial x_j} \cdot (O_{kl} - C_{kl}) \\
\frac{\partial C_{kl}}{\partial x_i} &= \delta_{il} - \delta_{ik}
\end{aligned}$$

where i and j are antenna indices from $2..n$, x_i is a solution vector of phase offsets, and O_{kl} and C_{kl} are the observed and calculated (model) phases baseline phases (φ), respectively, between antennas k and l. The double sums are over all $n(n-1)/2$ visibilities, which are used to determine the $n-1$ phase offsets. In this work, each correlator dump yielded a separate solution; but for weaker sources some integration over time may be necessary. Since all antenna weights were treated as unity, the A matrix had a simple, constant form with 15 along the diagonal elements, and -1 elsewhere. This allowed it to be inverted once. With weighting factors (relative to the reference antenna) included, it would be necessary to invert A whenever the antenna weights change, though this is still a minor computational burden.

Though the phases are determined relative to an arbitrary reference antenna, it is important then to adjust the phases to be relative to the array average. This "common mode" phase doesnt have any effect on the coherence of the ALMA phased sum, but it is significant when correlating with other VLBI antennas. The premise is that the mean of the phases of all antennas is better behaved (with respect to atmospheric and instrumental fluctuations) than a single antenna. This was born out by the observation that the rms over time of the antenna phases relative to a pool mean was lower than the rms with one antenna fixed, by amounts in the range of 10–25 %.

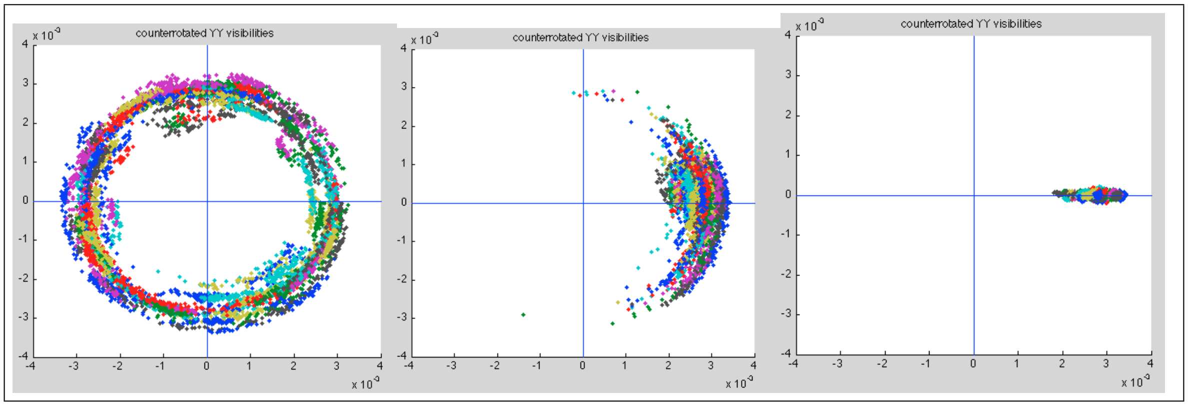

Figure 8: WVR-corrected YY fringe visibilities for all 120 baselines (colorcoded by baseline), for 63 accumulation periods. The left panel shows the complex visibilities direct from the correlator, after scaling via autocorrelation values. The central panel shows data that have been counter-rotated using antenna phases that were derived at the center of the time span. The right panel has had all baselines counter-rotated by a best-fit set of phases for each accumulation period. The coherent sum of the visibility vectors had a magnitude of 99.99% as large as the sum of the incoherent magnitudes.

Though the phases are determined relative to an arbitrary reference antenna, it is important then to adjust the phases to be relative to the array average. This "common mode"

phase doesnt have any effect on the coherence of the ALMA phased sum, but it is significant when correlating with other VLBI antennas. The premise is that the mean of the phases of all antennas is better behaved (with respect to atmospheric and instrumental fluctuations) than a single antenna. This was born out by the observation that the rms over time of the antenna phases relative to a pool mean was lower than the rms with one antenna fixed, by amounts in the range of 10-25 %.

Since voltage data from the antennas were not available, we tested the solution by counter-rotating the complex visibilities from each antenna pair. A plot of the rotated visibilities can be seen in Figure 8. Since this problem is linear in the correction phases, there was no iteration necessary in the fitter. Also, it is necessary to start with some initial guess at the phases, and to counter-rotate the data prior to the fit with the a priori values. To the least squares fitter a phase of 20° is not the same as a phase of 380°, since the mean phase is determining the fit result. The counter-rotation can be approximate, with no effect on the end result. An automated system can start with phases relative to a single antenna to start, and then track the phases over time for use as a subsequent a priori.

This model solver can now be used to fairly simply explore a number of effects, *e.g:*

- time lags in the application of the extracted phases
- poorer snr (by narrowing the bandwidth)
- smoothing of phase solutions over time and predictive filtering
- the effect of corrupted antennas

The MATLAB code is in the Haystack SVN area in the `trunk/Study/phase-study` directory.

10.3 Simple Least Squares Solution with a Slope

As mentioned in Section 6, residual delays might be an issue in forming a coherent sum if they are large. One could contemplate fitting each baseline for a residual slope, removing it, and then subjecting the relative phases of the baselines to the procedure carried out as described in Section 10.2 above.

Alternatively, one could adopt the following procedure. On each baseline average the phases in the frequency channels taken as two statistically comparable groups. (*I.e.* in the simple case of equally spaced channels with equal statistical behavior, the upper half and the lower half.) Then one carries out the least-squared fit process described above on each half, assigning zero phase to the reference antenna for each half. If one then interprets the resulting antenna phases ϕ for each frequency group in terms of a simple delay-like relation (*i.e.,* linear), then these two phases can be considered that at the average frequency, and a simple linear interpolation can provide the frequency for each channel.

Since the reference antennna has been given zero phase on each half, this effectively assigns it zero delay and moves its true delay to the the other antennas.

As an example, this procedure was implemented (in C) and is in the Haystack SVN at `trunk/Study/phase-study/lsf-games` The sample code has a number of options:

```
Usage: lsfstudy [options]
where the options are:

 -v             increases verbosity
 -c <file>      file for commentary (stdout by default)
 -n <int>       specifies number of antennas (<64)
 -m <int>       specifies solution method (<2)
 -o <float>     specifies maximum offset (degrees)
 -p             force phases to principal branch
 -r <float>     receiver phase sigma (degrees)
 -s <float>     maximum phase slope (degrees/band)
 -t <int>       specifies number of channels (<=32)

which creates and solves an n-antenna phasing problem.
The methods available are:

 0 Brute force LSF with one parameter
 1 Brute force LSF with two parameters
```

and sample output is shown in Figure 9.

```
lsfstudy -vv -c sample-m0-63.txt -n 63 -s50 -r5 -o140 -m0
lsfstudy -vv -c sample-m0-63.txt -n 63 -s50 -r5 -o140 -m1
```

The argument flags set a random distribution of slopes across the band no larger than 50 degrees, assume the per-channel phase noise is (gaussian with sigma of) 5 degrees, and spreads the frequency offsets of the antennas across 140 degrees. The two invocations (`-m0` and `-m1'` perform the two fitting procedures, a single fit, and a pair of fits, respectively. These parameters are hopefully rather poorer than typical ALMA performance, and are mostly for illustrative purposes. However, in the two cases, the sigmas of the residuals were:

$$\begin{array}{rrrrrrr} \text{one fit}: & \sigma_{\mathrm{m}in} = & 0.908371 & \sigma_{\mathrm{ave}} = & 13.121825 & \sigma_{\mathrm{m}ax} = & 26.436438 \\ \text{two fits}: & \sigma_{\mathrm{m}in} = & 0.255233 & \sigma_{\mathrm{ave}} = & 1.767526 & \sigma_{\mathrm{m}ax} = & 4.684120 \end{array}$$

the key point being that in no case was the residual of the double fit worse than the per-channel phase noise.

Obviously, further testing is suggested.

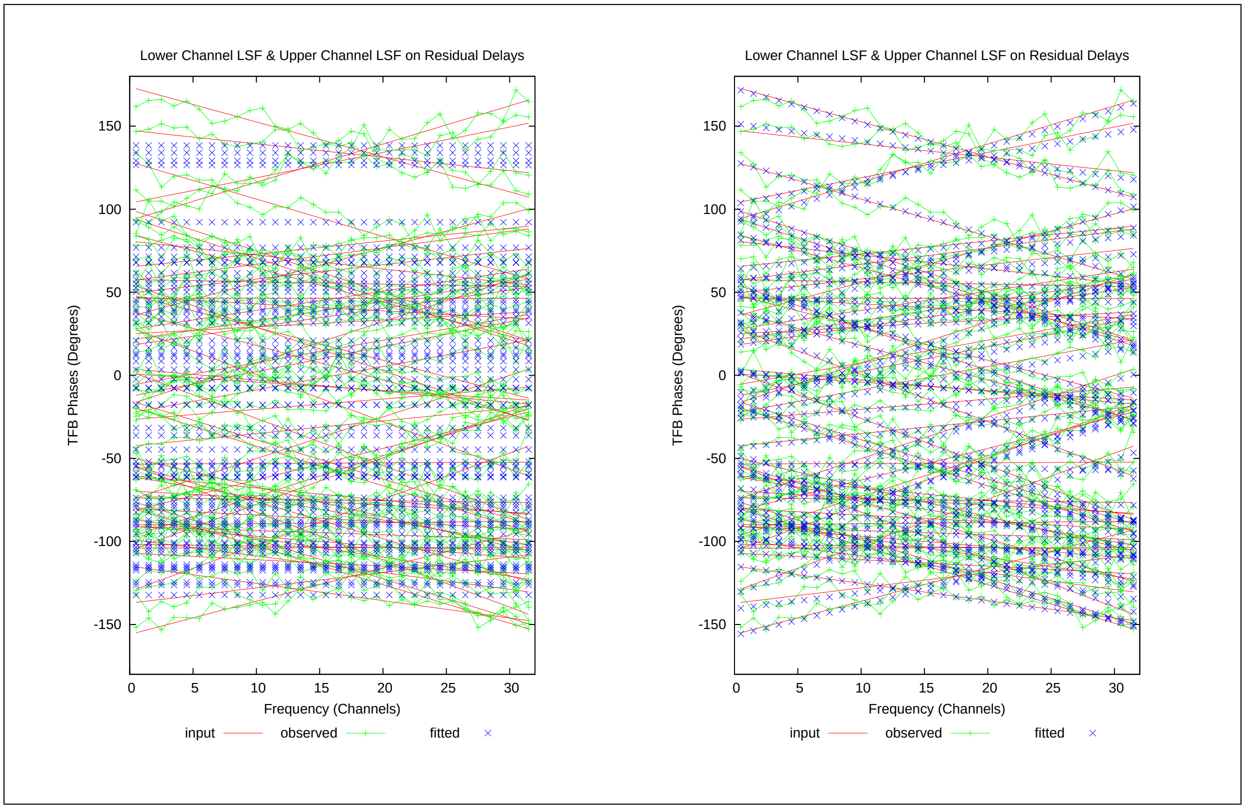

Figure 9: Simulation of 63 antennas with *rather large* residual slopes. The left-hand panel shows the result with a single least-squared fit process carried out on the average over all channels for a single antenna. The right-hand panel shows the result of a the fit process carried out twice: once each on the average of the upper frequency channel, and again on the lower frequency channels. Those fits were then interpreted for simple linear delay slopes at each antenna.

10.4 More Algebra

Transcribing my algebraic notes into this space, the minimization of χ^2 as defined in Section 5.1 with respect to the reference antennas is:

$$\begin{aligned}\chi^2 &= \textstyle\sum_{i\neq j} \frac{1}{\sigma_{ij}^2}[\varphi_{ij} - (\psi_i - \psi_j)]^2 \\ \frac{\partial\chi^2}{\partial\psi_k} &= \textstyle\sum_{i\neq j} \frac{2}{\sigma_{ij}^2}[\varphi_{ij} - (\psi_i - \psi_j)](\delta_{jk} - \delta_{ik})\end{aligned}$$

where i and j range from $0..N-1$ and k ranges from $1..N-1$ since we are setting ψ_k identically zero. (A singular matrix results if we try to use the N^{th} equation.) (I do this for eventual use when the baseline sigmas are found not to be equal.)

The linear and constant parts (in ψ) are transformed into the form $(Ax = B)$ used in the preceding sections as follows. First, for the linear part:

$$\begin{aligned}
\textstyle\sum_{i\neq j} \frac{2}{\sigma_{ij}^2}(\delta_{jk}-\delta_{ik})(\psi_i-\psi_j) &= \textstyle\sum_{i\neq j} \frac{2}{\sigma_{ij}^2}(\delta_{jk}\psi_i-\delta_{ik}\psi_i) - \sum_{i\neq j} \frac{2}{\sigma_{ij}^2}(\delta_{jk}\psi_j-\delta_{ik}\psi_j) \\
&= \textstyle\sum_{l\neq j} \frac{2}{\sigma_{lj}^2}(\delta_{jk}\psi_l-\delta_{lk}\psi_l) - \sum_{i\neq l} \frac{2}{\sigma_{il}^2}(\delta_{lk}\psi_l-\delta_{ik}\psi_l) \\
&= \textstyle\sum_{l\neq j} \frac{2}{\sigma_{lj}^2}(\delta_{jk}\psi_l-\delta_{lk}\psi_l) - \sum_{j\neq l} \frac{2}{\sigma_{jl}^2}(\delta_{lk}\psi_l-\delta_{jk}\psi_l) \\
&= \textstyle\sum_{l\neq j} \frac{4}{\sigma_{jl}^2}(\delta_{jk}-\delta_{lk})\psi_l \\
&\equiv A_{kl}\psi_l
\end{aligned}$$

where we've first substituted l for i in the first term and l for j in the second term and then relabelled i as j in the second term (and noting that $\sigma_{lj}^2 = \sigma_{jl}^2$ and likewise for the delta functions). The elements of the A matrix correspond to the sum over j for the lk^{th} element.

For the B term, as it is convenient to store only half of the antisymmetric matrix φ_{ij} (*i.e.* $i < j$), we write:

$$\begin{aligned}
B_k &\equiv \textstyle\sum_{i\neq j} \frac{2}{\sigma_{ij}^2}(\delta_{jk}-\delta_{ik})\varphi_{ij} \\
&= \textstyle\sum_{i<k,k<i} \frac{2}{\sigma_{ik}^2}\varphi_{ik} - \sum_{j<k,k<j} \frac{2}{\sigma_{kj}^2}\varphi_{kj} \\
&= \textstyle\sum_{i<k,k<i} \frac{4}{\sigma_{ik}^2}\varphi_{ik}, \mathrm{or} \\
&= \textstyle\sum_{i<k} \frac{4}{\sigma_{ik}^2}\varphi_{ik} - \sum_{k<i} \frac{4}{\sigma_{ik}^2}\varphi_{ki}
\end{aligned}$$

which makes it convient for code that will access φ_{ij} via a sequential index b (*i.e.,* as in the following pseudo-code scrap):

```
for (i = 0, b = 0; i < N; i++)
    for (j = i+1; j < N; j++, b++)
        ... phi[b] ...
```

For the case of equal σ_{ij} the factor in all the sums reduces to $4/\sigma^2$ which may be dropped everywhere to obtain the form of the A matrix and B vector described in Section 10.2 above.