NATIONAL RADIO ASTRONOMY OBSERVATORY
MEMORANDUM

DATE: September 16, 1996
TO: M. Clark, B. Garwood, D. Hogg, H. Liszt
FROM: Ron Maddalena
SUBJECT: GBT and Aips++ requirements for traditional, all-sky pointing

During the commissioning of the GBT, we will need to do certain tests on the pointing of the telescope (see, for example, GBT memo 125). One test is an all-sky measurement of the pointing of the telescope. The test will find the coefficients used by what has been labeled the traditional pointing model. This memo describes some software requirements for the all-sky pointing tests.

The all-sky observations probably will be similar to how we test the pointing on the 12-m and 140-ft telescopes. As such, most of the requirements for the GBT are not far beyond the capabilities of the current 140-ft or 12-m software. In the tests, hundreds of pointing measurements are made toward a host of sources. The steps in all-sky pointing are:

- Create the observing file,
- Make the pointing observations,
- Reduce each of the pointing observations to produce pointing offsets,
- Fit the derived pointing offsets to the traditional model to find the pointing coefficients that produce the best all-sky pointing,
- Analyze the results of the all-sky fit.
- Feed any new coefficients or pointing terms into the GBT control system.

I. Observing files:

I envision that we will create observing files that nearly automate 24-hours of pointing measurements. A file would be specialized for a range of frequencies (since one would use different sources at 22 GHz than at 1.4 GHz), whether the observations are from Gregorian or prime focus (since the models might depend upon the receiver location), and the form of the pointing model. The files should concentrate observations at the locations in the sky that best fit the model. For example, if the elevation pointing offset is expected to depend on, let us say, A + B*cos(elevation) + C*sin(elevation), then one should not concentrate measurements at elevation=45° since A, B, and C could not be determined independently. Rather, one should concentrate observations as close as possible to elevation=90° and elevation=0°. The files should be usable for years without further modification.

Programs should be developed to help in the creation of the files and should be a little bit better than those currently in use. I picture a program that will take as input a user-supplied catalog of strong, point-like sources, a starting source, and a start time. The program would generate a line in the observing file that describes a pointing observation for the first source. Then, it would graphically display the sources that would be above the telescope's horizon at the end of the observation and the user would use a pointing device to specify the source to be observed next. The program would generate a line in the observing file for the next chosen source. After allowing for the time for the observation, telescope move time, and any overhead, the program would regenerate the display and show the sources that would then be above the telescope's horizon. It would also add something like comments to the file to tell the operator what time a source is expected to be observed. (I do not think we will need a 'dream' program that will use A.I. and a set of criteria to pick without human intervention the order of sources). The user and software would continue adding sources to complete the observing file.

II. Observations:

The observations might go on faster or slower than planned, or the observations might be interrupted, or might not start on time. Thus, the operators will need the powers to jump around within the file and to start and stop anywhere in the file.

Each pointing observation can be taken in many ways but in this memo I will assume we will use a method similar to what we use on the 140-ft. A measurement might consist of four scans, the first two with the telescope slewing east-to-west and west-to-east through the expected source position and the last two with the telescope slewing north-to-south and south-to-north. Slewing in opposing directions eliminates any pointing offset introduced by the time constant of the backend. Other possibilities include five-point observations, commonly used at the 12-m, or slewing the telescope in circles around the expected source position (see GBT memo ???). The GBT M&C system and Aips++ has to be able to easily switch between any of the commonly-used pointing techniques.

The data might be reduced in real time with updates in the pointing offsets sent automatically to the control system for the next observation. This memo does not describe the on-line, real-time reduction of pointing data but the more casual, off-line reduction of hundreds of observations for the all-sky pointing. I am also assuming that the measurements were made explicitly to measure the all-sky coefficients (though limited use could be made of the observations made during routine observing).

III. Reduction of a pointing observation:

Since we will be dealing with many observations, a data base manager should pick the user's desired observations. The manager should allow choosing observations at least according to project code, date, time, scan numbers, and observing technique. The software would then loop through the list of selected observations and reduce the data for each observation. The details of the data analysis depend upon the chosen observing technique but the analysis for most techniques will follow the same steps. If we use the four-scan 'slew' technique described above, then the data reduction of the individual pointing measurements could consist of the following:

- If the observations involved any switching scheme (e.g., nutation of a secondary or tertiary, or firing of a noise diode), then the software needs to combine the multiphase data. The software also needs to find the **measured** position of the telescope for each data sample. All following analysis steps should deal with **measured** intensities (i.e., in units of T_A) as a function of **measured** positions.
- A baseline needs to be fit to those samples in a scan when the telescope is off the source. The order of the baseline should have a default value of one but the user

might want to specify a higher order if, for example, ground pickup or solar radiation is a problem. The software should have an automated but user-modifiable way to decide what samples it should use for the baseline. For example, the program might use the first and last 20% of the samples for the baseline fit unless the user specifies a different percentage.

- If the data is very noisy (e.g., the weather was bad and the observing frequency was high), one should have the option to convolve the data with a function that best represents the expected shape of the beam (e.g., a Gaussian). This optimal filtering technique has proven very useful at the 140-ft for recovering useful information out of what at first looks like useless data.
- If the beam is very nearly Gaussian, then the software would do a nonlinear, least-squares fit of a Gaussian to the data. The software would first find initial guesses for the center, height, and width of the Gaussian. The software could pick as first guesses the sample where the peak intensity occurs (to define the center and height) and the samples where the intensity is half the peak intensity (to define the half width). Also, the Gaussian should not be fit to all of the samples in the scan (most beams are only Gaussian near their center) but to some user-defined fraction of the scan around the peak intensity. A default region for the fit could be the samples that lie between the half-power points.
- If the beam is non-Gaussian (e.g., at high frequencies), then the software might need to fit a different function (see GBT memo ???). For a non-symmetrical beam, one has to address the issue of whether the pointing will be defined by the position of the peak response or by the median of the response. The decision should be based on what kinds of observations the telescope will be doing with the derived pointing offsets. For example, if the astronomical observations will be of an extended source (like a high-frequency, spectral-line observation of TMC-1), one might want to use the median of the response to define the pointing offset.
- No matter what fitting technique we use, the results of the fit would be a measured position for the source in two directions and maybe the beamwidths in two directions and the measured source intensity. Not only should the software report the results of the fit but also the rms of the residuals and the formal errors of the derived quantities. We can use the measured beamwidth and intensities as a diagnostic check on the observation since one can predict these quantities. Also, we can use the intensities and widths to derive antenna and beam efficiencies.
- The software should take the difference between the measured and catalog positions of the source and derive the difference between the measured and catalog position in the telescope's native coordinate system (azimuth and elevation for the GBT).
- The software should repeat the above for the four scans within the pointing measurement. It then should take averages or differences between the derived quantities and provide a human-readable summary of the results. Included in the summary should be auxiliary information like LST, UT, weather conditions (since the refraction terms in the pointing model depend upon the weather). A programmer could use a 140-ft pointing summary, an example of which I have attached, as a reasonable starting point for a GBT summary file.
- Since the results of the fit will then be fed into the algorithms for finding the all-sky pointing coefficients and maybe an all-sky efficiency curve, the software also needs to create tables that the all-sky algorithm(s) can digest.

We also need to include certain tools to debug the occasional pointing problems. For example, we will need software that will plot the **measured** telescope position against either time or the commanded telescope position. A description of these tools is beyond the scope of this memo.

IV. All-sky model fitting:

The form of the traditional pointing model that we should use for the GBT has been a matter of some debate (see GBT memos ???). Both of the competing models describe the pointing of the telescope as a function of azimuth and elevation. The reduction of the all-sky function is least-square fit of the measured pointing offsets to the model; the fit will find the values of coefficients in the model that produce the best all-sky pointing. The two competing methods would require separate fitting routines since their basic assumptions are different. Thus, Aips++ should devise an easy way to switch between the two fitting methods. Similarly, M&C should allow one to easily switch between the model used for the on-line pointing.

- The all-sky fitting must allow the user to specify which coefficients are to be fit and which are to have values that the user will supply. This is imperative in the cases that a component of the telescope has been altered and one wishes to update those coefficients that depend on that component. For example, if we have replaced an elevation encoder, we do not need to do an all-sky observation but instead one can make a limited set of observations to measure the few pointing coefficients that depend upon the elevation encoder (e.g., zero offset and eccentricity of the encoder).
- If the algorithm uses a nonlinear, least-squares fit, some method must be provided that will allow the user to specify initial guesses for the coefficients.
- A fair fraction of the measurements at high frequencies usually cannot be used because of either the weather or the typical weakness of sources at high frequencies. The software can probably expunge bad data according to some basic criteria. For example, if there are 100 measurements and, after a fit of the model, one measurement has a residual more than, let us say, four times the rms of the residuals, the software should eliminate that point and redo the fit. One algorithm I have used first does a least-magnitude fit (not a least-squares fit) to find discrepant points and then after eliminating the bad points, finishes with a least-squares fit.
- Care should be taken that the proper function is fit. For example, if Δx and Δy are the two measured pointing offsets, the fitting procedure should not minimize $\Delta x+\Delta y$ but $\sqrt{(\Delta x^2+\Delta y^2)}$.
- The fitting algorithm should report the formal errors on the derived coefficients, the rms of the residuals, and the covariance matrix. The covariant matrix is an important diagnostic tool one uses to test whether, for example, the observing file did not concentrate enough observations at certain sky locations.

V. Analysis of the all-sky fit:

The tools we will need to look at the results of the all-sky fit depend upon how well the telescope performs. If the telescope's pointing is reasonable, then we can get away with a few plotting tools. A full list of the necessary tools will become apparent only after we have used the telescope.

One tool should plot the location in azimuth and elevation at which the non-expunged measurements were made. Another should plot the azimuth and elevation residuals for every source (which should help catch the occasional source with a badly-specified catalog position). Other plots should be of the pointing residuals (either azimuth, elevation, or both) as a function of azimuth for all measurements in a certain range in elevation. Similarly, another set of plots should be of the pointing residuals (either azimuth, elevation, or both) as a function of elevation for all measurements in a certain range in azimuth. Any significant, residual shape in the latter plots suggests that the model do not fit the measurements. This might lead the user to investigate what is it about the telescope that has produced an unexpected pointing error. At the least, the plots should help determine what other terms might need to be added to the model.

VI. Applying the results of the fit to the control system:

From time to time, and especially in the telescope's early years, we will be updating both the pointing model and the coefficients used by the model. As such, the control system should easily allow for the addition and changing of models and coefficients. Since different focus locations or observational styles might require different models or coefficients, the control system must have the ability to simultaneously store different models or coefficients and to easily switch between different models or different coefficients. I am not sure whether we should give observers the power to switch to models or coefficients; this power might best be held only by NRAO staff. Also, the GBT software should make it apparent to the operator and astronomer what model and set of coefficients are currently in use.

VII. Conclusion:

There are many decisions that we will need to make concerning various aspects of the technique for measuring the all-sky pointing on the GBT. The above gives the detailed requirements for only one set of options but I hope that my discussion helps define the requirements for other possible options.

Staff at Green Bank, Tucson, and Socorro has developed many tools to help determine the all-sky pointing of their telescopes. The GBT software we develop should use as a starting point the ideas, software and experience accumulated over the years at the various NRAO sites. The requirements I have outlines above are not that more sophisticated than the existing software. As such, producing the necessary GBT software should not be a major effort.