ATA Memo 33 rev 1

Pointing/Trajectory Considerations for the ATA

David DeBoer

October 22, 2001

ABSTRACT

This memo considers the effect of relative offsets and parallax due to the spatial extent of the array, as well as the performance of different trajectory models. Sending array-center corrected quadratic or cubic az/el trajectories every few tens of seconds, with a simple correction buffer pair at each antenna, is suggested as an effective approach. Sending a vector of points is also discussed.

I. INTRODUCTION

The baseline design for pointing of the ATA has the central computer feeding each antenna processor a pointing trajectory, which may be *N* terms of a Taylor series or coefficients of a fitted polynomial, which gets periodically updated. Presumably, these terms represent the azimuth and elevation at some reference point near the array center and are hence identical for each antenna. The local processor would then apply its own coordinate corrections for known pointing offsets in its mount. Other relative pointing offsets are (1) differences in pointing due to offsets in latitude and longitude and (2) parallax differences across the array for nearby objects. These will be discussed in this memo, along with an analysis of polynomial order and update rate for the trajectories.

II. POINTING DIFFERENCES

To first order, the elevation corrections for offsets from the reference position due to displacement in longitude and latitude are

$$\Delta El = \frac{\cos\delta\cos L\sin H}{\cos El}\Delta\ell \qquad (1a)$$

$$\Delta El = \frac{\sin\delta\cos L - \cos\delta\sin L\cos H}{\cos El}\Delta L \qquad (1b)$$

Where L is the reference latitude, ℓ the longitude, H the hour angle and δ the declination. For azimuth, the first order corrections are

$$\Delta Az = -\frac{\cos\delta\cos L(\sin\delta\cos L\cos H - \cos\delta\sin L)}{[\sec Az(\sin\delta\cos^2 L - \sin L\cos L\cos\delta\cos H]^2}\Delta\ell \qquad (1c)$$

$$\Delta Az = \frac{\cos\delta\cos L[\cos H\cos^2 L(\cos\delta\sin H - \sin\delta) + \sin L\cos L(2\sin\delta + \cos\delta\cos^2 H) - \cos\delta\cos H\sin H\sin^2 L]}{[\sec Az(\sin\delta\cos^2 L - \sin L\cos L\cos\delta\cos H]^2}\Delta L \qquad (1d)$$

To get an idea of their maximum values, note that ℓ ≈ 121.°5, L ≈ 40.°8 and δ_{min}≈-30° for Hat Creek. Looking at the elevation expressions, the denominator is smallest at the

highest elevation, which is at transit where H=0. For variations in longitude, this is zero, except for sources passing directly overhead, which are already excluded by the keyhole. Figure 1 shows these variations for various declinations ranging from -30 to 90, varying hour angle rather than longitude and plotting cross-elevation rather than azimuth.

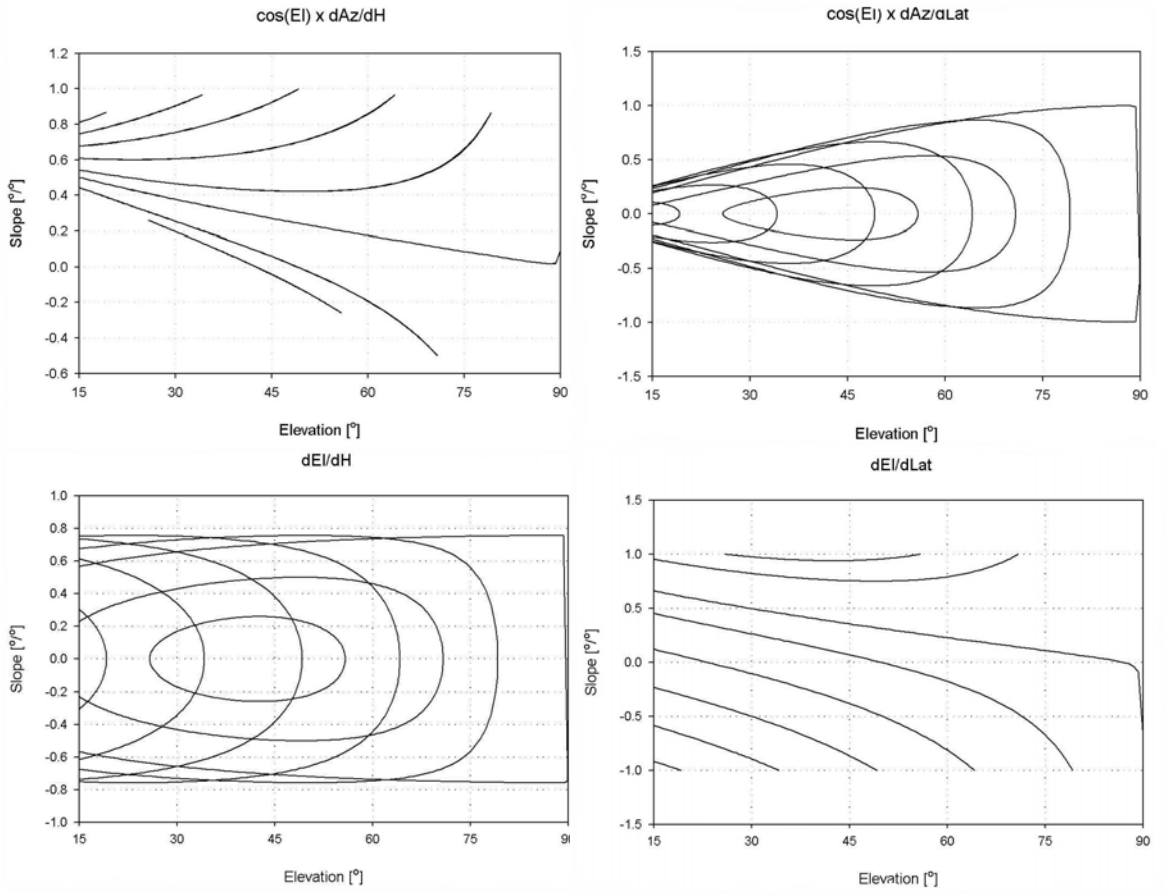

Figure 1 – Variations of pointing during tracks of celestial sources at various declinations.

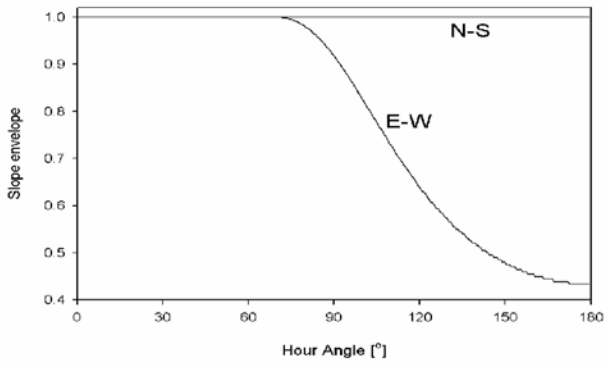

Figure 2 – Maximum envelope from Figure 1.

Figure 2 shows the maximum envelope of the slopes of the total angular separation differences for the positional offsets of celestial sources as a function of hour angle, as opposed to Figure 1, which shows individual tracks for elevation and cross-elevation separately. Figure 3 plots the total variation with offset distance for E-W and N-S baselines assuming the maximum slope of 1 seen in Figure 2. The vertical line in the inset shows the effect for the scale of the ATA as planned. This corresponds to an expected maximum of about 0.3′, or about $\theta_{3dB}/60$ at the highest frequency.

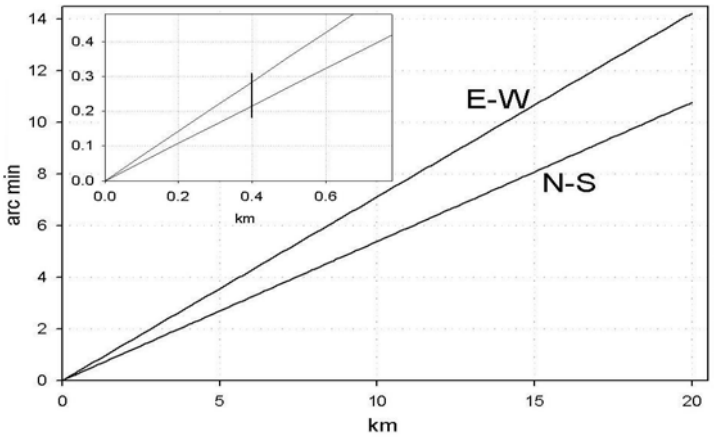

Figure 3 – Maximum pointing offset as a function of baseline.

The effect is therefore sufficiently small to ignore for the planned extent of the initial ATA (where the offset from a central reference position is about 400 meters). For large arrays, the effect does start to get appreciable. One could incorporate a local look-up table based on Eq. 1 to fix this, since an Az/El pair will correspond uniquely to a ΔAz/ΔEl pair.

III. Parallax Differences

Another effect is relative parallax for objects near the earth (primarily the moon, geostationary satellites and low-earth orbiting satellites, but also potentially comets). Figure 4 shows the envelope of maximum parallax differences between the two sides of the array (1 km baseline N-S and E-W) for sources at different distances, representative of Space Station Alpha, Iridium, GPS, the geostationary arc and the moon. The dashed horizontal line is approximately $\theta_{3dB}/20$ for the highest frequency. As is evidenced in Figure 4, objects within about the geostationary orbit (42,242 km) will not be tracked well by the outer antennas, even when fed array-center parallax-corrected az-el trajectories. It should be noted, however, that the GPS and Iridium values are not derived from actual tracks, but rather a (possibly hypothetical) worst case.

Several solutions to account for this relative parallax issue present themselves:

(1) don't worry about it since we probably only want to track satellites with a few dishes, which could be near the center of the array
(2) always send each antenna individually corrected trajectories (ICT)
(3) only send ITC's in a "near-earth mode", otherwise send array-center corrected trajectories (ACCT)

(4) send a source type (and time) along with ACCT's such that it has enough local information (and processing power) to correct itself
(5) send ACCT's and individual parallax corrections, with the parallax corrections potentially updated on a much longer time-scale than the ACCT's.

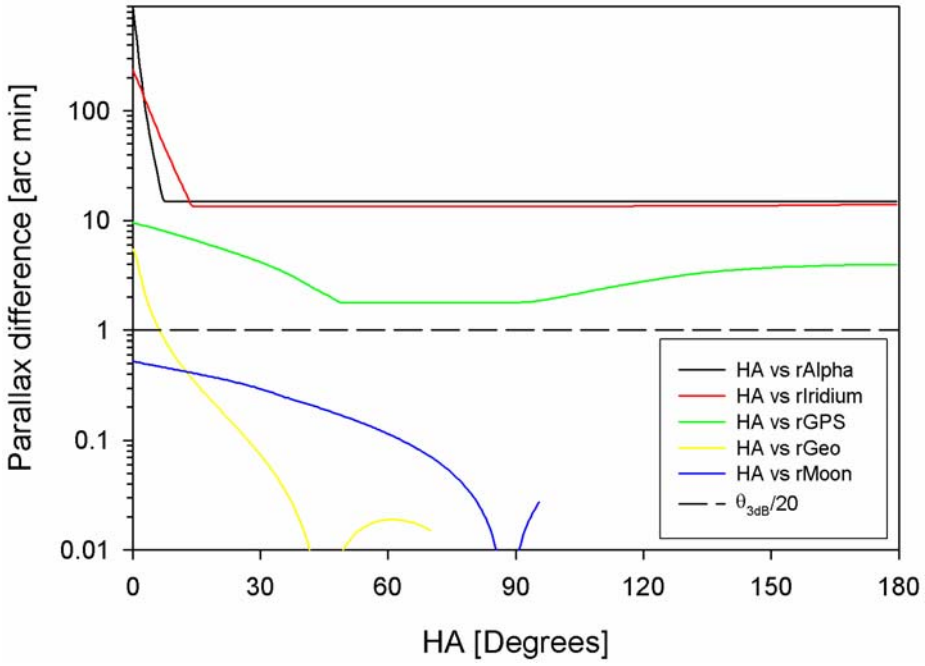

Figure 4 – Maximum parallax difference across a 1 km baseline array.

It could be argued that if you are going to have different "modes", a la (3) above, you would probably do better to do a more comprehensive set of modes, a la (4) above. One possible set could be

(4a) Equatorial – send just RA and Dec
(4b) Geostationary – send sub-satellite point
} Send once and forget about it.
(4c) LEO – send sub-satellite point trajectories
(4d) ACCT
(4e) ICT
…

Starting to get ugly, so I would argue for (1), (2) or (5). To examine how often the parallax corrections would need to be updated, another envelope plot of the rate of parallax difference change is shown in Figure 5, which assumes the same diagonal 1 km N-S and E-W baseline. It is again, an amalgamation of worst cases. Furthermore, Figure 5 assumes a static celestial position, however all of these satellites are obviously moving very fast. The 88 min and 100 min orbits (similar to the space station and Iridium) will pass horizon-to-horizon in less than 7 min and 15 min respectively, with the equality if they happen to pass straight overhead. Obviously this movement will dominate the relative parallax rate, not the rate in Figure 5. A simple-minded estimate of that effect for Iridium is the blue line (the rate is just scaled by the above-horizon-time/24 hours). For the 12 hour orbit (similar to GPS), it could on order of 5 hours, so the parallax rate of change in Figure 5 is a more plausible worst case for that case. The rate obviously drops

with baseline (as do all of the other effects). A statistical analysis of real tracks is necessary for the real answer and is currently being pursed. Note that the $\theta_{3dB}/20$ line throughout is for 11 GHz, which is about $\theta_{3dB}/200$ at the frequencies where those satellites transmit.

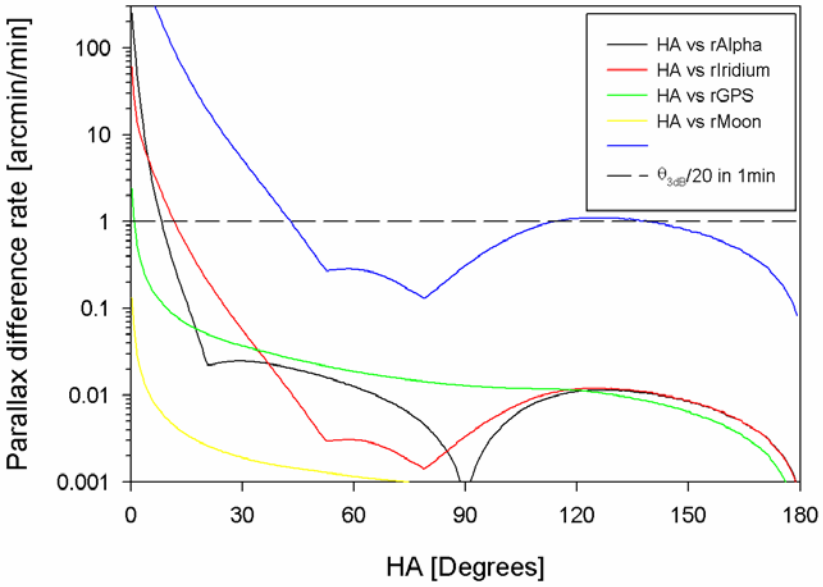

Figure 5 – Maximum rate of parallax change for 1 km (and a 100 meter) baselines for different orbits.

III. TRAJECTORIES

The other issue to address is the order of the trajectory, as well as the update rate. Figure 6 shows the performance of 0 – 3rd order polynomials with a one minute update rate at several declinations. Only the portions above a 15° elevation are shown and the horizontal dashed lines show 1″, 1′ and 1°. The black line shows the expected 15′ pointing error after one minute for a constant value (note that the zeros have been chopped off by the log plot, otherwise the 0 error at the beginning of the constant plot – and for the linear, at the beginning and end – would obscure the quadratic and cubic plots). As expected, the only pointing issue occurs near the keyhole, where the quadratic and cubic errors approach a good fraction of a degree.

Figure 7 shows the quadratic and cubic models with different update rates for a declination of 40.6°. The gaps are, again, places where the error is zero to the precision used. Note that the value at transit is not appreciably reduced with shorter update times (it may be difficult to see, but the black line has essentially the same maximum as the others). To the desired level, the accuracy is basically the same for the three cases. The error at transit depends on the exact sampling of the track, but this would be difficult to exploit. The error is knowable, and one could conceivably use the correction buffers to fix it for that very limited range of observing. Figure 8 shows the cubic expression error for the same declinations with a 10 minute update time, which is sufficient for most observations.

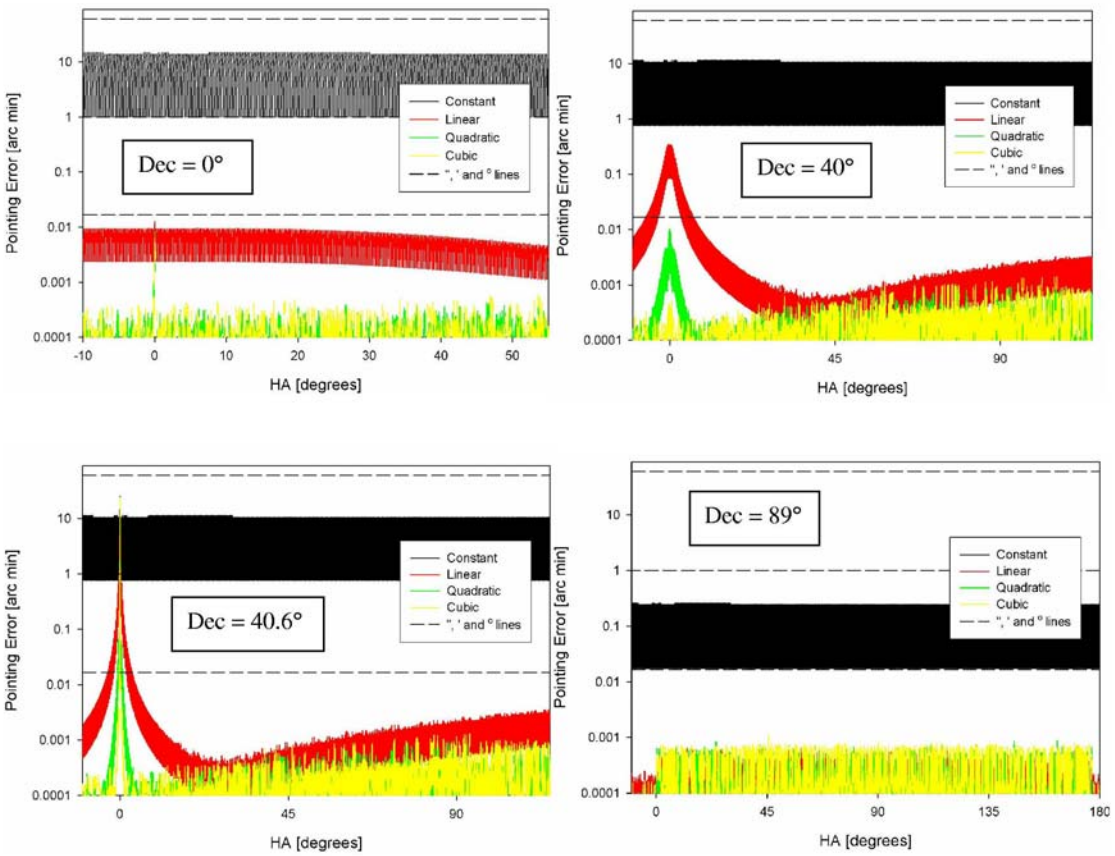

Figure 6 – Pointing error for polynomial trajectories with a one minute update time.

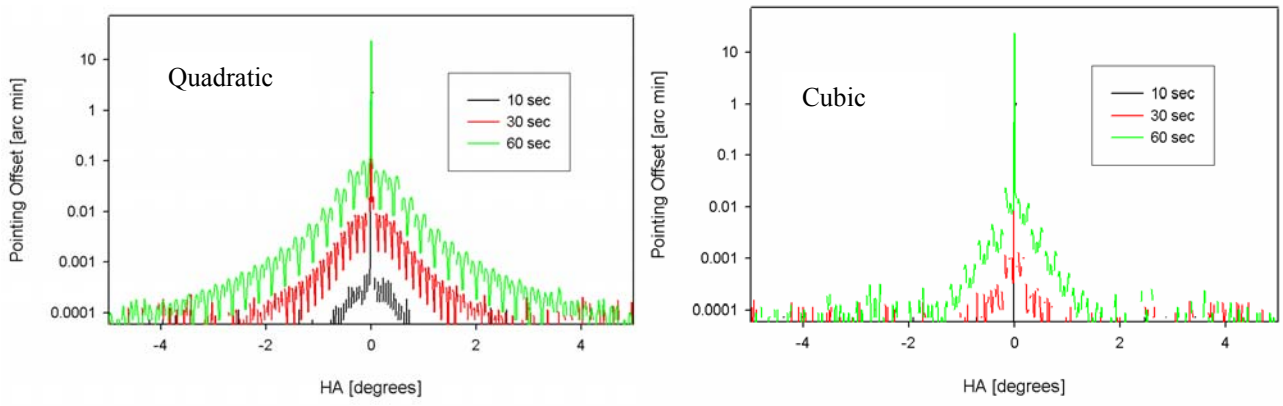

Figure 7 – Pointing error for quadratic (left) and cubic (right) expressions at a dec of 40.8°, with several different update times.

The use of trajectories also implies that the antenna knows when to start applying the given trajectory. The implications of time accuracy have already been examined in Section II, since longitude and time differences are essentially equivalent (the lack of complete equivalence comes whether you use sidereal or civil time). The 1 °/° maximum

of Figure 2 becomes 0.25′/1 sec in terms of time, *i.e.* if you are off by one second in your timing your pointing will be off by a maximum of a quarter of an arc min.

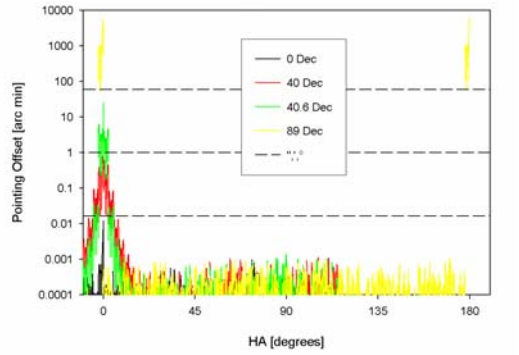

Figure 8 – Pointing errors for cubic expression at different declinations with a 10 minute update time.

IV. CONCLUSIONS

There are many, many caveats of the above (primarily the parallax rate material for LEO's), but some reasonable conclusions may be drawn based on these worst case amalgamations. With the 1 km maximum baseline assumed above, sending only the array-center corrected az-el trajectories is adequate for the vast majority of observing scenarios (e.g., any astronomical body – if a comet gets close enough, we'll be very worried).

If desired, a look-up table could be implemented to account for pointing differences, although for a one km extent these differences are fractions of a beamwidth ($\sim\theta_{3dB}/60$ at the highest frequencies). This table can be incorporated easily into the pointing correction look-up tables, so it might as well be done.

Parallax corrections for GPS and larger orbits may be effected by a simple az-el correction buffer updated on a (many) minute time-scale for each antenna (and obviously for geostationary, one need only send it once). For low-earth orbit satellites it is unlikely that we will need/want to track them with the entire array, and using a few close-in dishes should be fine. Furthermore, for some orbits and some portions of orbits, accurate tracking with the entire array should be achievable with the same strategy of slowly updating the parallax-difference correction buffer, particularly since these satellites transmit in L-band.

Updating the pointing trajectories for celestially stationary sources on the order of a minute using a cubic expression appears adequate (the quadratic is probably adequate as well), with some concern very near the keyhole. The correction buffer could be used here as well to allow pointing at that point in the sky (the values are the same array-wide, so could be broadcast).

V. APPENDIX

Another good alternative would be to send an array of "pointing ensembles" to each antenna and let it calculate its intra-sample trajectory locally. One such ensemble is [Az, El, 1/*r*, *t*], where *r* is the distance to the object relative to the center of the earth scaled to the earth's radius and *t* is the time at which that point is valid. This approach has been suggested by Gerry Harp and Rob Ackermann (with the suggestion to send 1/*r* rather than *r* by Mike Davis) and provides flexibility. Using 1/*r* obviously makes it easier for sources at infinity (*i.e.* at distances greater than that of about the geostationary arc). This value is then bounded by $0 \le 1/r < 1$ (earth radii)$^{-1}$.

The number of ensembles to send is still related to the desired net update time (*e.g.* the time frames discussed in section III) and the order of the polynomial. For a desired update period, *U*, and polynomial of order *n*, the ensembles to be sent must be tabulated at time intervals of *U/n.* The total number of ensembles to send to track for a total time *T* may be expressed as

$$N = \left(\frac{nT}{U} + 1 \right).$$

The total number of values to send is ρN, where ρ is the number of terms in each ensemble. Assuming n=3, U=10s, T=10min and ρ=4 above, requires N=181. Using single precision then requires about 5.7 kB of memory for the ρN =724 values. Note that using linear interpolation every 1 second over the same 10 minutes requires about 18.8 kB of memory.

If two such arrays of ensembles are used, an interrupt may be set to switch between them; both allowing a seamless extension of the net total tracking time as well as the ability to change sources. It also allows for a means to change the update period, *U*, for example if required near the keyhole. The interrupt would also trigger when at a prescribed distance from the end of the buffer to send it to the other ensemble array.

There are many implementation details that affect the precise numbers. For instance, it may make more sense to send values associated with each ensemble array that contain the quantities [t_o, *U/n, N*], where t_o is the reference time for the ensemble array, and again *U/n* is the tabulated time and *N* the total number of ensembles. The ensemble would then be an ordered list of [Az, El, 1/*r*] (so ρ=3). This requires about 4.3 kB for the values above and may also have implementation advantages.