Effects of Residual Delays on the Phasing Loop

Geoff Crew

May 13, 2013

1 Preamble

As mentioned in the ALMA Phasing Project white paper, the ALMA system uses a `DelayServer` to calculate the geometric delays for the array antennas and then propagate these to the agents in the system which manage the data for the correlation. For each antenna, the delay is essentially decomposed into three parts: some integral number of 250-ps samples, a fractional part (250/16 ps), and a residual. The first two pieces are applied in the hardware; and the the CDP nodes can correct the data for the residual portion on its way to the archive. The phasing system benefits from the first two adjustments, but sum of antenna signals is calculated before the data leaves the correlator, so the residual portion is unavoidably included in the summed signal.

As pointed out recently by R. Hills[1], the discrete nature of the adjustments in the hardware potentially pose a challenge to the phasing system. For very short baselines, they are probably ignorable. For the longest baselines, these corrections are made several times per TE. Since this is significantly faster than even the fastest phasing loop contemplated, this is ultimately a noise term about which the phasing project can do nothing.

As R. Hills points out, there are two consequences of this: there is a loss of sensitivity as the summed signal is less coherent than it otherwise could be. In addition, because the delays change dynamically, the latency inherent in the phasing loop will always be responding with corrections appropriate to an earlier time, and in some cases, these corrections may make things worse. This note is written to look into these issues in somewhat more detail.

2 Residual Delay Examples

In order to quantitatively address these (and related issues), a (rather simple) ALMA delay simulator was written. (Some usage notes appear in the Appendix A.) It includes the ALMA pad positions and several canned position lists and allows one to also generate random antenna dispositions based on min/max distance from the center and then calculate a number of things.

[1]"Residual Delay Errors—Details of Effects and Implications for Phasing-up", private communication, March 2013

Our present interest here is to explore the issues raised by R. Hills.

We begin with the case of a single antenna (A081, 239 m from the delay center) as shown in Figure 1. In the plot and the ones that follow present the summed signal from a set of

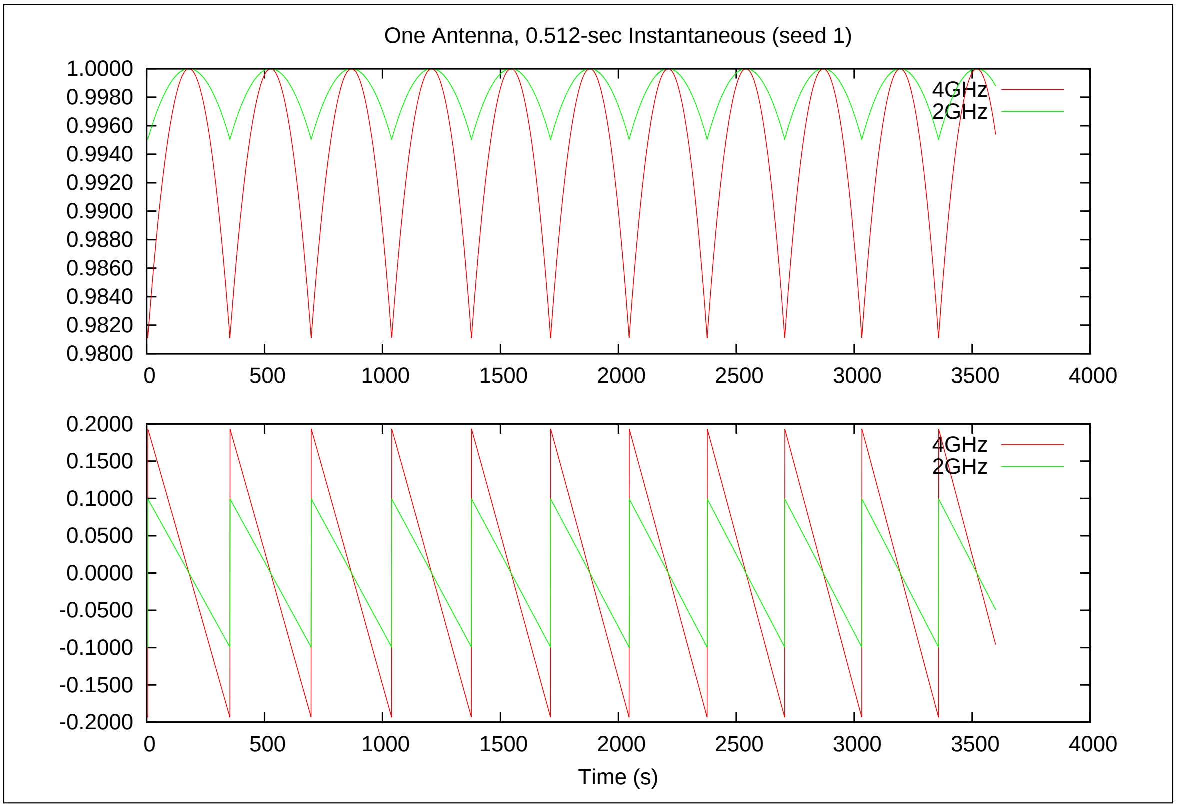


Figure 1: Single antenna (A081, 239 m) real and imaginary signal contributions resulting from delay residual, instantaneous values.

antennas (in this case just one) is shown as a function of time. The "observation" is of a target at the vernal equinox for an hour near zenith. (This is yet-another simulation option which is at present, not exercised.) The top panel and bottom panels, respectively are the cosine (real) and sine (imaginary) parts of the signal. The two lines plotted (red and green) are for the two tunable filter banks (TFBs) at the ends of the 2 GHz IF band (4 and 2 GHz, respectively). (The other 30 lie in between.) The phase swings (obvious in the lower panel) between $\pm$ 11.25 and $\pm$ 22.5 degrees are rather obvious every 400 s as the digitizer clock is shifted by 1/16$^{\text{th}}$ of a sample. From the top panel, we see that this antenna's contribution to the sum is degraded by no more than 0.4% at 2 GHz and 1.8% at 4 GHz.

Figure 1 plots instantaneous values every 512 ms (the dump time for Mode 13, providing dual-polarization, full bandwidth correlations). In reality, of course, the correlator averages signals over this window (Appendix B), which produces a further (minor) degradation as shown in Figure 2. Numerically, the values are very similar; hereafter we use the integrated values. Two more antennas are shown in Figures 3 and 4.

The phasing loop will be making calculations over longer samples than the dump time. In

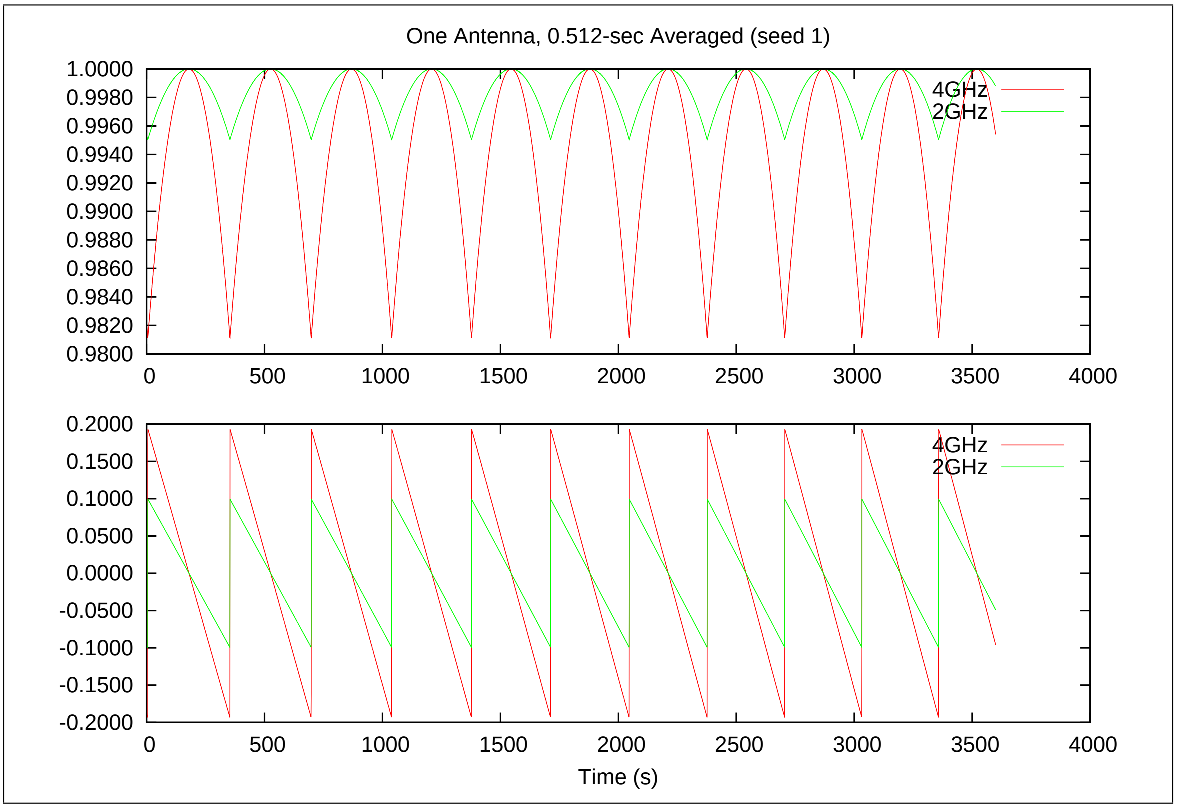


Figure 2: Single antenna (A081, 239 m) real and imaginary signal contributions resulting from delay residual, integrated over 512 ms dump time.

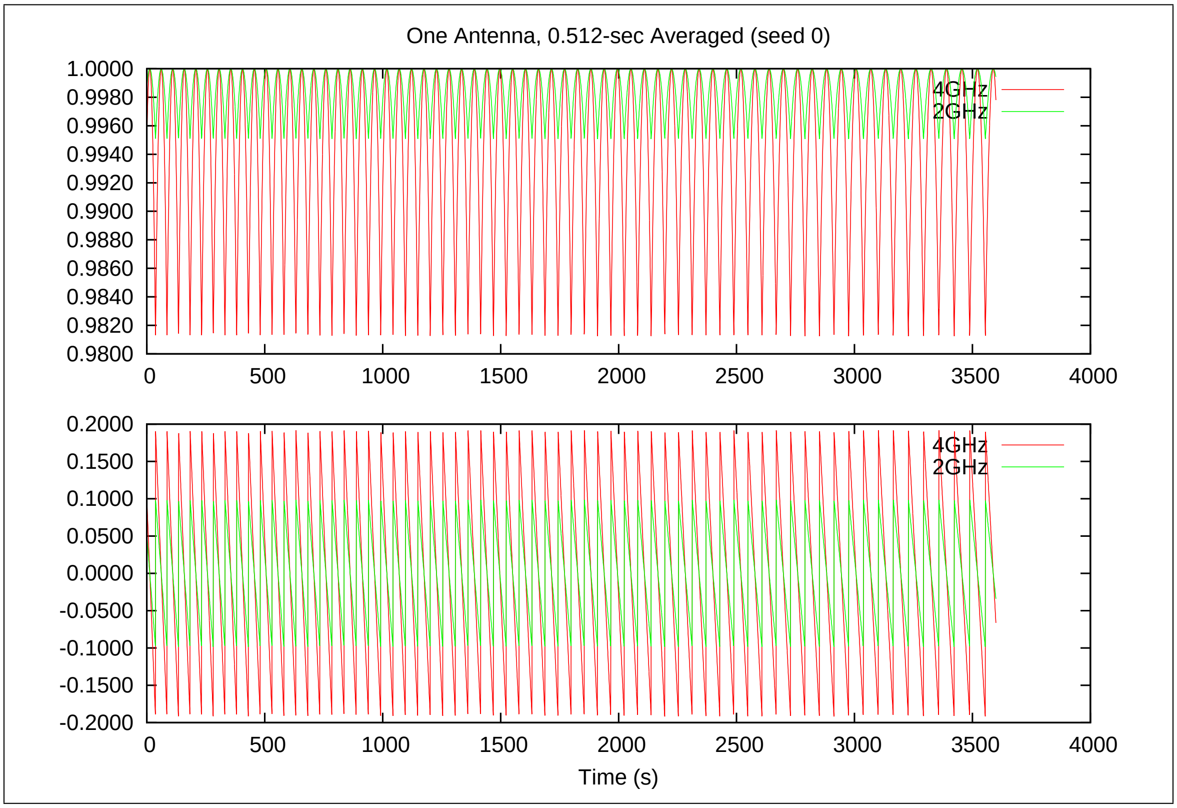


Figure 3: Single antenna (W210, 3730 m) real and imaginary signal contributions resulting from a relatively fast delay residual, integrated over 512 ms dump time.

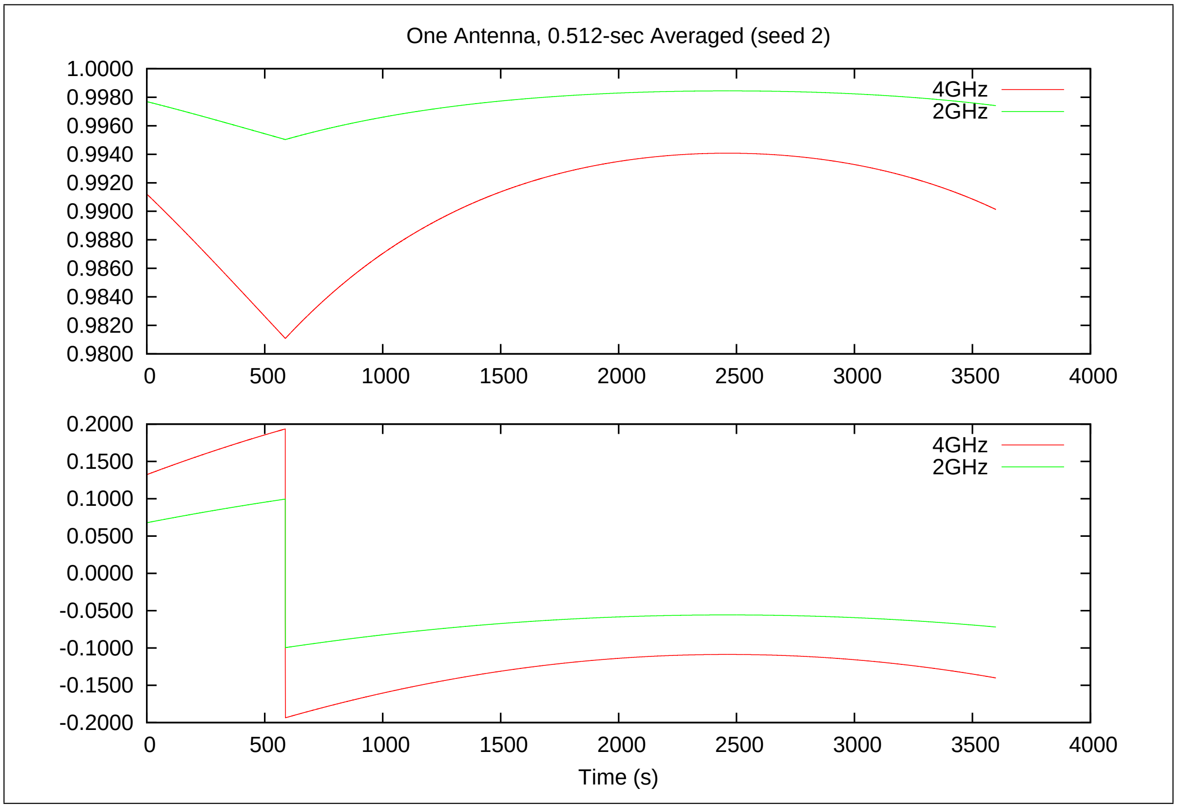


Figure 4: Single antenna (A084, 295m) real and imaginary signal contributions resulting from a relatively slow delay residual, integrated over 512 ms dump time.

Figure 5 we show the just the real part for one through 32 times the dump period. Clearly at

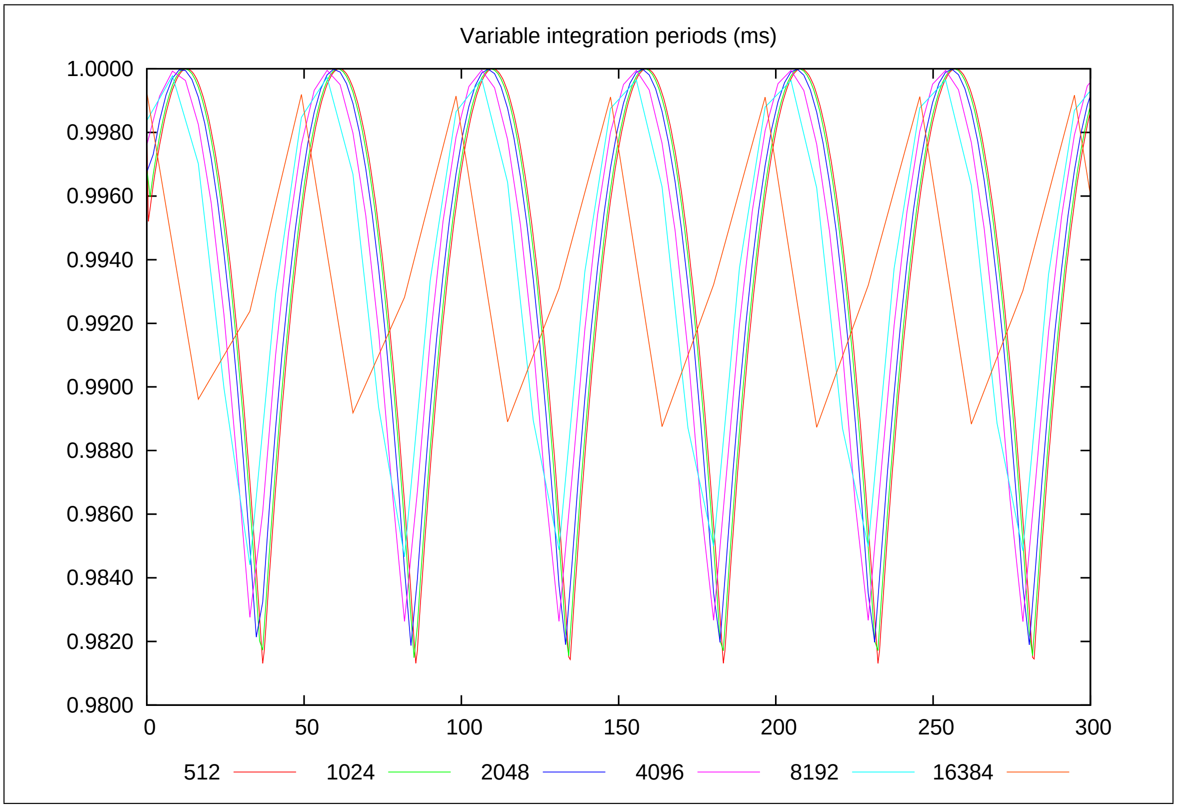


Figure 5: The single antenna amplitude has been integrated over multiple dumps. Note that this is just the first 5 minutes of data—there is little loss of fidelity until 16-s integrations. All curves are for the TFB at 4 GHz in the IF.

16-s integrations for a fast delay rate, we are not tracking the extremes of signal degradation. Hereafter we shall work with 4-s integrations, and where only one TFB signal is shown, we present the one with the worst excursions—*i.e.* the one at 4 GHz in the IF. The phasing system can of course use longer integrations than 4 s, but the point is this simulator is not of sufficiently high fidelity to track the degredation in that case. And we are mostly looking to verify that the effect of the delay rate residual is not toxic to our goals. The full “observation” for this signal is in shown in Figure 6 for completeness.

Obviously the case becomes more interesting with more than one antenna. With three antennas, shown in Figure 7 the contributions of the several antennas hit their extrema at various times. The amplitude is more chaotic, but not worse. Several other antenna triads (to show the variety) are seen in Figure 8. The red curve on the bottom (marked “seed:0” in the figure) is the same one as in Figure 7. As the number of antennas increases (shown in Figure 9) the signal gets more chaotic as the timing of the phase discontinuities is essentially random. As with dice, a dozen is enough for the character to shift from “a few” with some organization to “many” with none. The variance of the signal is reduced as well, since relatively more antennas are to be found near the target delay minium rather than the

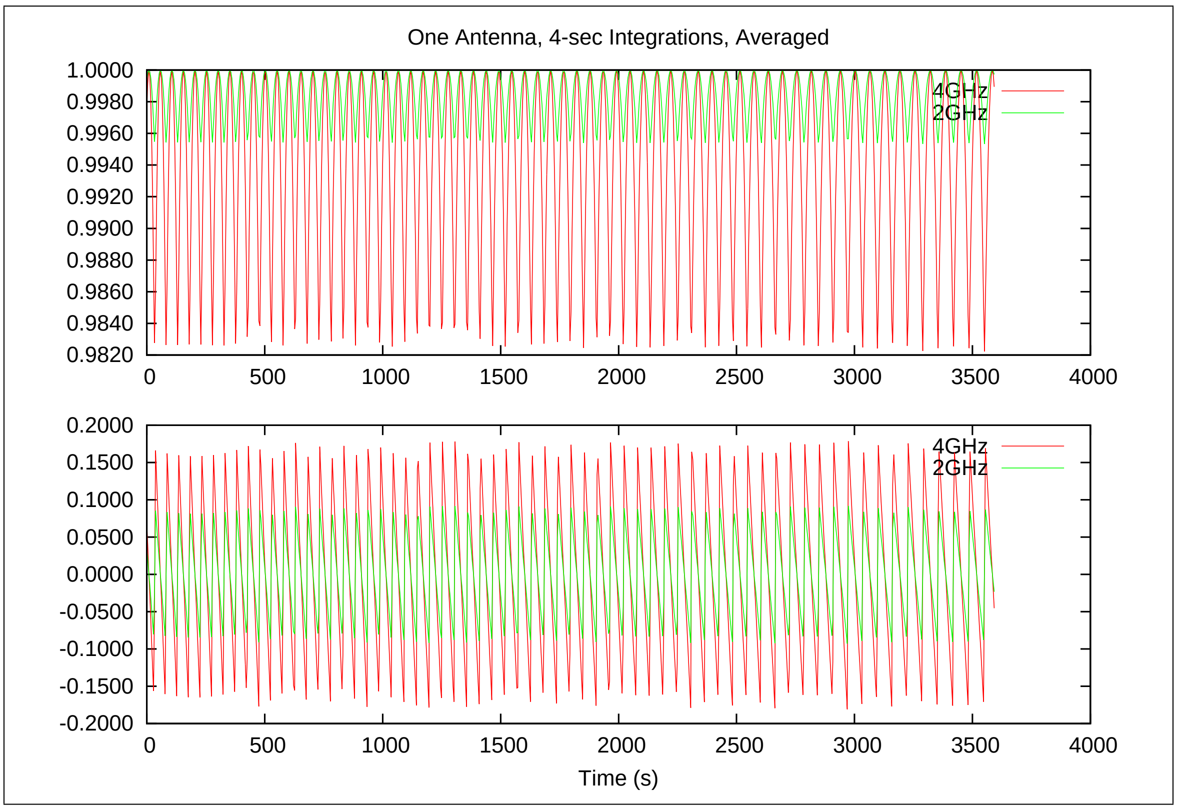


Figure 6: The single antenna amplitude signal integrated over 8 dumps (4 seconds).

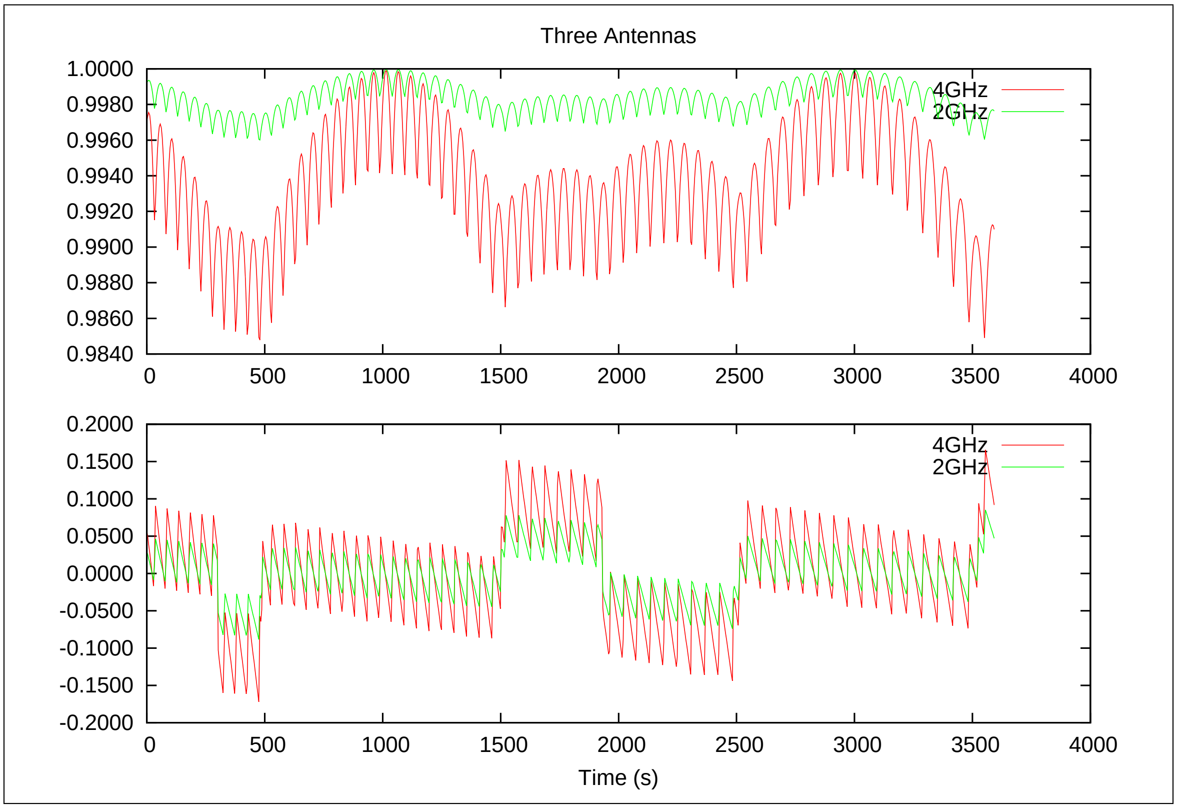


Figure 7: The co-added signal from three antennas, is presented. In the sine part (lower panel) the discrete delay jumps are obvious for each antenna.

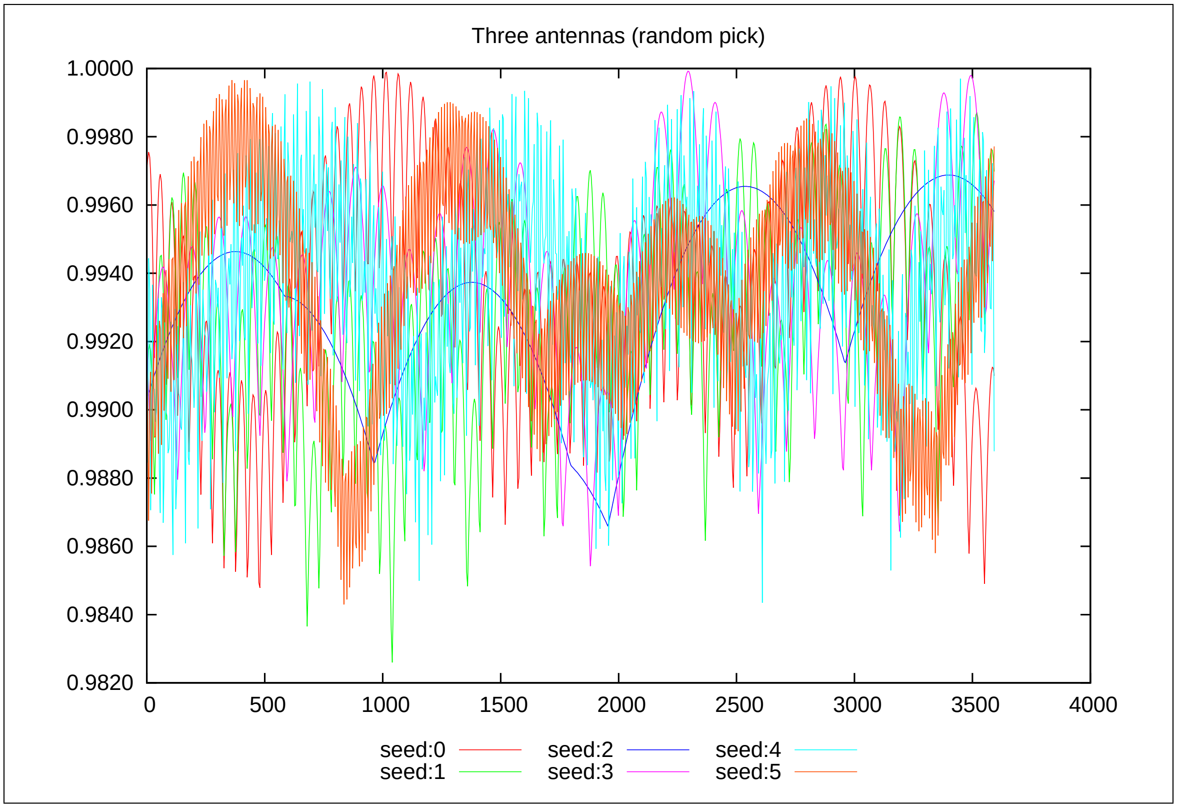


Figure 8: Same as the real part of Figure 7, for several antenna triads.

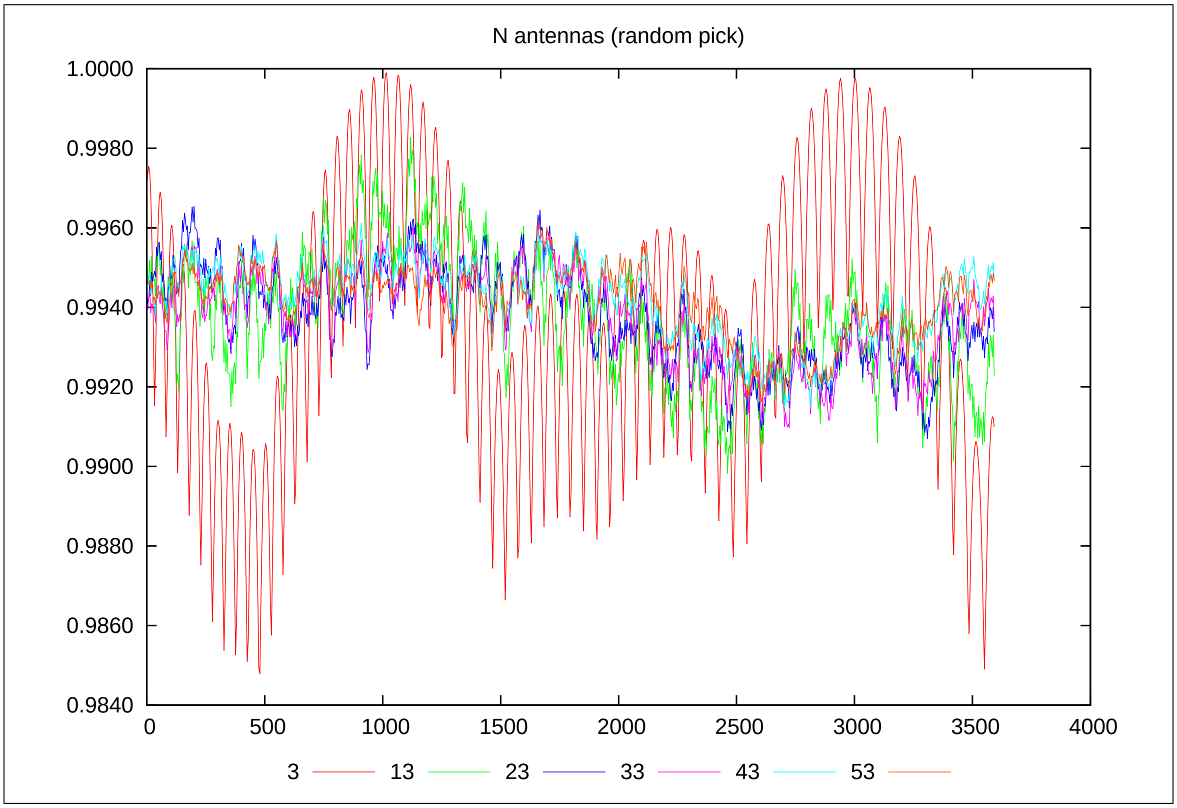


Figure 9: As the number of antennas increases from 3 to 53, the signal becomes rapidly chaotic and less dispersed.

extremes. The distribution of antennas is also a factor. In Figure 10 we show the signal for

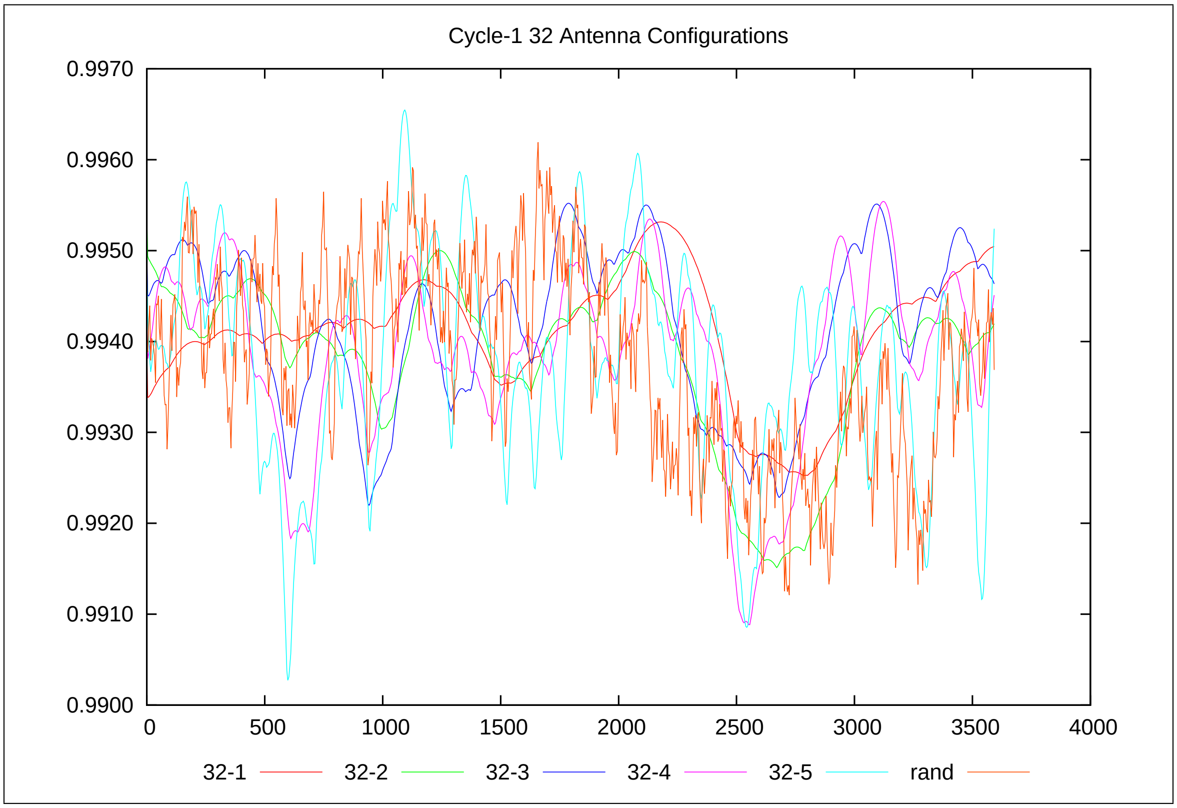


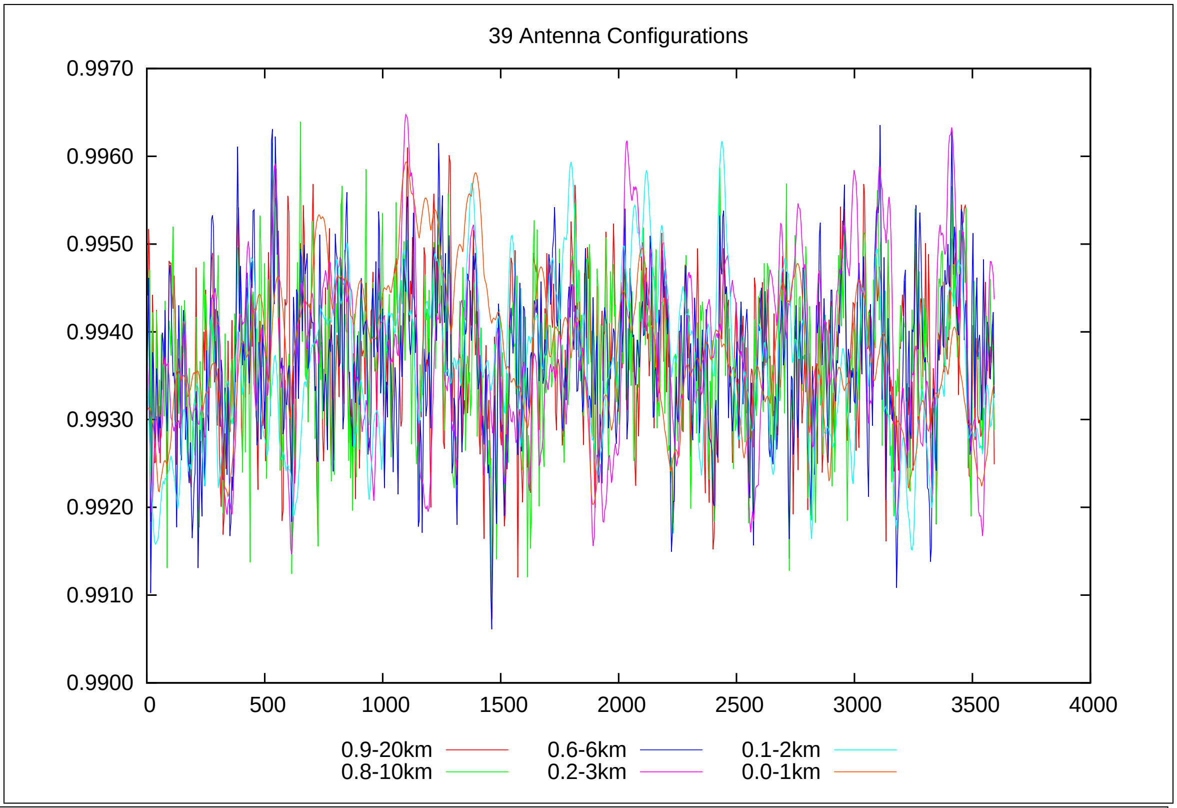
Figure 10: The signal for the 5 Cycle-1 configurations are shown and compared to a random choice of 32 pads for the antennas.

6 collections of 32 antennas—the 5 Cycle-1 configurations together with a random choice of 32 pads. Notice how 32-1 and 32-2 vary smoothly—these antennas all have a similar delay, and so more antennas are working together.

I do not have configurations for more than 32 antennas; however it is clear we can't do worse than a random choice. In Figure 11

we show results for a random choice of 39 and 50 pads restricted in various sized-annuli. The main point of the figure is that there does not seem to be significant differences between the various cases.

I believe the take-home message here is that there is a general reduction in the phased signal, but that it is relatively small (0.5%) so we can probably live with it. It introduces amplitude variations at a smaller level that perhaps could be estimated with a variant of this tool (or equivalent based on existing observational data).



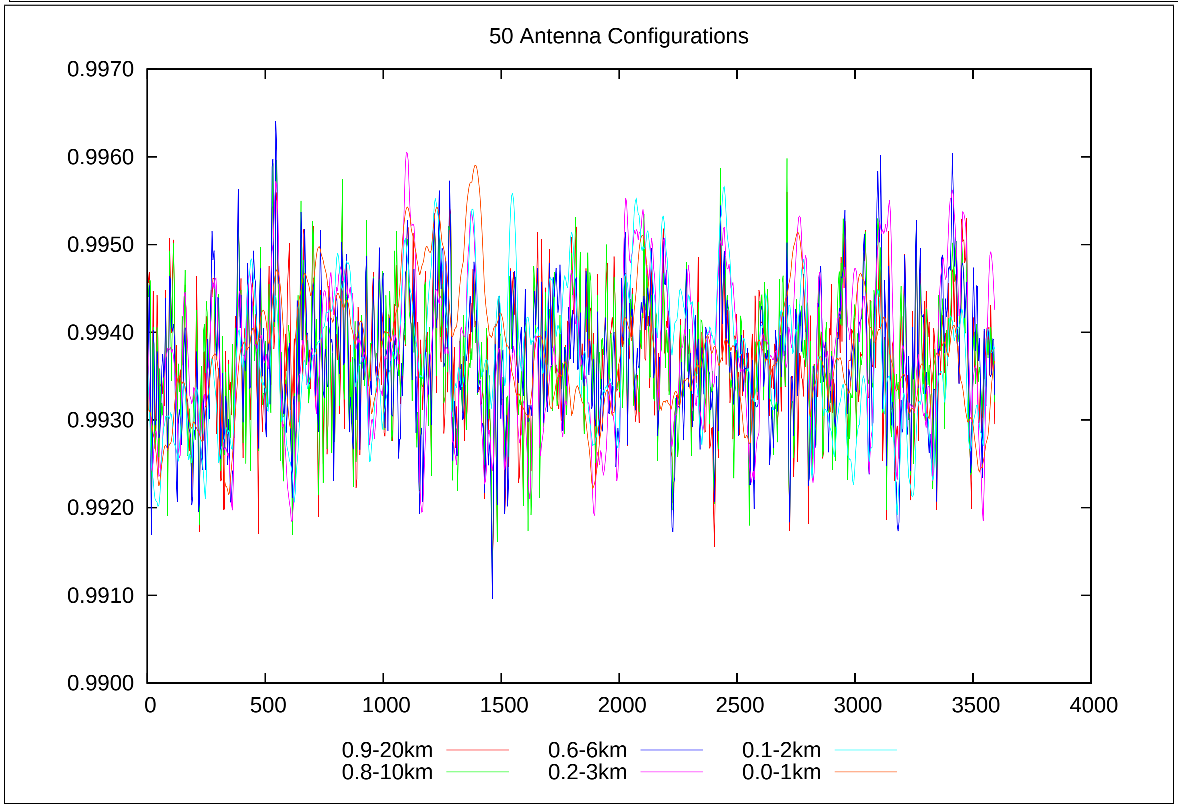


Figure 11: Some 39 and 50-antenna cases.

3 WVR and Fast Loop

As indicated many times, the proposed APP solution includes a phasing loop operating on two timescales ("fast" and "slow") which operates in the context of the residual delays discussed in the preceding Section 2. On the fast loop, the CDP nodes have access to recent WVR data from each antenna and can apply per-antenna phase corrections.

`CORR/CDP/Node/SpectralDataProcessor/src/SpectralDataProcessor.cpp`, in which the current correction is made (look near line 183), includes a `pathLengthDiff` between the two antennas on the baseline. The required correction is either an absolute one (*i.e.* a difference relative to some previous value), or a relative one (*i.e.* relative to the reference antenna). In the test code, it is easiest to handle the absolute case, and make the corrections relative the start of the scan.

A sample delay history for one antenna is shown in Figure 12 assuming a band 6 sky frequency (230 GHz); *i.e.* well less than a 1.3 mm wavelength. Of course, the WVR data is

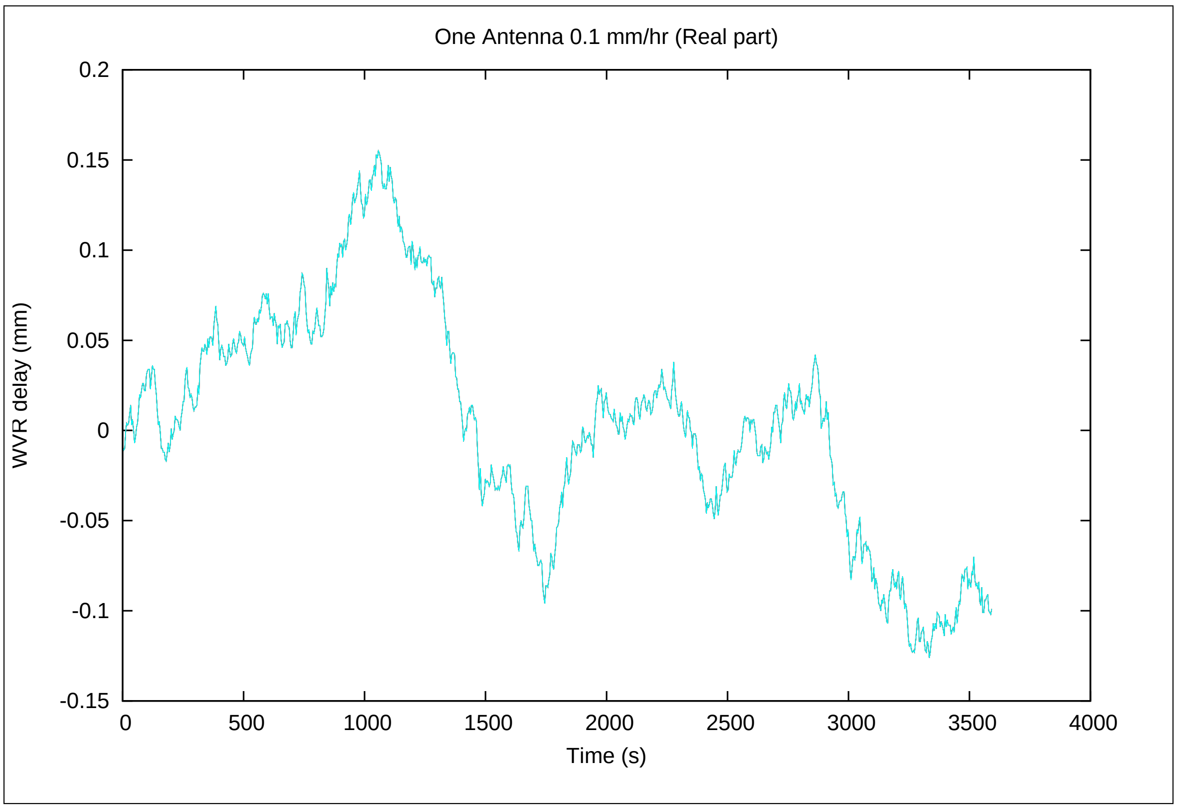


Figure 12: Absolute WVR path delay for one antenna. The absolute path (in mm) for one antenna is shown as a function of time.

stale by the time the CDP nodes get it, and the correction that is made will be applied still later. The precise timing is not known at this time; to survey the possibilities, we consider that the dump time is 0.512 s, and that the WVR data used for the correction is some number of samples late (keeping it even for an easy conversion to seconds.) Figure 13 shows five possibilites for delays of 2, 6, 10, 14 and 18 samples (preceded by an "f" in the figure key), compared with no correction whatsoever ("f0" in the plot). For this example, the sawtooth in the phase (lower pair of plots) is completely obliterated without the WVR correction. The cyan curve, representing the fastest correction (about 1 second), mostly recovers what we would expect based on the delay residuals. A subtlety to note in the red curve (uncorrected) is that reaching the flipping point for the delay setting sometimes produces a better result than previously, and sometimes much worse. (*I.e.* the large excursions of the red curve at about 900, 2100 and 3200 seconds are due to the shifting of the delay—the sawtooth edge).

The relative (to the reference antenna case) is not implemented at this time.

Adding more antennas produces about what you'd expect. For the 32-antenna cycle-1 configuration (32-1) we show (similar conditions) the WVR data in Figure 14 Note that noise at each antenna is uncorrelated with its peers. In reality, there would be some common-mode atmospheric fluctuations together with a random part. The sum signal is shown in Figure 15.

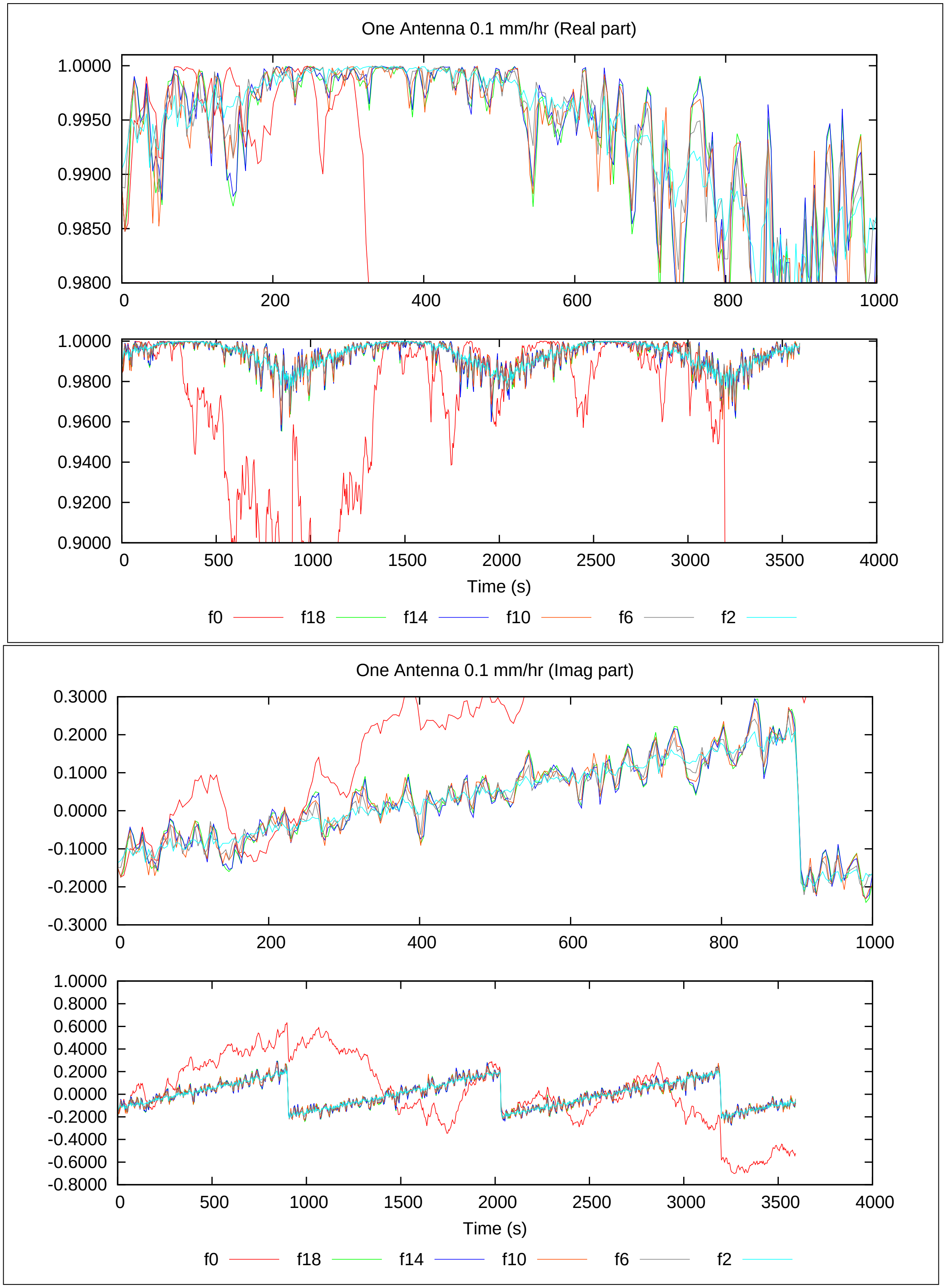


Figure 13: Abolute correction of one antenna with WVR data. The top panel in each pair is an expanded scale version of the lower panel. The top pair is the real part (*i.e.* essentially the amplitude), the lower pair is of the imaginary part (*i.e.* essentially the phase). The different curves provide different delays as explained in the text.

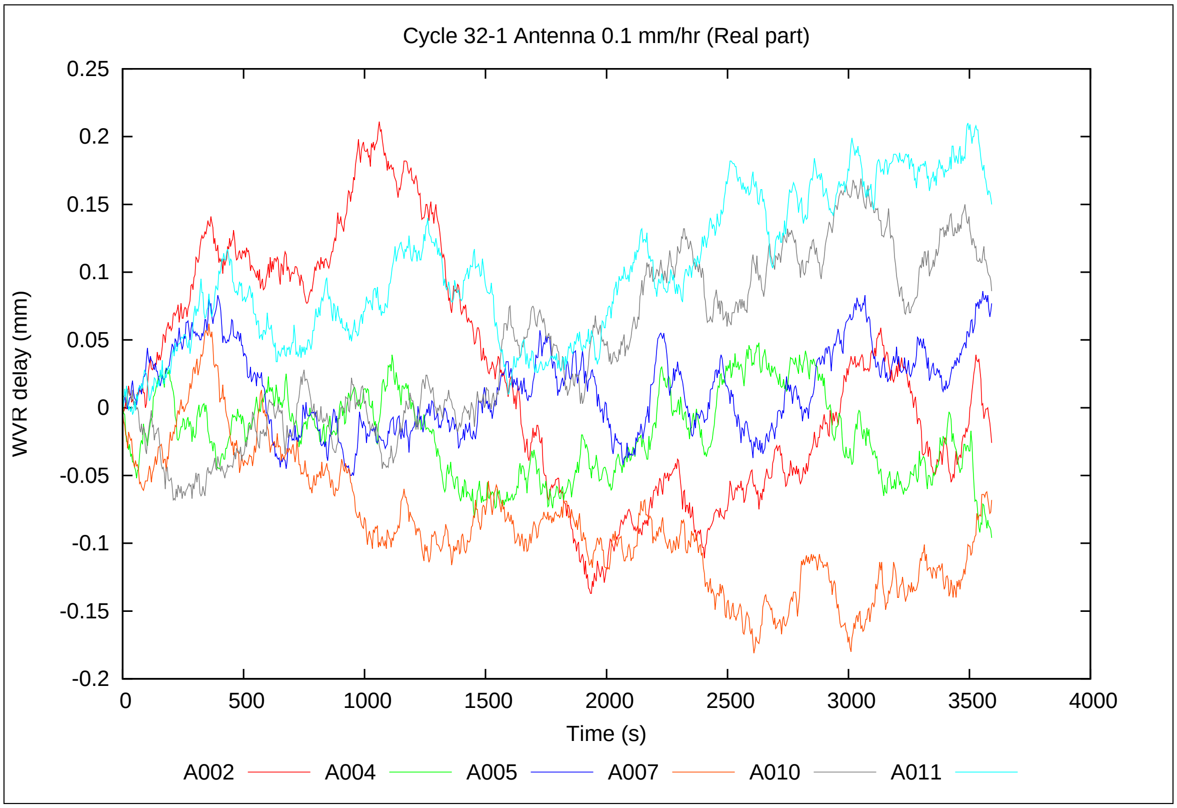


Figure 14: Absolute WVR path delay for six antenna of the 32-1 configuration. The absolute path (in mm) for one antenna is shown as a function of time.

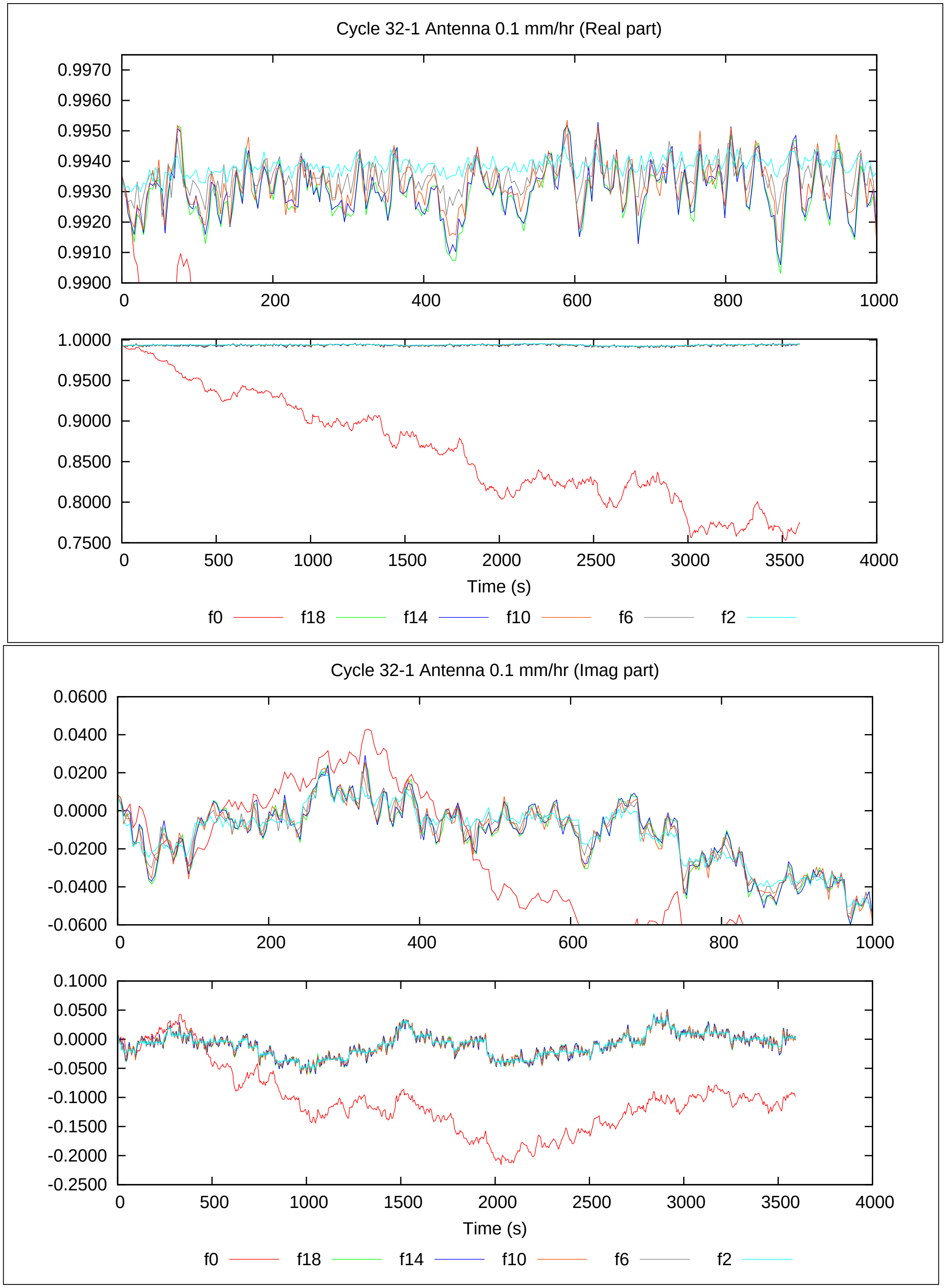


Figure 15: Summed signal for the 32-1 configuration with only a fast, WVR-correcting loop. Same format as Figure 13.

4 Slow Loop

The slow loop is the one operating in TelCal which takes the baseline phase measurements and solves for per-antenna phase values. These can then be applied to as corrections to rotate the individual antenna signals into a coherent sum. It was straightforward to incorporate into the simulator the simple least-squares fitter code that was assembled for the phasing white paper. That treatment considered two ways to solve for the individual antenna phases, and the choice of which to use depends on what one considers is the problem to be corrected.

One way is to average the channels together to produce and average phase error per antenna. This is appropriate if one assumes the source of the errors does not have a strong frequency dependence. On the other hand, with uncorrected sources of delay, one can have a strong frequency dependence in the phases. In this case one might solve for a phase and a slope for each antenna (two parameters). This is equivalent to solving for a phase for the average of the upper channels and also for a phase for the average of the lower channels (assuming a linear dependence with frequency).

The simulation code is controlled with a slow phase loop parameter `slow` set to `0` (for no solution/correction), to `1` (for one solution/correction per antenna) or to `2` (for two solutions/corrections per antenna). As a demonstration, we consider adding a (not necessarily) physical (random) steady phase drift to each of the antennas. In Figure 16 we show an example with seven antennas. The top group of three panels shows the real part of the sum signal as a function of time. These three panels differ only in the axis ranges (zooming-in for the top panel). The lower group shows the imaginary (phase-like) part. For each setting of the `slow` parameter (labelled `s0`, `s1`, and `s2` in the figure), the signal for the extremes of the 2-GHz band are shown, *i.e.* for 2 GHz and 4 GHz.

As can be seen in Figure 16, without the slow loop, the coherent sum (red curves) departs significantly from unity; it doesn't drop all the way to zero as with seven antennas, it's hard to have them all exactly out of phase with each other. The ridges on that signal show the comparatively minor glitches from the active adjustments to the delay solution. The middle panels of each trio zoom in on the vertical scale so that it is apparent that the fit with 2 parameters (blue curves, not surprisingly) does a much better job than with just one (green curves), although those are not horrible. There are however, glitches in the process which occur at the delay solution adjustments. *I.e.* when the delay solution shifts by 1/16 turn this introduces noise into the fits which shows up as spikes.

Shorter integrations help with this, as relative amount of time "glitched" is shorter. On the other hand, longer integrations are more likely given some of the (unavoidable) latencies in the ALMA system. The preceding plots were made assuming a 4-sec integration period. In Figure 17 we show comparisons to Figure 16 for the real part of the phase sum for 32-sec and 1-sec integrations. Not surprisingly the 1-sec loop does better. What is also noticeable is that for the 32-second integration time case, the benefits of solving for a phase-slope are beginning to be lost. (*I.e.* the `s1` and `s2` curves are not noticeably distinguished.)

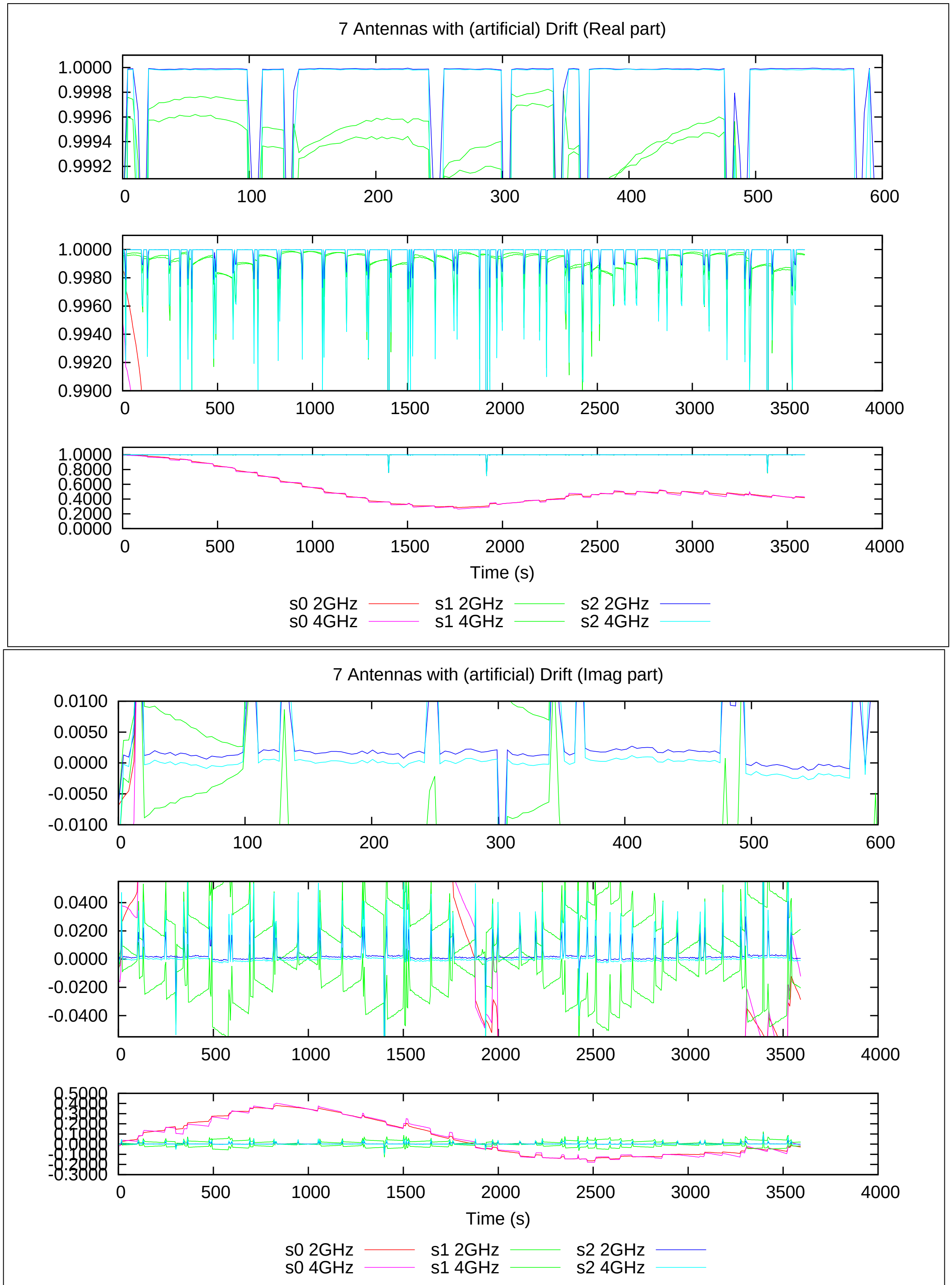


Figure 16: The performance of the slow loop with a per-antenna random drift is shown for a seven-antenna case. The top panel group show the real part of the phase sum, and the lower panel group show the imaginary part. See text for detailed explanation.

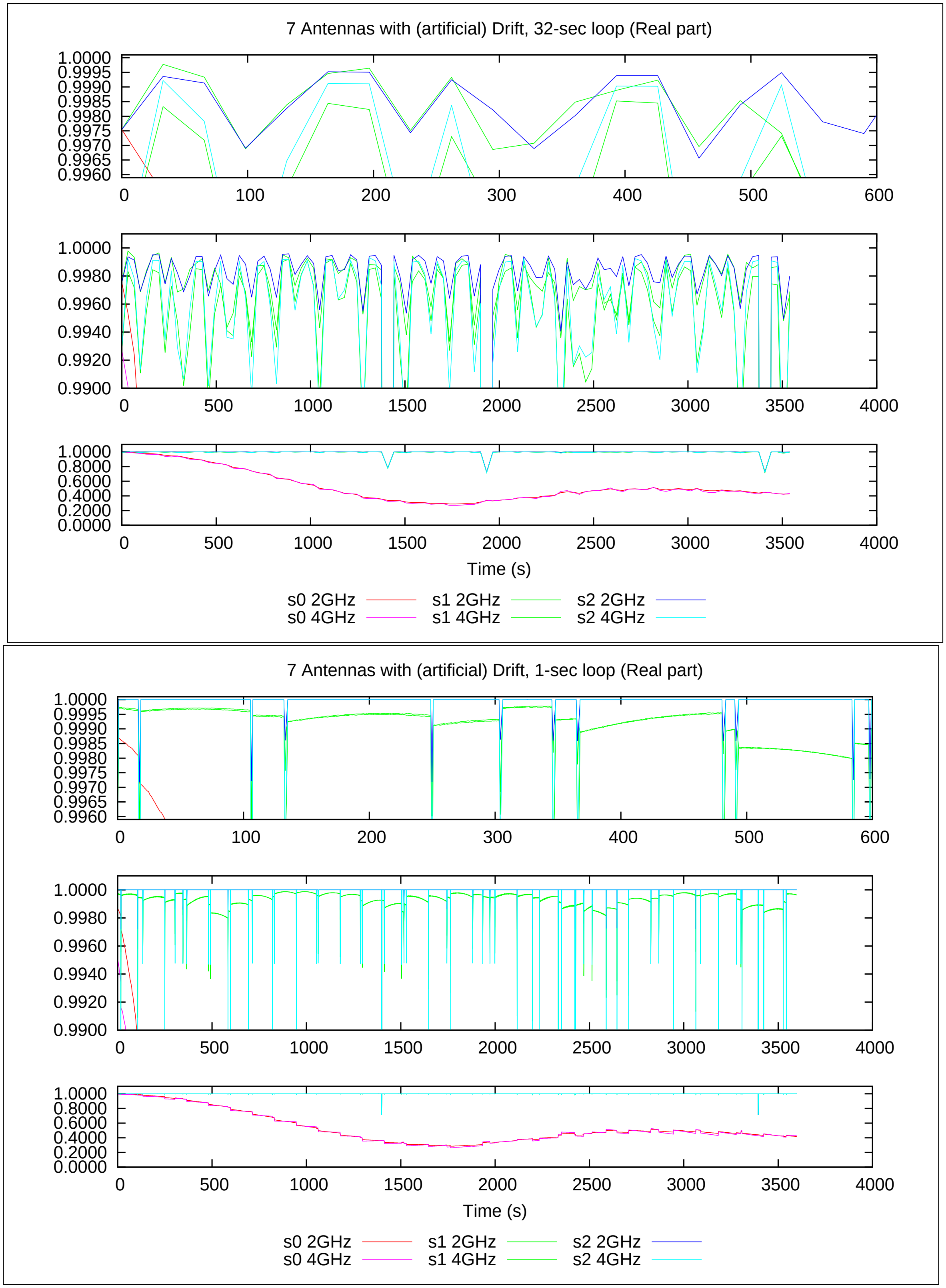


Figure 17: The performance of the slow loop with a per-antenna random drift is shown for a seven-antenna case for 32-sec and 1-sec integrations. All panels are for the real part of the phase sum. See text for detailed explanation

5 Complete Phasing Loop

Putting together the machines of the fast (WVR, Section 3) and slow (Fit, Section 4) loops, we have a complete simulation of what we propose to insert into ALMA.

We begin with a set of survey examples covering a 7-antenna configuration with all antennas within 600 m of a central point. This makes it possible to see some of what is happening. The four examples that follow consider increasingly more water vapor (path lengths of approximately 0.01, 0.1, 1 and 3 mm/hr of path-length growth). The same random sequences were followed for all cases—just the overall path length scaling, as shown in Figure 18.

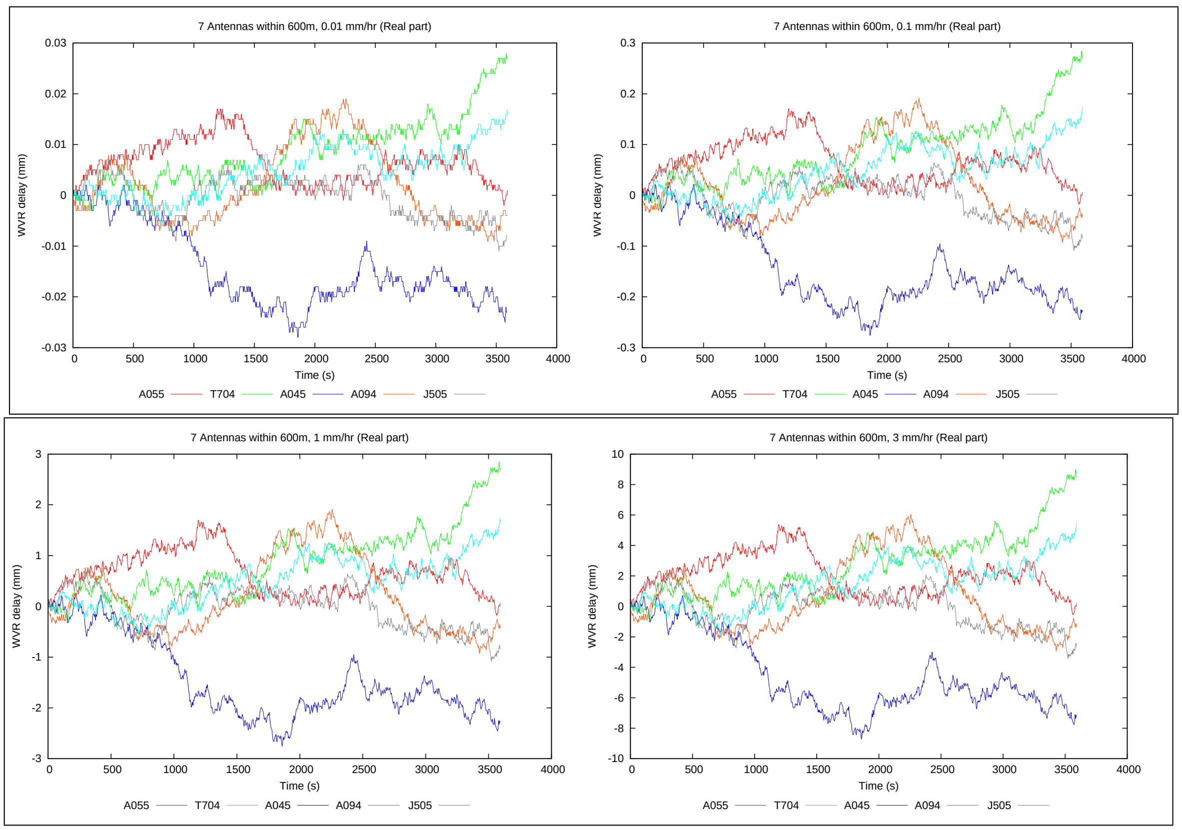
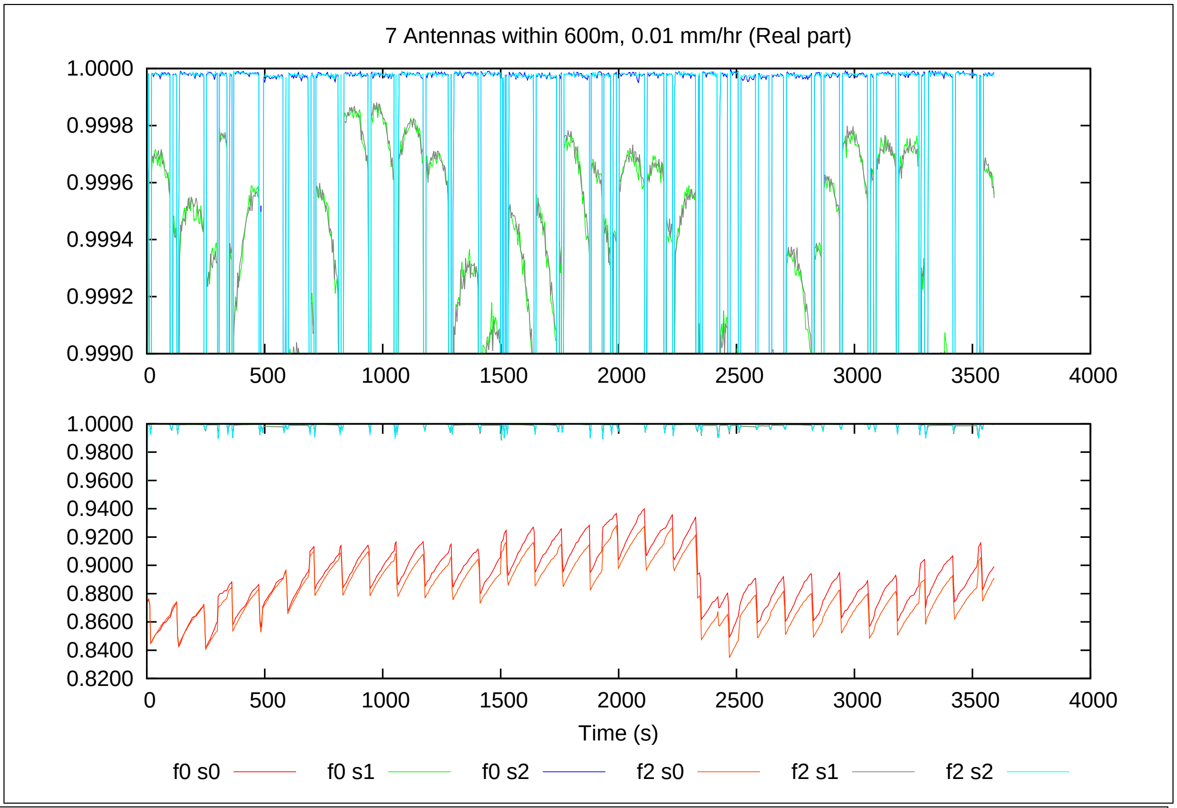


Figure 18: Per-antenna path delays for the cases that follow.

The corresponding real/imaginary parts of the phased sum are shown in Figures 19, 20, 21 and 22.



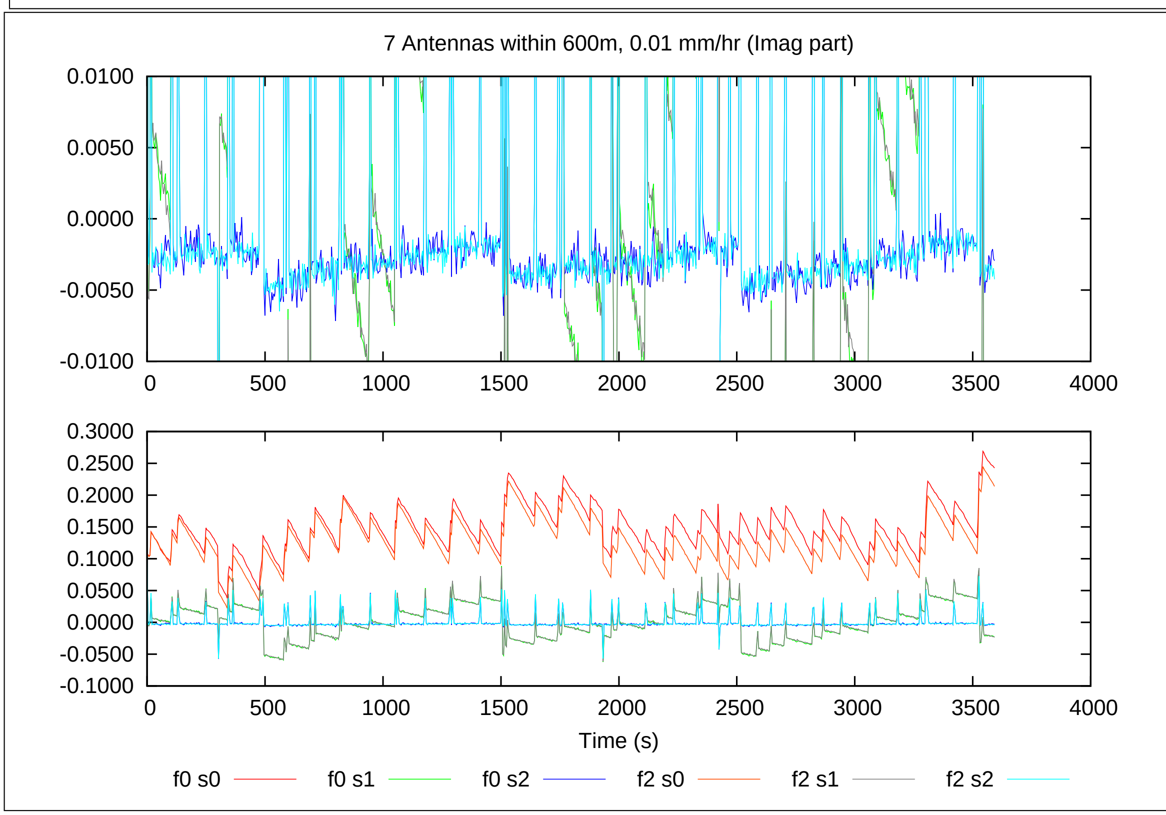
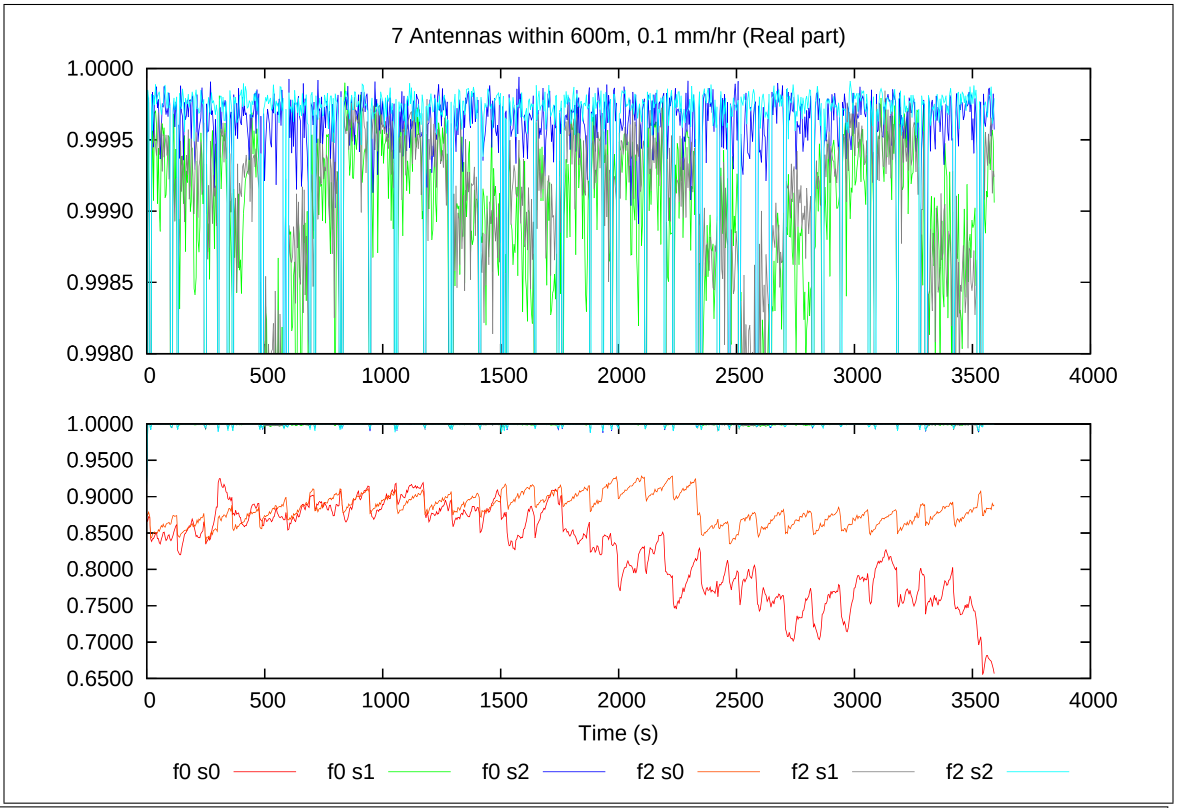


Figure 19: A 7-antenna phasing loop sample with almost no moisture.



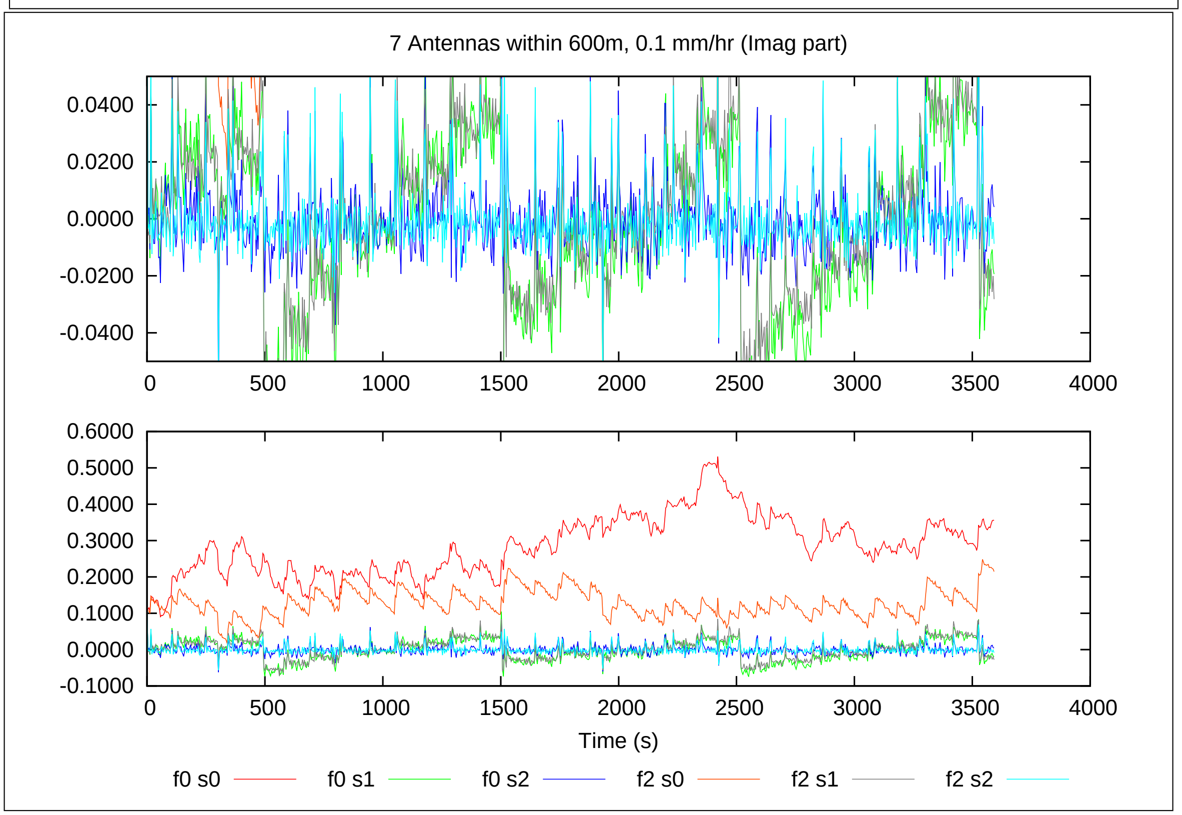
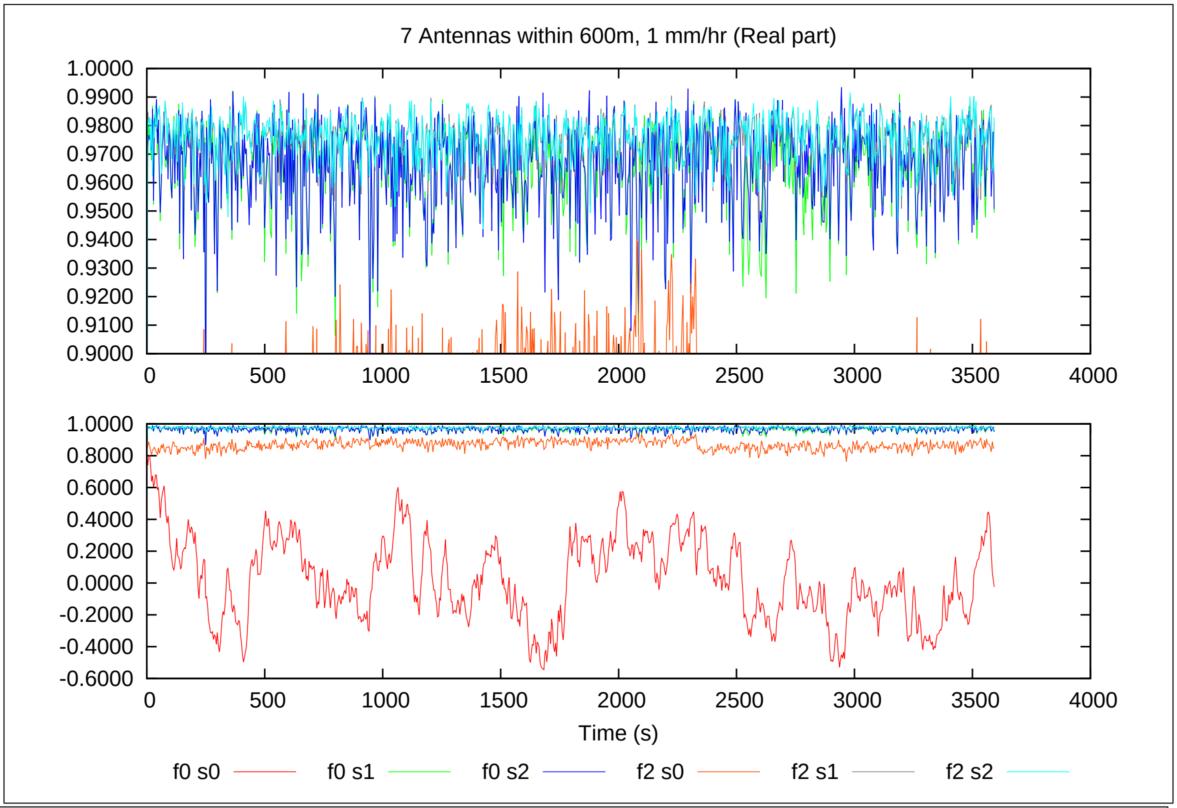


Figure 20: A 7-antenna phasing loop sample with more moisture.



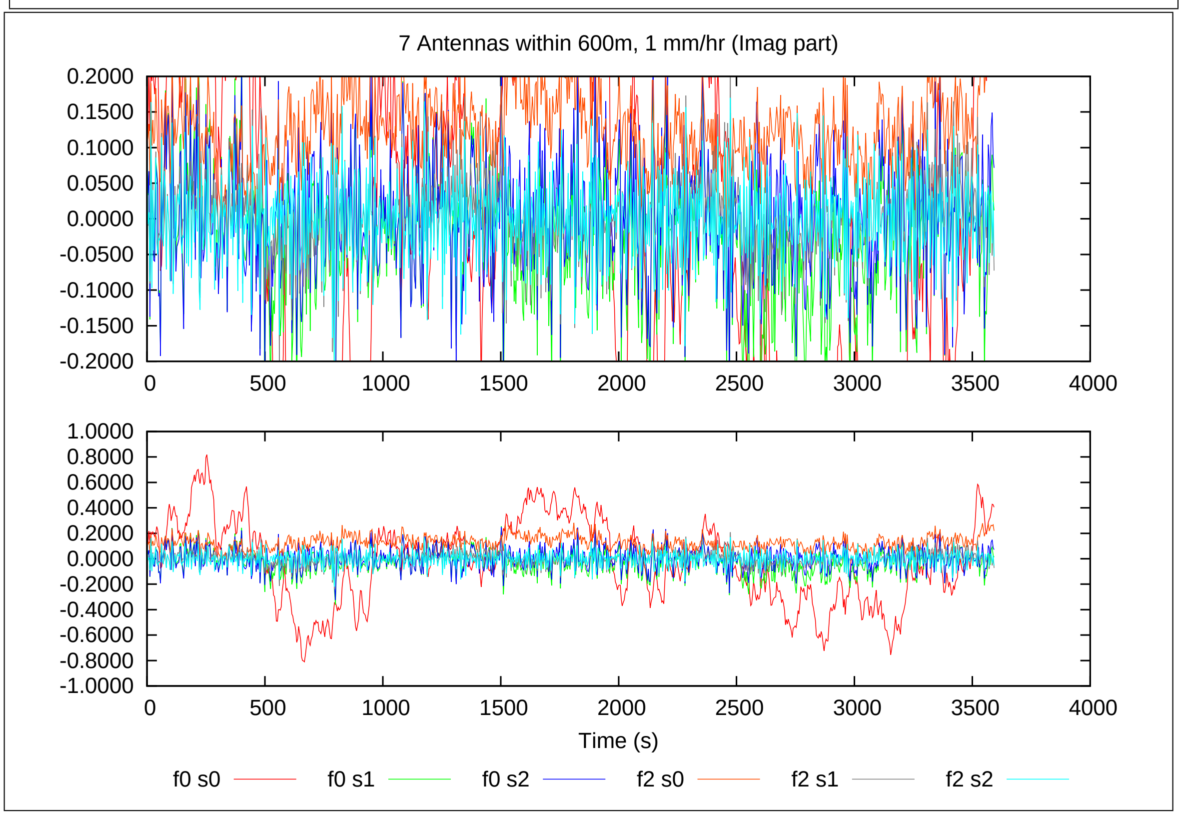


Figure 21: A 7-antenna phasing loop sample with even more moisture.

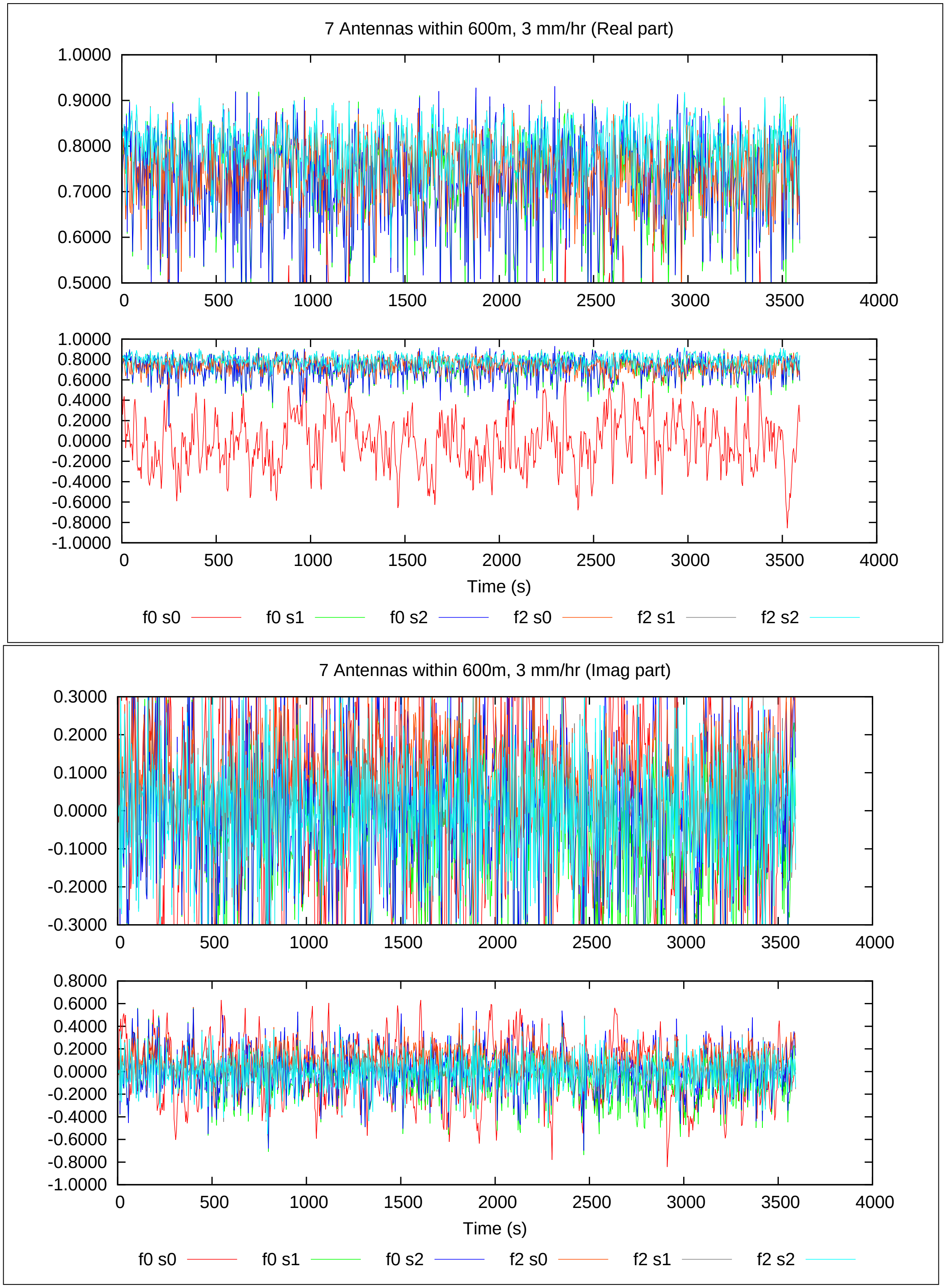


Figure 22: A 7-antenna phasing loop sample with too much moisture.

6 Next Steps...

Correlating these results with sample ALMA data is a next step. The 16-antenna sample was carried out in conditions corresponding to good conditions, and about 1 turn of phase in 5 ks.

```
$ delaystudy -a160 -n-160 -d0
# 160 center of 16 is 2225073.89,-5440105.59,-2481571.61
#-r:  min 76.28 < ave 298.35 < max 1972.89
#-l A075,A068,A077,A082,A076,A021,A046,A071,A011,A072,A025,A074,A069,A138,A053,A132
#-bl: min 42.65 < ave 448.98 < max 2287.51
```

The center appears to have been in the ACA array.

A Usage Notes

The code `delaystudy` was written to examine some of the questions that have come up. It is intended to be played with and expanded. It probably can be enhanced to perform a higher fidelity simulation of the ALMA phasing system.

It does have a help facility which is largely self-explanatory.

```
$ delaystudy --help
Usage: delaystudy [options]
where the options are:

 -v             increases verbosity
 -a <int>       calculate/use array center
 -c <file>      file for commentary (stdout)
 -d <float>     specifies duration (1 h)
 -i <float>     specifies integration time (4096 ms)
 -n <int>       specifies number of antennas (<64)
 -o <float>     specifies offset (0 h)
 -r <mx[,mn]>   specifies (max,min) array radius (12000,0 m)
 -s <float>     specifies step time (512 ms)
 -t <ra,dec>    specifies target (0 deg, 0 deg)
 -q             toggle rms delay effects to sums (integrated)
 -p <phase>     add phasing loop (use help for options)
 -w <wvr>       add WVR delays, &c. (use help for options)

which creates and solves an n-antenna delay problem for a
target at the specified position for the specified duration.
The default is 3 antennas chosen at random with relation to
the center of the current delay center of the array (CoA).

Set GSL_RNG_SEED in your environment for different seedings.
Set ANTENNA_REPORT_WIDTH with number of columns of output.

The target R.A. is ignored at present--rather the observation
is scheduled to straddle maximum elevation subject to offset.

Currently implemented lists are:
    1  current AOS pads
    2  current ACA pads
    3  current POW pads
  160  Cycle-0 16-ant test case
  321  Cycle-1 32-1 array
  322  Cycle-1 32-2 array
```

```
  323  Cycle-1 32-3 array
  324  Cycle-1 32-4 array
  325  Cycle-1 32-5 array
Use -a X to compute and use their centers.
Use -n -X to load the list, rather than a random set.
Use -a 9999 to use CoA for the center.
```

There are a variety of options controlling the noise in the simulator:

```
$ delaystudy -w help
Help on WVR noise parameters:

  seed=s        specifies a random number seed
  lofreq=f      baseband LO frequency (GHz)
  coeff=c,n-m   assigns a diffusion coefficient value c
                (e.g. 0.01 mm^2/s) to antennas n-m
  csigma=s      fuzzes up the coefficients with rel. sigma s
  wsigma=w      fuzzes up the wvr readings with rel. sigma w
  rate=r,n-m    assigns a phase drift rate r
                (e.g. 0.5 deg/s) to antennas n-m
  rsigma=s      fuzzes up the drift rates with rel. sigma s
  repant=n      antenna index to report path delay on
  phsigma=d     initial antenna phase noise sigma (deg)

The antenna range specifier also may be empty (in which case
the assignment is to all antennas, or just provide a value
to be assigned to a single antenna.  The ?sigma cases treat
the corresponding supplied value as a mean and pulls a value
to use from a gaussian distribution with the provided sigma.
(The sigmas are relative to the mean value.)

The -w option should be repeated as needed to fully specify
the desired noise configuration.
```

And still more options governing the phasing loop:

```
$ delaystudy -p help
Help on Phasing loop parameters....

  fast=f        activates fast(WVR) correction
                using the f-th previous sample (f>0)
  slow=c        do slow correction using c channels
  ref=i         reference antenna index
```

The code is a work in progress....

B Average Phase

Assuming perfect response, each antenna signal is 1 and is rotated by a phase $\varphi = \omega_T F B \delta(t)$ because of the residual delay $\delta(t)$. On short timescales Δt, $\delta(t)$ varies linearly in time, so

$$\langle e^{i\varphi} \rangle = \frac{1}{\Delta t} \int_t^{t+\Delta t} e^{i\varphi(t)} dt$$

through a change of variables becomes

$$\langle e^{i\varphi} \rangle = \frac{1}{(\beta - \alpha)} \int_\alpha^\beta e^{i\varphi} d\varphi = \frac{e^{i\beta} - e^{i\alpha}}{i(\beta - \alpha)}$$

where we have used $\beta \equiv \varphi(t + \Delta t)$ and $\alpha \equiv \varphi(t)$ for notational convenience. Decomposing into real and imaginary parts:

$$\langle e^{i\varphi} \rangle = \frac{\sin\beta - \sin\alpha}{(\beta - \alpha)} + \frac{\cos\beta - \cos\alpha}{i(\beta - \alpha)} \rightarrow \cos\alpha + i \sin\alpha$$

in the limit $\beta \rightarrow \alpha$.

For antennas where the delay changes slowly enough, this average over the dump or integration time can be used. (The code tracks α and β at each step.) The discontinuities are easily noticed (α and β are not close); and for those cases, the integral here is replaced by integrals of two pieces (*i.e.* integrating from α to one extreme ($\pm\varphi_m$), then from the other extreme ($\mp\varphi_m$) to β).

For antennas with a faster delay variation, it is more work to get this right: the average includes these two end pieces plus some number of integrals sweeping between the extremes (with a contribution of $\sin\varphi_m/\varphi_m$). We have not bothered with this refinement.