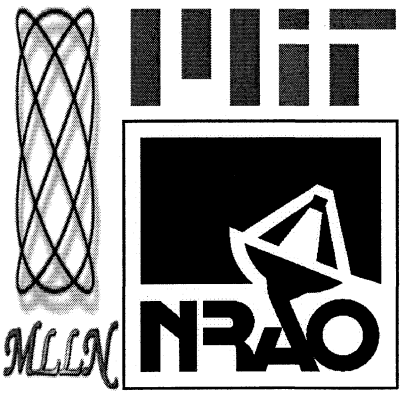

MLLN Note #6

Review of Green Bank Antenna Control Software

Glen Langston
NRAO Green Bank
May 18, 2005

ABSTRACT

The MLLN note reviews the existing Green Bank observing system software packages and their applicability to the MLLN Project.

This review finds several observing system control packages have been used to control telescopes in Green Bank and that each package has its own unique advantages.

The packages considered here are: the GBT M&C system, the 85-3 control system, the VLBA control system, the 20m VLBI field system, and the OVLBI control system.

Change Record

Revision	Date	Author	Sections/Pages Affected
	Remarks		
1.1	2005-May-18	G. Langston	All
	Flesh out details		
1.0	2005-Apr-25	G. Langston	All
	Initial version.		

1. Background

This document is a reference for comparison of the features of the NRAO antenna control systems a that will be used to build the control system for the 43m. The MLLN bi-static radar project is primarily a requires a standalone, non-interactive system that will function automatically.

For the bi-static radar project, 43m will be controlled by command files that are generated weekly and will be executed without operator intervention. The primary functions of the real-time system are implement the scheduled commands, logging monitor data and calling out for technical support in case of detected system anomalies.

The bi-static current development plan calls for a number of computers to be used for real-time control. The functions of these computers are described in MLLN Note 7. The computer functions are summarized here.

1.1. YGOR/GBT

The YGOR system has been designed for control and monitoring of the many pieces of hardware that make up the Green Bank (Radio) Telescope (GBT). In order to do this YGOR must be able to command (i.e. setup system parameters) and control (i.e. make an observation happen) these devices in an efficient manner. YGOR also must provide monitoring for logging purposes, and an alarm/messaging system for those un-expected failures.

The YGOR documentation is at: `http://www.gb.nrao.edu/GBT/MC/doc/WhoYgor.html`. The YGOR code is written in C++ with addition of a few C routines and some libraries of Fortran code. YGOR does not provide a graphic operator interface, but contains an interface that allows connecting a separate software package as a user interface. For the GBT, several graphical interfaces have been the developed. The engineering graphical interface is the CLEO package:

See `http://www.gb.nrao.edu/ rmaddale/GBT/CLEOManual`.

For astronomers, the interface has been GBT Observe (GO), which is now being replaced by the Astronomer's Integrated Desktop (Astrid):

See `http://wiki.gb.nrao.edu/bin/view/Data/AstronomersIntegratedDesktop`. The Astrid system is not yet stable.

The YGOR and GBT specific code is kept under revision control using CVS. The source code root directory is: `/home/gbt2/repository`

1.2. YGOR/GBI

The Green Bank Interferometer control code is built on the YGOR system, with specific software for control of the three eight-five foot diameter telescopes.

The C++ source code is also kept under CVS revision control and may be extracted using the commands for the GBT code (above) with the addition of the command `cvs checkout gbi`.

The GBI system uses single board computers as Antenna Control Units and a Sun Solaris machine as the station computer.

The GBI user interface is a combination of CLEO screens and Glish scripts. The interface has been reasonably reliable. The GBI observation scheduling software interface is written in Glish. Support for the Glish language is being phased out by NRAO.

1.3. VLBA System

The VLBA antenna control code is installed at each of the 10 VLBA antenna sites. In addition, the software required to configure the IF, formatter and data recorder is installed at the GBT, the Max Planck Institute telescope at Effelsburg and at other sites throughout the world. (There was not VLBA software design document readily available on-line, but the memos leading up to the station design are at http://www.vlba.nrao.edu/memos/).

The VLBA code is written in C. Several versions of the VLBA code are kept on line in Green Bank. The currently the code is kept under the `gbvlbi` account. The latest version (updated 2005 May 11!) is `/users/gbvlbi/code_mk5`.

The VLBA code includes a graphical user interface (SCREENS). The VLBA user interface is fairly simple, but has the advantage that it runs well with very limited network bandwidth requirements.

The VLBA code is kept under revision control using SCCS. The reference source source code is kept at the Array Operations Center in Socorro, NM.

1.4. OVLBI System

The Orbiting VLBI code is based on the VLBA code, with extensive revisions. The OVLBI code is written in C. All code needed to run the OVLBI system is in a single MVME 147 computer (until a recent upgrade of the antenna control unit).

The station computer interface to Tim Weadon's new ACU code has already been implemented in the OVLBI software environment.

The OVLBI control system uses the VLBA SCREENS package for operator display.

The OVLBI code is kept under revision control using SCCS. The reference source source code is kept in the `ovlbi` user area. The latest source code is at:

`/users/ovlbi/codeNew`.

The OVLBI software design documentation is on the web at:

`http://www.gb.nrao.edu/ovlbi/software`

1.5. 20m VLBI Field System

The Field system code was used to control the 20m for the VLBI observations. The field system is written and maintained by the Goddard Space Flight Center. There is extensive on-line documentation at:

`http://lupus.gsfc.nasa.gov/vlbi.html`. Field system code is mostly Fortran with some C functions.

In Green Bank, the Field System code can be found by logging into "spock.gb.nrao.edu".

In Green Bank, to view the Field system code, type:

```
ssh spock.gb.nrao.edu
cd /usr2/fs-9.3.15/
ls
```

2. System Review

The YGOR/GBI/GBT code provides a good model for writing Linux based programs to monitor and control hardware. The YGOR system is the basis for Tim Weadon's code implementing an antenna control unit (ACU) in Linux. This same ACU code will be the basis for the 43m antenna control unit.

3. Pointing Model Software

The pointing model calculations for the 140ft were done offline from the control computer.

For the 20m system, a field system command, `fivpt`, implements a five point peak scan, commanding antenna motion, measuring intensities using the VLBA Base Band Converters (BBCs) and fitting a Gaussian the the peak position offset. The code for `fivpt` is in directory `/usr2/fs-9.3.15/fivpt`.

`fivpt` scans across the source in AZ and EL, fits a Gaussian to the data and reports the offsets in

the log file.

After a set of observations with the 20m telescope, programs were used to summarize and display the data. The program `error` reads the offsets from the log file and fits a pointing model.

Frank Ghigo wrote a independent set of programs in "C" to fit the 140ft pointing model. Alternately, there is an independent set of programs in my area, all written in "C" in `/users/fghigo/TELS/INT/src/` In `gbipoint.c` pointing data is collected from interferometer data records and the pointing model is fit using routines from `pntlib.c` and `fitlib.c`" In `pntlib.c`, a routine defines the pointing model, and the fitting is done with routines in `/user/fghigo/lib/fitlib.c`

4. Conclusions

There are a number of good starting points for the development of the MLLN station computer. Parts of the GBT/YGOR/GBI system will be the basis for the ACU software.

The station computer software will interface via RPCs with the YGOR MCB functions and the 43m ACU. The OVLBI software which implements the interface to the ACU, and will be a good starting point for the 43m station computer.

REFERENCES

This and all other MLLN notes are available on the web at:

`http://wiki.gb.nrao.edu/bin/view/Projects/MLLNProjectNotes`

Glen Langston, Mike Shields, John Stueve, 2005, NRAO. MLLN Note 3. "Long Term Sequence of Events and Local Hour Angle Declination Files"

`http://wiki.gb.nrao.edu/pub/Projects/MLLNProjectNotes/mllnNote003.pdf`

A. YGOR

To work on GBT M&C code, a large number of files, includes and make information must be retrieved from CVS. It is easy to check all of the required files out of CVS, but ofter very difficult to figure out which files are actually used by the function.

The process of checkout is as follows:

```
bash                              % These functions only work in bash shell
source /home/gbt/gbt.bash         % setup the Ygor environment
mkdir ygor                        % start in a clean directory
cd ygor
cvs checkout ygor
touch .depend
make
```

B. S/X Receiver Manager

The 43m pointing tests will be done using the 20m S/X band receiver. The code for the GBI S/X receivers is at:

`/home/gbt1/gbi/devices/receivers/SX_Receiver/Manager`

Ray Creager wrote a recipe for porting VxWorks code to Linux. It is on the wiki:

`http://wiki.gb.nrao.edu/bin/view/Knowledge/HotToPortManagerToLinux`