Observing With The Green Bank Telescope

by GBT Scientific Staff

October 8, 2010

Version 3.0

This guide provides essential information for the preparation of Observing Scripts for observations with the Green Bank Telescope.

Authors, Contributors, and Editors

This Guide is the product of many of the GBT Scientific Staff, with major contributions from:

Toney Minter

Dan Perera

Frank Ghigo

Richard Prestage

Ron Maddalena

Brian Mason

Scott Ransom

Jim Braatz

Dana Balser

Contents

List of Figures

List of Tables

Chapter 1

How To Use This Manual

This document provides the necessary information to be able to perform successful observations with the Green Bank Telescope (GBT).

- In Chapter 2 we briefly outline the features of the GBT and the general observing process.
- In Chapters 3 and 4 we provide an introduction to the Astronomer's Integrated Desktop (Astrid), the GBT observing interface.
- In Chapter 5 we provide example observing scripts that can be used in Astrid. We also provide detailed descriptions of the contents of observing scripts.
- In Chapter 6 we provide information on the strategies that should be used and advanced techniques for observing with the GBT.
- In Chapter 7 we provide a short overview of the GBT Intermediate Frequency system (IF).
- In Chapter 8 we provide the locations of where to find more information about Radio Frequency Interference (RFI).
- in Chapter 9 you are introduced to the Dynamic Scheduling System, and in Chapter 10 is a discussion of the effect of weather conditions on observing.
- In Chapter 11 there is advice on remote observing.
- In Chapter 12 we provide information on what happens before your observations and directions on getting to Green Bank.
- In Chapter 13 we provide information on how to take your data home with you and where to obtain the GBT IDL Data Reduction Package (GBTIDL).
- Later chapters give basics of observing with various specific instruments: Chapter 14 for pulsar observing with GUPPI, Chapter 15 for continuum observing with the CCB, and Chapter 16 for mapping with the Mustang bolometer array.

Additional information and special topics are covered in the Appendices.

New users should read Chapters 2, 3, 4, 5, and 6 in their entirety. They should also read the remaining Chapters as needed.

Chapter 2

The GBT Observing Process

2.1 Overview Of The Green Bank Telescope

The 100 meter Green Bank Telescope is intended to address a very broad range of astronomical problems at radio wavelengths and consequently has an unusual and unique design. Unlike conventional telescopes that have feed legs projecting over the middle of the surface, the GBT's aperture is unblocked so that incoming radiation meets the surface directly. This increases the useful area of the telescope and reduces reflection and diffraction, which ordinarily complicate a telescope's pattern of response to the sky. To keep the aperture unblocked, the design incorporates an off-axis feed arm that cradles the dish and projects upward at one edge. This requires that the figure of the telescope surface be asymmetrical. To make a projected circular aperture 100 meters in diameter, the dish is actually a 100 by 110 meter section of a conventional, rotationally symmetric 208 meter figure, beginning four meters outward from the vertex of the hypothetical parent structure (see Figure 2.1). The GBT's lack of circular symmetry greatly increases the complexity of its design and construction.

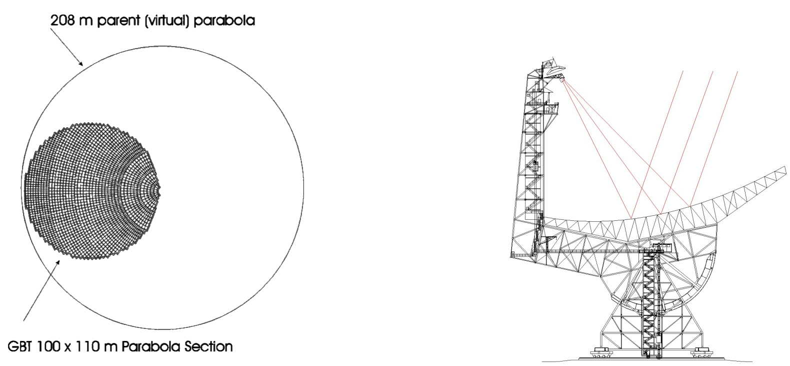


Figure 2.1: Left: The parent parabola for the GBT. Right: The off-axis optics of the GBT.

To maintain precise surface figures and pointing accuracy at high frequencies the telescope is equipped with a complex Active Surface (AS). At higher frequencies gravity distorts the surface figure of the telescope to unacceptable levels. Temperature variations and wind can also deform the figure of the dish. To compensate for these distortions, the surface of the GBT is 'active' i.e. it is made up of 2008 independent panels and each of these panels are mounted on actuators at the corners, which can raise and lower the panels to adjust the shape of the dish's surface.

2.1.1 Main Features of the GBT

- **Fully steerable antenna** $+5^{\circ} to + 90^{\circ}$ elevation range ($-47^{\circ} to + 90^{\circ}$ declination); 85% coverage of the celestial sphere. But note that observing near the zenith ($elevation > about\ 86^{\circ}$) may fail due to the high azimuth rates required.
- **Unblocked aperture** reduces sidelobes, Radio Frequency Interference (RFI), and spectral standing waves.
- **Active surface** compensates for gravitational and thermal distortions.
- **Frequency coverage of 100 MHz to 100+ GHz** 3 orders of magnitude of frequency coverage for maximum scientific flexibility.
- **Location in the National Radio Quiet Zone** Comparatively low RFI environment. (See Figure 2.2).

2.1.2 National Radio Quiet Zone

The National Radio Quite Zone (NRQZ) was established by the Federal Communications Commission (FCC) and by the Interdepartment Radio Advisory Committee (IRAC) on November 19, 1958 to minimize possible harmful interference to the National Radio Astronomy Observatory (NRAO) in Green Bank, WV and the radio receiving facilities for the United States Navy in Sugar Grove, WV. The NRQZ is bounded by NAD83 meridians of longitude at 78d 29m 59.0s W and 80d 29m 59.2s W and latitudes of 37d 30m 0.4s N and 39d 15m 0.4s N, and encloses a land area of approximately 13,000 square miles near the state border between Virginia and West Virginia. More information on the NRQZ can be obtained at http://www.gb.nrao.edu/nrqz/nrqz.html.

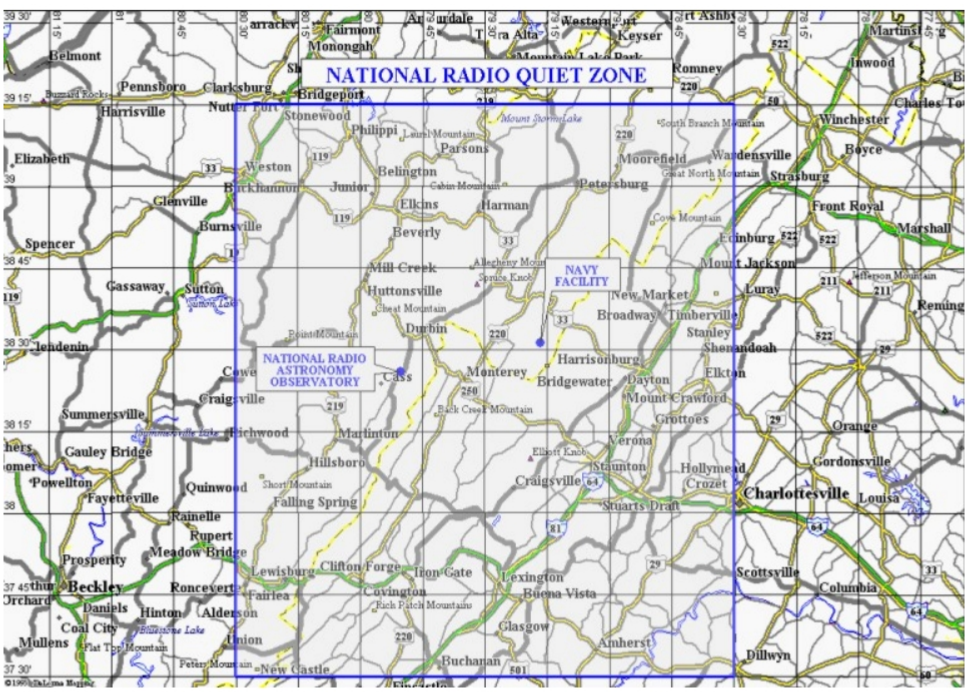


Figure 2.2: The National Radio Quiet Zone.

2.1.3 Front Ends

The GBT receivers cover several frequency bands from 0.290 - 50 GHz and 80 - 100 GHz. Tables 2.1 and 2.2 list the properties of the Prime Focus receivers and the Gregorian Focus receivers. System temperatures are derived from lab measurements or from expected receiver performance given reasonable assumptions about spillover and atmospheric contributions.

The GBT Proposer's Guide (http://www.gb.nrao.edu/gbtprops/man/GBTpg/GBTpg_tf.html) provides more information on the GBT receivers.

Table 2.1: Properties of the Prime Focus Receivers.

Name	Freq. Range (GHz)	Polarization	Beams	Polns/Beam	T_{rec}	T_{sys}
PF1	0.290-0.395	Lin/Circ	1	2	12	46
PF1	0.385-0.520	Lin/Circ	1	2	22	43
PF1	0.510-0.690	Lin/Circ	1	2	12	22
PF1	0.680-0.920	Lin/Circ	1	2	21	29
PF2	0.910-1.230	Lin/Circ	1	2	10	17

Table 2.2: Properties of the Gregorian Focus Receivers.

Name	Freq. Range (GHz)	Polarization	Beams	Polns/Beam	T_{rec}	T_{sys}
L–band	1.15-1.73	Lin/Circ	1	2	6	20
S–band	1.73-2.60	Lin/Circ	1	2	8-12	22
C–band	3.95-5.85	Lin/Circ	1	2	5	18
X–band	8.00-10.1	Circ	1	2	13	27
Ku–band	12.0-15.4	Circ	2	2	14	30
K–band (lower)	18.0-22.5	Circ	2	2	21	30-40
K–band (upper)	22.0-26.5	Circ	2	2	21	30-40
KFPA	18.0-26.5	Circ	7	2	15-20	25-30
Ka–band (MM-F1)	26.0-31.0	Lin	2	1	20	35
Ka–band (MM-F1)	30.5-37.0	Lin	2	1	20	30
Ka–band (MM-F1)	36.0-39.5	Lin	2	1	20	45
Q–band	38.2-49.8	Circ	2	2	40-70	67-134
MUSTANG	80-100	-	64	-	-	-

Note: The K–band Focal Plane Array (KFPA) receiver is in its testing phase and not yet available for general observing.

2.1.3.1 Prime focus receivers

The Prime focus receivers are mounted in a Focus Rotation Mount (FRM) on a retractable boom. The boom is moved to the prime focus position when prime focus receivers are to be used, and retracted for Gregorian receivers. The FRM holds one receiver box at a time. Currently there are two receiver boxes, PF1 and PF2. A change from PF1 to PF2 receivers requires a box change, taking about 4 hours and done only during scheduled maintenance days.

The PF1 (0.29 - 0.92 GHz) receiver is divided into 4 frequency bands within the same receiver box. The receivers are cooled Field Effect Transistor (FET) amplifiers. The feeds for the lower three bands are short-backfire dipoles, and the feed for the fourth (680-920MHz) is a corrugated feed horn with an Ortho-Mode Transducer (OMT) polarization splitter. A feed change, required to switch beteen bands, takes 4 hours. The PF2 (0.920 - 1.23 GHz) receiver uses a cooled FET and a corrugated feed horn with the OMT.

2.1.3.2 Gregorian focus receivers

The Gregorian receivers are mounted in a rotating turret in a receiver room located at the Gregorian Focus of the telescope. The turret has 8 portals for receiver boxes. Up to 8 receivers can be kept cold and active at all times. The Gregorian subreflector can be used for slow chopping, with a minimum 4.5 second half-cycle.

Changing between any two Gregorian receivers that are installed in the turret takes about one to 1.5 minutes.

2.1.3.3 The MUSTANG Receiver

MUSTANG (MUltiplexed SQUID TES Array at Ninety GHz) is a multi-pixel bolometer array observing at 80-100 GHz mounted at the Gregorian focus. It is both a receiver and the associated back end, and is described in Chapter 16. The MUSTANG receiver must be used at elevations above 30 degrees, due to the design of the cryogenics.

2.1.4 Backends

The GBT has two continuum backends, the Digital Continuum Receiver (DCR) and the Caltech Continuum Backend (CCB). Two spectral line backends are also available, the GBT spectrometer and the Spectral Processor. Pulsar observations can be done with the Spectral Processor and GUPPI. There is a single dish mode for the Very Long Baseline Array (VLBA) backend that is available for high time–resolution observations. Planetary radar uses a specialized backend.

For more information on GBT backends please see the "GBT Proposer's Guide" which is available at http://www.gb.nrao.edu/gbtprops/man/GBTpg/GBTpg_tf.html.

2.1.4.1 Digital Continuum Receiver (DCR)

The digital continuum receiver is the GBT's general purpose continuum backend. It is used both for utility observations such as pointing, focus, and beam-map calibrations, as well such as for continuum astronomical observations including point-source on/offs, extended source mapping, etc.

2.1.4.2 Caltech Continuum Backend (CCB)

The CCB is a sensitive, wideband backend designed exclusively for use with the GBT Ka–band (26-40 GHz) receiver. It provides a carefully optimized Radio Frequency (RF) (not an Intermediate Frequency (IF)) detector circuits and the capability to beamswitch the receiver rapidly to suppress instrumental gain fluctuations. There are 16 input ports (only 8 can be used at present with the Ka-band receiver), hard-wired to the receiver's 2 feeds x 2 polarizations x 4 frequency sub-bands (26-29.5 , 29.5-33.0; 33.0-36.5; and 36.5 - 40 GHz). The CCB allows the left and right noise-diodes to be controlled individually to allow for differential or total power calibration. Unlike other GBT backends, the noise-diodes are either on or off for an entire integration (there is no concept of "phase within an integration"). The minimum pratical integration period is 5 milliseconds; integration periods longer then 0.1 seconds are not recommended. The maximum practical beamswitching period is about 4 kHz, limited by the needed 250 micro-second beamswitch blanking time. Switching slower than 1kHz is not recommended.

2.1.4.3 Spectrometer

The GBT Spectrometer provides the observer with a remarkable variety of spectral line observing modes, so that the scientific return of their experiment may be optimized. The spectrometer performs auto- and cross- correlation of the input signals. The input signals may be a) dual polarization IFs in a selected frequency range, b) IF inputs from different feeds of multi-feed receivers, or c) combinations of the preceding with different frequency ranges.

The spectrometer modes are divided into two major types, wide bandwidth, low resolution (800 and 200 MHz) and narrow bandwidth, high resolution (50 and 12.5 MHz). The maximum resolution is 49 Hz in a 12.5 MHz bandwidth.

2.1.4.4 Spectral Processor

The Spectral Processor is an Fast Fourier Transform (FFT) spectrometer primarily designed for high time–resolution pulsar observations. Because of its wide dynamic range it is also useful for spectral line observations at low frequencies where strong interference is a problem. It contains two FFT engines, each with 1024 channels over a maximum bandwidth of 40 MHz which may be divided into 1, 2, or 4 separate passbands. The two FFT engines are synchronous and their outputs may be cross-multiplied to measure polarization. The most commonly used types of observing with the Spectral Processor are pulsar timing and either total power or frequency switched spectral line observations.

2.1.4.5 GUPPI

The Green Bank Ultimate Pulsar Processing Instrument (GUPPI) has one hardware mode and many software modes. GUPPI can be used with any receiver with the exception of MUSTANG. Only one polarization would be available for the Ka-band receiver. GUPPI uses 8-bit sampling to dramatically improve upon the dynamic range and RFI resistance of the SP. Currently GUPPI can use bandwidths of 100, 200 and 800 MHz with 2 polarizations and full stokes parameters. The minimum integration time is 40.96μ s using 2048 channels and an 800 MHz bandwidth. See the introduction to GUPPI in Chapter 14.

2.1.4.6 MUSTANG

See the introduction to MUSTANG in Chapter 16.

2.1.4.7 Zpectrometer

The Zpectrometer is a wide-band spectrometer used only with the KA-band receiver. It covers a frequency range of 25.6-36.1 GHz with partial and degraded performance available up to 37.7 GHz. It contains four independant sub-bands with a few channels of overlap between each adjacent pair. The standard frequency resolution is 24 MHz, with frequency resolution constant across all bands. With a 34% fractional bandwidth the velocity resolution changes noticably from one end of the band to the other. It is possible to push the frequency resolution to 18 MHz at the cost of modest increases in calibration and processing time.

For more information, refer to Chapter 17.

2.1.4.8 VLBI

The GBT supports VLB observations with a Mark5 VLBA recorder. This recorder can also be used in a "single-dish" mode to make high time–resolution observations.

For more information, consult the web page:
/http://www.gb.nrao.edu/ fghigo/gbtdoc/vlbinfo.html.

2.1.4.9 Radar

Planetary radar observations are supported by a "portable fast sampler", sampling at 2-4 bits at rates up to 20 MHz. This backend is used and maintained by groups at Arecibo, Cornell, and CalTech. Those interested in planetary radar should consult people in those groups. As an introduction, refer to the web page:
https://safe.nrao.edu/wiki/bin/view/GB/Observing/RadarObserverAdvice.

2.1.5 Polarization Measurements

Measurement of Polarization and Stokes parameters is possible using the GBT Spectrometer (ACS), the Spectral Processor (SP), and GUPPI. This is an "expert user" mode; users should contact their GBT "friend, and consult Tim Robishaw or Carl Heiles for advice. As an introduction, refer to "A Heuristic Introduction to Radioastronomical Polarization," by C. Heiles, ASP Conference Series Vol 278, 2002.

2.2 Computing Facilities

Workstations are available for visitors in Room 105 in the Jansky Lab. Most are Unix stations, and there is also a Windows machine. Laptop connections are provided there and in several locations around the Observatory, including some rooms in the Residence Hall.

Visitors should obtain a login account on the Green Bank system before arriving. Accounts may be requested from Wolfgang Baudler (wbaudler@nrao.edu) or Chris Clark (cclark@nrao.edu). Any problems with connecting a personal computer to our network should be referred to these same two gentlemen.

2.3 The GBT Observing Process

The following list summarizes the general flow of how GBT observing proceeds. By the time you are reading this document you should have already been through several of the steps.

Steps for Proposing and Obtaining Time on the GBT

Step 1 You get a brilliant idea for an observation. You research which telescopes would be best suited for the observations and find that the GBT is the instrument for you.

Step 2 You read through the Proposers Guide (http://www.gb.nrao.edu/gbtprops/man/GBTpg/GBTpg_tf.html) and determine which receivers and backends will be best for you to use. You then write a proposal to use the GBT using the Proposal Submission Tool (https://my.nrao.edu). Proposal submission dates will change from a trimester to a semester system as of February 2011; the next submission deadlines are the first of October 2010 and of February 2011. After that the due dates will be the first day of November and August (or the next business day if these are on a weekend or a holiday).

Step 3 Your proposal is sent to referees for ranking and given a technical review by NRAO staff. The Telescope Allocation Committee (TAC) uses these reviews to decide which proposals can be scheduled given the resources available in a trimester.

Step 4 If your proposal is not granted GBT time then you go back to step 2. If your proposal is granted time on the GBT by the TAC then your observations are scheduled.

Steps for Observing With the GBT

Step 5 Before you observe, you need to prepare for your observations (see Chapter 12). You will be assigned a scientific contact person (GBT "friend"), whom you should contact well in advance of your observing to determine optimum dates for a visit and ensure the telescope and hardware will be available for the project while you are on site. Your "friend" will help you develop an appropriate observing strategy for your proposal (see Chapter 6). They will also help you with any technical questions, dealing with Radio Frequency Interference (RFI) (see Chapter 8), etc. At this time you should develop your Observing Scripts (see Chapter 5).

Step 6 If you are an experienced GBT observer, you can observe remotely (see Chapter 11. If you are new to the GBT, you must plan to spend at least a week and preferably two weeks at the site to ensure appropriate weather conditions for the observations. After hands-on experience with observing you will qualify for remote observing.

Step 7 You will travel to Green Bank for your observations (see Chapter 12) or if you are an experienced GBT observer you can observe remotely. You should arrive in Green Bank at least one business day before your observations. This will allow you to meet with the contact scientist and also with the scientific staff person who will be "on call" during your observations (these might be different people).

Step 8 It is very likely that your observations will be dynamically scheduled. See Chapter 9 for details on how dynamical scheduling is done with the GBTẆhen your project is scheduled, you will receive an e-mail notification indicating the exact time the observing session will start. Notifications go to the project PI and all others designated as observers on the project. Thus you should have prepared your scripts and be ready to observe with 24-36 hours notice. If there are periods of time or dates when you cannot observe, you should indicate these as "blackout dates" in the DSS web page https://dss.gb.nrao.edu.

Step 9 If you are present in Green Bank, go to the control room shortly before your observations begin. You can log into one of the computers and bring up any programs that you need so that you are prepared when you observation time begins.

If you are observing remotely you should contact the GBT operator (304-456-2341 or 304-456-2346) about 30 minutes before your observations. You should give the operator your contact information (phone numbers, emails) so that they can contact you during the observations if necessary. You will also need to let the operator know what computer you will be using during your observations. At this time you will begin to open a Virtual Network Computer (VNC) session that you will use for the remote observations. Starting this early will allow for any problems encountered while preparing to observe remotely to be solved before the observations are to begin. You can find information about GBT remote observing policies at
https://safe.nrao.edu/wiki/bin/view/GB/Observing/GbtObservingPolicies
and you can find information about opening a VNC session at
http://www.gb.nrao.edu/gbt/remoteobserving.shtml.

Step 10 The operator on duty will handle several tasks for you at the beginning of your observations. They will "put you in the gateway" (give you security access) so that you can control the GBT. They will also get the correct receiver into the focus position of the GBT, get the antenna motor drives ready for movement, place the correct pointing models into the system, and set the GBT's active surface (AS) into the proper state. The operator is there to take care of all safety issues concerning the GBT.

Step 11 Now you are ready to observe. You will use the Astronomer's Integrated Desktop (Astrid) (see Chapter 3) to perform your observations by submitting an Observing Script (see Chapter 5). The steps you will take in observing are generally:

A Configure the receiver, Intermediate Frequency system (IF), and backend to the desired states. The parameters used to determine these states are known as the "configuration" and the act of setting these states is known as "configuring."

B Slew to your source.

C Balance the IF. In this step you adjust amplifier and attenuator settings in the IF to ensure that all components operate within their linear regime.

D Execute your observations using one of the Observing Script Scan Types (see Chapter 5).

E If problems develop let the operator know and they will contact the on-call support scientist for assistance.

Step 12 Once you are done observing you should close Astrid, log out of the computer you were using, and leave the control room.

Step 13 During and after your observing run you will reduce your data. You will generally use GBTIDL for data reduction of spectral line data. This can be done either at NRAO or at your home institution. Only rudimentary continuum data reduction support is available for the GBT at this time and you should contact your contact scientist for more information. A pulsar data reduction package, **PRESTO**, is available from Scott Ransom.

Use one of the data reduction machines, not the workstations used for running the observations. Refer to http://www.gb.nrao.edu/pubcomputing/data-reduction.shtml for a list of data reduction machines at Green Bank.

Step 14 Once you are ready to leave Green Bank you will want to take home your data (see Chapter 13).

Step 15 Finally you will want to write your Nobel Prize winning paper. NRAO can help you with your page charges (see Chapter 13). You should also notify your scientific contact person of your paper to help the Observatory keep track of how successful all observing projects have been.

Chapter 3

Introduction To Astrid

3.1 What Is Astrid?

The Astronomer's Integrated Desktop (Astrid) is a single, unified workspace that incorporates the suite of applications that can be used with the GBT. Astrid provides a single interface from which the observer can create, execute and monitor observations with the GBT. Some of the features of Astrid are:

- Creates and executes Scheduling Blocks (which perform astronomical observations) from Observing Scripts.
- Provides a real time display of GBT data.
- Provides an update on the status of the GBT.
- Provides an area to edit Observing Scripts. They may be edited offline and saved before observing.
- Allows a second observer to monitor an observation that is in progress.

Astrid is a GUI (Graphical User's Interface) that is built from python code. Many aspects of Observing Scripts will thus contain python commands along with specialized functions designed specifically for the GBT. Astrid brings together many applications into a single, unified GUI. The GUI places each application into its own tab window. Applications available in Astrid are:

Observation Management The GBT Monitor and Control (M&C) systems can roughly be thought of as a group of programs - one for each hardware device - and a master program, the Scan Coordinator. Astrid interfaces with the Observing Management Application in order to run Observing Scripts on the GBT.

Data Display Astrid provides a real time data display by connecting to the GBT FITS Monitor (GFM). This allows the automatic processing of pointing and focus scans that can immediately update the GBT M&C system with the determined corrections. GFM can show raw, uncalibrated continuum data as a function of time. It can also show raw, uncalibrated bandpasses for spectral line data.

GBT Status Astrid also provides a screen that provides information on the real time status of the GBT.This provides meta–information such as the LST, UTC, observer, project ID, etc., information on the antenna such as current position, etc. and information on the current scan and IF setup.

Python Editor This application is a windows–like text editor that features syntax highlighting for Python code. This is the editor that is part of the Astrid Edit Tab (see § 3.3.1) where you can edit, copy, and save Observing Scripts.

Command Line Interface Astrid also has a command line interface that allows some specialized interaction with the M&C system. This is to be used by expert observers and is currently only used during some pulsar observations.

3.2 How To Start Astrid

3.2.1 Running Astrid

From any Linux computer in Green Bank just type "astrid" to start the program. It usually takes Astrid between 10-20 seconds to launch from the command line. The first thing you will see is the astrid "splash screen" which is shown in Figure 3.1.

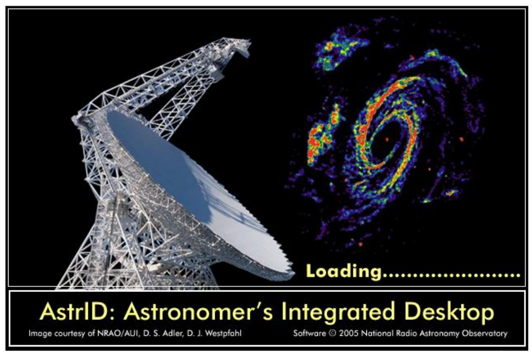


Figure 3.1: The Astrid splash screen.

Astrid will now ask you what mode you would like to operate in. This will be done via a "pop-up" window which is shown in Figure 3.2.

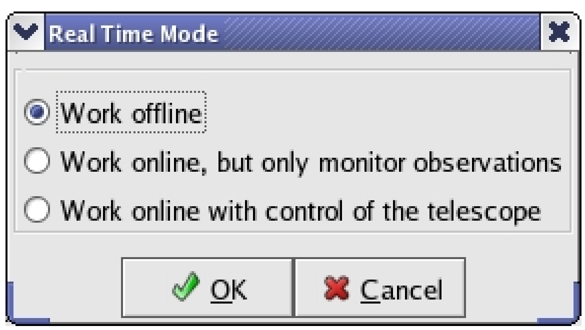


Figure 3.2: When you start Astrid you will see this "pop-up" window which will allow you to choose which mode of Astrid that you would like to use.

There are three different modes in which you can run Astrid:

Work online with control of the telescope. In this mode you can run Observing Scripts and you are in full control of the GBT observations. In this mode you can also see near–real–time data from the GBT.

Work online, but only monitor observations. In this mode you can actively watch what is currently happening in Astrid for the current GBT observations. However, you will not be allowed to submit any Observing Scripts for execution or to affect the current observing in any manner. You will be able to see near–real–time data from the GBT.

Work offline. In offline mode you can edit and syntactically validate your Observing Scripts. You can also use the data display part of Astrid to look at previously obtained data.

Once you see the screen shown in Figure 3.3 you have successfully started Astrid.

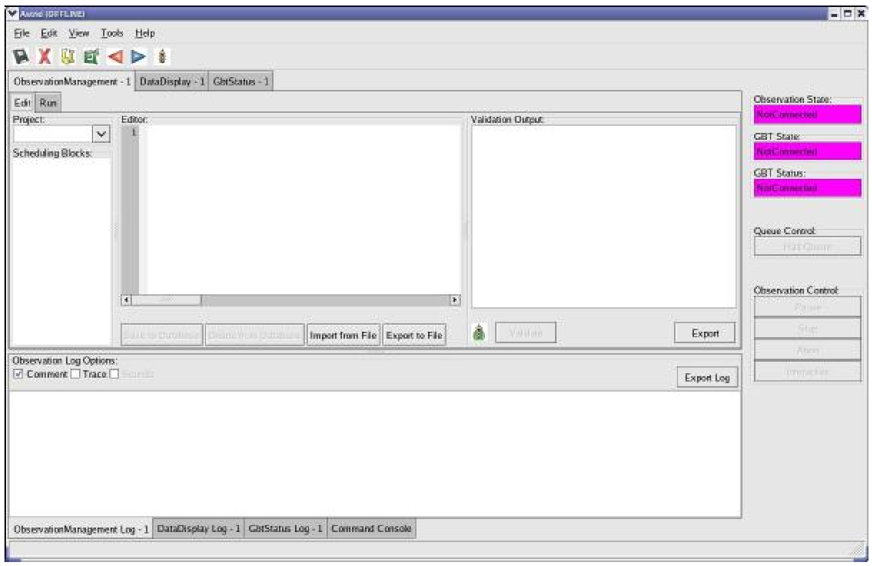


Figure 3.3: The initial Astrid screen upon startup.

3.2.2 Astrid GUI Composition

The Astrid GUI layout consists of drop–down menus, a toolbar, application component tabs, an observational status section, a queue control section, an observation control section, the application component, and a log window. These are shown in Figure 3.4.

3.2.2.1 Drop–down Menus

In the top, left hand side of the Astrid GUI you will find the drop–down menus. The contents of the drop–down menus change according to which Application is currently being displayed on the Astrid GUI. We will not discuss all of the options under the drop-down menus in this document but we will provide some highlights.

File

Under the File drop–down menu you will find the "New Window" option. Under the "New Window" side–menu option you will be able to launch Applications within the Astrid GUI or in an independent GUI. The "Close Window" option will allow you to close the currently displayed Application in the

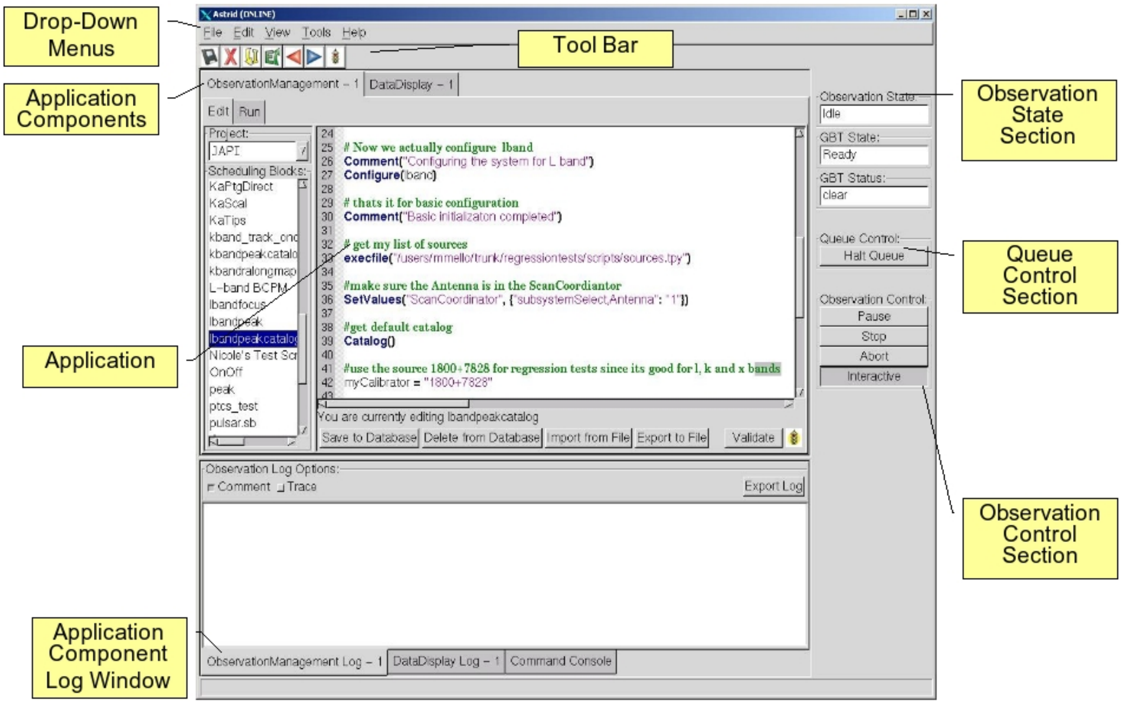


Figure 3.4: The different components on the Astrid GUI.

Astrid GUI. The "Real Time Mode..." option allows you to change between the operational modes of Astrid.

Edit

Under the Edit drop–down menu you will find standard "Windows" cut and paste options.

View

The options under the View drop–down menu allow you to turn "on" or "off" the display of the drop–down menus and the toolbar. You can also toggle between having the Astrid GUI taking up the full screen or not.

Tools

The Tools drop–down menu is only active when the Data Display Application is being shown in the Astrid GUI. You can zoom and pan within the data plots shown in the Data Display Application using the Tools drop–down menu. You can also change the "Fitting Heuristics" used during the reduction of Pointing and Focus observations (see § 4.1.3.2).

Help

Under the Help drop–down menu you can bring up documentation for some but not all Applications.

3.2.2.2 Toolbar

The Toolbar is located just under the Drop–down Menus near the top of the Astrid GUI. The contents of the Toolbar change depending on which Application is being displayed in the Astrid GUI. The Toolbar options are a subset of commonly used options from the Drop–down Menus.

When you leave the mouse situated over one of the Toolbar buttons for a few seconds a "pop-up" will appear that tells you what action the Toolbar button will invoke.

3.2.2.3 Application Component Tabs

The Application Component Tabs are located under the Drop-down menus and the Toolbar and at the very bottom of the Astrid GUI. There is a tab for each Application that Astrid is currently running. By clicking on the tabs you can switch the Astrid GUI so that it shows the contents of the selected Application.

3.2.2.4 Application

This comprises the majority of the space within the Astrid GUI. This shows the contents of the Application selected by the Application Component Tabs.

3.2.2.5 Log Window

The Log Window is located in the lower portion of the Astrid GUI underneath the Application display area. It shows the log information for the currently selected Applications. Note that each Application has its own logs. Some Applications allow the contents of the logs to be saved to an external file.

3.2.2.6 Observational Status

The Observation Status area is located in the upper right corner of the Astrid GUI. This provides information on whether or not Astrid is talking with the Monitor and Control (M&C) system, as well as the current state of the GBT and Status of the GBT.

Observation State

The Observation State indicates Astrid's state. The observation state is either:

- Not Connected
- Idle
- SB Executing
- SB Paused

If Astrid is not communicating with the M&C system (such as in its "offline" mode) then you will see "Not Connected". If Astrid is communicating with the M&C system and there isn't a Scheduling Block (SB) being executed then you will see "Idle" and if a Scheduling Block is running (or has been paused) then you will see "SB Executing" ("SB Paused").

GBT State

The GBT State indicates the state of the M&C system. The GBT state is either:

- Not In Service
- Not Connected
- Unknown
- Ready
- Activating
- Committed
- Running
- Stopping
- Aborting

If the M&C system is not working properly you will see "Not In Service" or "Not Connected." "Unknown" indicates that the M&C system is working but does not know the state of any of the hardware devices. You will see the state be "Ready" when the GBT is not doing anything. It will be "Activating" or "Committed" when the GBT is preparing to perform an observation, etc. While taking data during a scan the state will be "Running". At the end of a scan you will see the state become "Stopping." If the scan is ended for any abnormal reason the state will be "Aborting."

GBT Status

The GBT status gives the error status of the M&C system. The GBT status is either:

- Not Connected
- Unknown
- Clear
- Info
- Notice
- Warning
- Error
- Fault
- Fatal

If the M&C system is not communicating properly with the hardware the status can be "Unknown" or "Not Connected." If the status is "Clear", "Info", or "Notice" then there are no significant problems with the GBT. If "Warning" then it is worth asking the Operator what the problem is, but it may not affect observation quality. If the status is "Error" then there is potentially something wrong that may need attention. If the status is "Fault" or "Fatal" then something has definitely gone wrong with the observations.

3.2.2.7 Queue Control

The Halt Queue Control button, located in the middle–right of the Astrid GUI, gives you some control over the execution of Scheduling Blocks. If this button is not activated then Scheduling Blocks in the Run queue will continue to be executed in order. If this button is activated it will finish the current Scheduling Block but will not allow the next Scheduling Block in the Run Queue to execute until the button is returned to its default “off” state.

3.2.2.8 Observation Control

The Observation Control area is located in the lower-right of the Astrid GUI. The Observation Control buttons gives the observer control of the GBT during the execution of a Scheduling Block.

The Pause button, when activated, will stop the execution of the current Scheduling Block/Observing Script when the next line of the Observing Script is encountered.

The Stop button will stop the current scan at the end of the next integration time. This is a nice, gentle way to stop a scan.

The Abort button stops the current scan immediately. This may lead to corrupted data.

The Interactive button, when selected, will cause Astrid to automatically answer any pop–up query. Astrid will always choose what it deems to be the safest answer. This is useful when you have to leave the control for an extended period of time (such as when you go to the cafeteria to eat, etc.).

3.2.3 Resizing Astrid Display Areas

It is possible to resize some of the display areas within Astrid. If you put the mouse over the bar separating two display areas you will get a double-arrowed resize cursor. If you then hold down the left–mouse button you can use the mouse to move the border and resize the display areas.

3.2.4 Changing Modes Within Astrid

Observers should login and setup for their observations before their scheduled time begins. Under these circumstances the observer will have already brought up Astrid in the “offline” mode or the “Work online, but only monitor observations” mode. When the observer’s scheduled time on the GBT begins, the Astrid mode can be changed without having to exit out of Astrid. This is done with the following steps:

Confirm with the Operator that you can go online.

Step 1 Click on File in the drop–down menus section.

Step 2 Click on “Real time mode...” in the drop–down menu.

Step 3 The pop-up window shown in Figure 3.2 appears.

Step 4 Click the radio button for the desired mode in the pop–up window.

Step 5 Click the OK button in the pop–up window.

If problems occur, inform the Operator, who will clear them up.

At the end of an observing session the observer should change the Astrid mode to “offline” immediately after their observing session ends.

3.3 The Observing Management Tab

The Observation Management Application consists of two sub–GUIs, the Edit Tab and the Run Tab (see Figures 3.5 and 3.6). In the Edit Tab you can create, load, save, and edit Observing Scripts. You can also Validate that the syntax is correct. The Run Tab is where you will execute GBT observations.

3.3.1 The Edit Tab

The Edit Tab has three major areas: a list of Project IDs, Observing Scripts that have been saved into the Astrid database for that project, an editor, a Validation area, and a log summarizing the observations. This is shown in Figure 3.5. Chapter 5 covers the contents and creation of Observing Scripts.

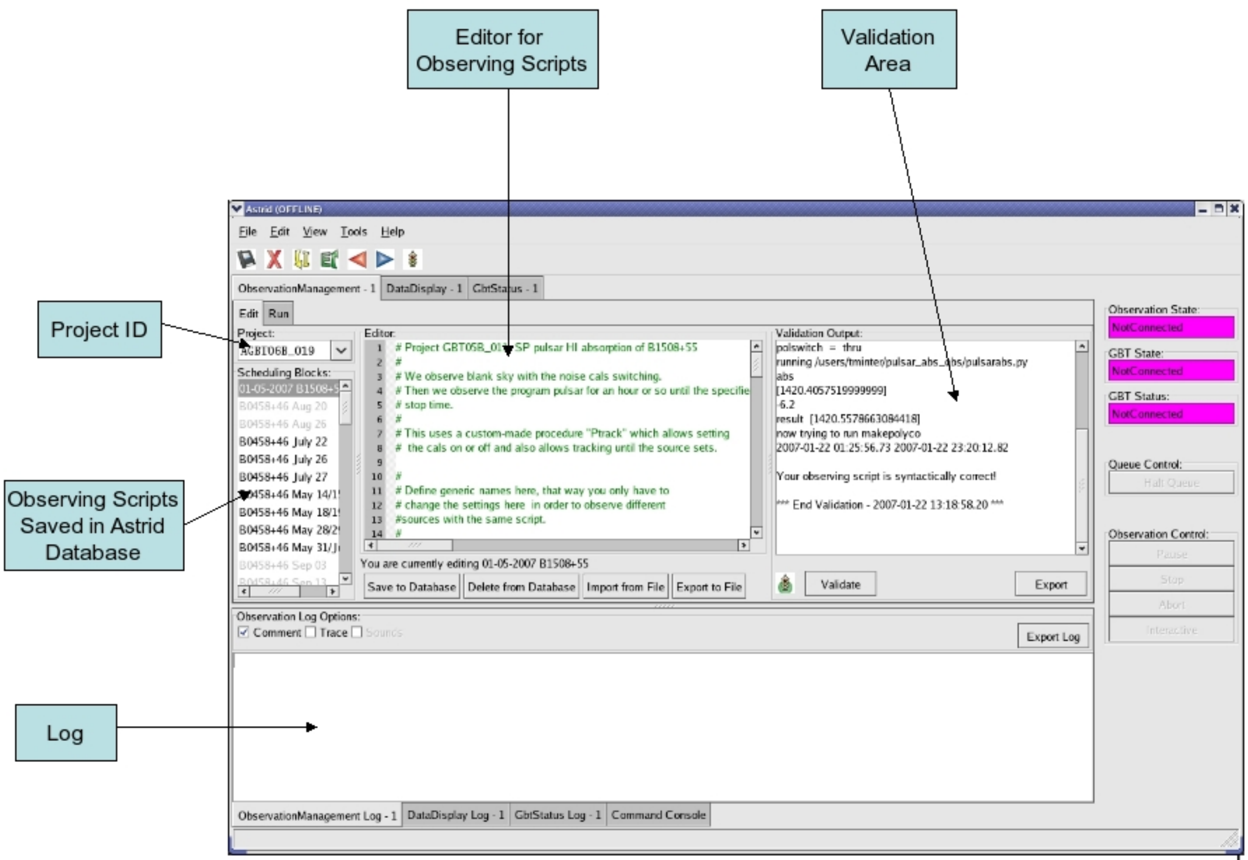


Figure 3.5: The Astrid Observation Management/Edit Tab.

3.3.1.1 Project ID and List of Observing Scripts

When you first go to the Edit Tab you will select your project name using the pull–down menu in the upper left part of the Edit Tab under the window labeled "Project." You may also just type in the project code. Your project name is the code that your GBT proposal was given. After doing this you will see in the window labeled “Scheduling Blocks” a list of Observing Scripts, if any, that have been previously saved into the Astrid database.

All of the saved Observing Scripts for a given project will show up in the “Scheduling Blocks” section of the Edit Tab. If an Observing Script has been Validated (i.e. it is syntactically correct) then it will appear in bold–face type. This means that it can be executed. If the script has been saved but is syntactically incorrect it will appear in lighter–faced type.

3.3.1.2 Editor

You can use the Editor to create or modify an Observing Script within Astrid. Standard Windows functions like Ctrl–X (to delete selected text), Ctrl–C (to copy selected text), and Crtl–V (to paste selected text) can be used in the editor. The editor lists the line number on the left hand side of the editor and uses color coding for the type of lines within the Observing Script. Green characters are for commented characters, black is for variables and standard python commands/syntax , purple/magenta is for strings, and dark blue is for function names. The contents of loops, if statements, etc. that are normally indented in python are also indicated on the left hand side of the editor. The start of a loop, for example, is indicated by the ⊖ symbol and the contents of the loop are within a black line that connects to the ⊖ symbol.

The editor also has four operational buttons. These are:

Save to Database This button will check the validation of the current Observing Script and then save it to the Astrid database. A pop–up window will notify you if the Observing Script did not pass Validation. A second pop-up window will allow you to set the name that the Observing Script will be saved under in the Astrid database.

Delete from Database This button will delete the currently selected Observing Script from the Astrid database.

Import from File This button will allow you to load an Observing Script from a file on disk.

Export to File This button will allow you to save the edited Observing Script displayed in the editor to a file on a disk. This does not save the Observing Script into the Astrid database.

The first time you select either of the “Import from File” or “Export to File” buttons you will have a pop–up window that lets you select the default directory to use. After selecting the default directory you will get a second pop–up window that shows the contents of the default directory so that you can select or set the disk file name to load from or export too.

3.3.1.3 Adding Observing Scripts to the Database and Editing Them

We will first describe how to add an Observing Script to the “Scheduling Block” list (i.e. database) and then we will describe how to manipulate and edit Observing Scripts in the list.

Saving an Observing Script to the Database

If you have already created an Observing Script outside of Astrid, you should go to the Edit Tab in Astrid and then use the “Import from File” button to load your Observing Script into the Editor. Otherwise you can just create your Observing Script in the Editor. To save the Observing Script into the Astrid database you just need to hit the “Save to Database” button. This will run a validation check (see § 3.3.1.4) on your script and then a pop–up window will appear which allows you to specify the name which you would like to use in the list for your Obsering Script.

Selecting an Observing Script

If you perform a single click on any Observing Script in the “Scheduling Block” list, the contents of the selected observing script will appear in the Editor. The selected Observing Script will be highlighted with a blue background.

Mouse–button Actions on the selected Observing Script

If you perform a right mouse button click on the selected Observing Script a pop–up window will appear that will let you rename, create a copy or save the Observing Script to the Astrid database. You can also delete the Observing Script from the Astrid database.

You may also rename the Observing script if you perform a left mouse button double click on the script name in the list.

3.3.1.4 Validator

The Validation area is where you can check that the currently selected Observing Script is syntactically correct. This does not guarantee that the script will do exactly what you want it to do. For example, it can not check that you have the correct coordinates for your source. You will also see error messages, notices and warnings from the Validation in this area.

Before an Observing Script can be run within Astrid it first must pass Validation. To Validate a script without saving it you can just hit the Validate button. An Observing Script automatically undergoes a validation check when you hit the “Save to Database” button in the editor. Any messages, etc. from the validation will appear in the “Validation Output” test area. You can export these messages to a file on disk by hitting the “Export” button in the validation area.

The state of an Observing Script’s validation is shown by the stop–light. If the script has never been validated or has been changed since the last validation the stop–light will have the yellow light on. If the Observing Script fails validation the stop–light will turn red, while it will turn green if the Observing Script passes validation.

Note that ”for” loops with many repeats can take an extended amount of time to validate since the Validator will go through each step in the loop. Also be careful of infinite loops in the validation process. Use of time functions such as Now() (see Chapter 5) always return “None” in the validation.

3.3.2 The Run Tab

The Run Tab is shown in Figure 3.6. In the Run Tab you will queue up Observing Scripts to perform the various observations that you desire to make. The Run Tab has five components. Across the top of the Run Tab you enter information that will be put into the headers associated with the observations. On the left is a list of Observing Scripts that you can execute. On the right are the Run Queue which holds Scheduling Blocks that are to be executed in the future and the Session History which shows which Scheduling Blocks/Observing Scripts have previously been executed. At the bottom is the observing log.

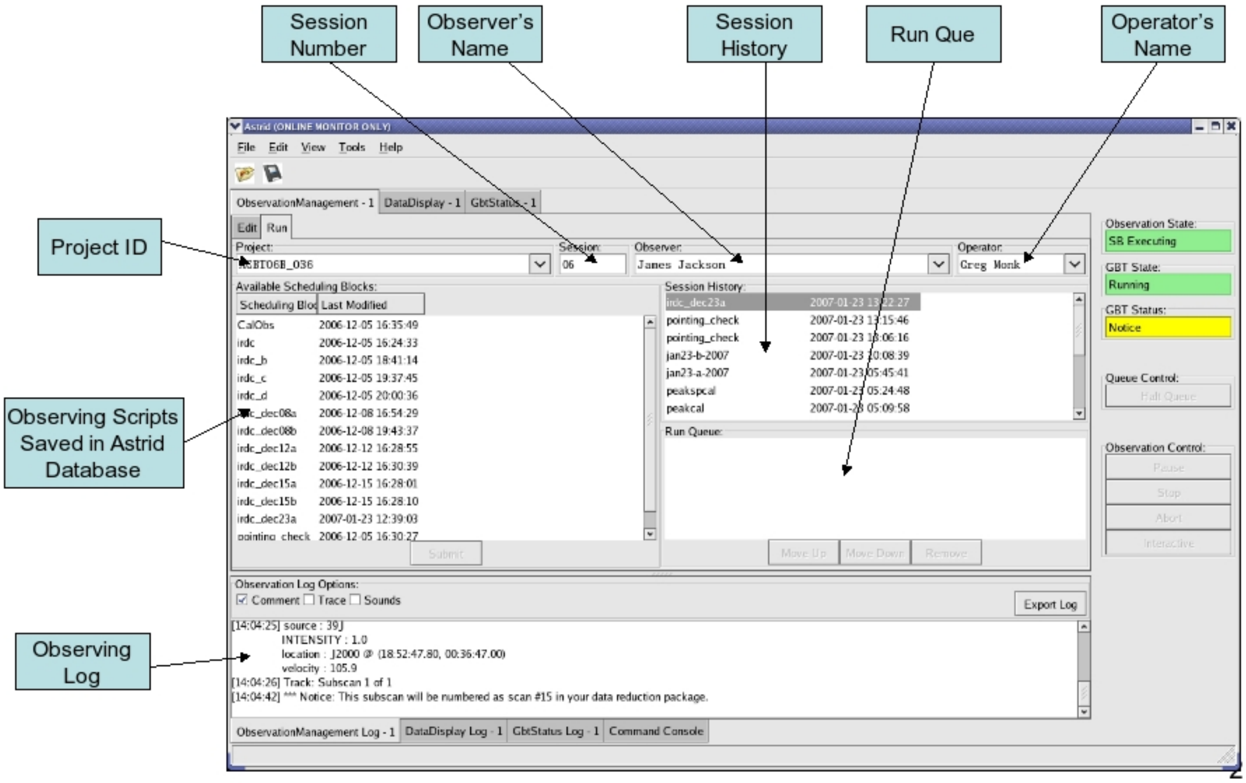


Figure 3.6: The Astrid Observation Management/Run Tab.

3.3.2.1 Header Information Area

The meta–data consists of the project, the session, the observer's name and the operator's name. All fields must have entries before an Observing Script can be executed.

Project

Just as in the Edit Tab you use the drop–down menu to select your Project ID. If your project is not listed, ask your GBT "friend" or the telescope Operator to add it to the database.

Session

A session is a contiguous amount of time (a block of time) for which the project is scheduled to be on the telescope. Each time a project begins observing for a new block of time it should have a new session number. The session number is usually determined by Astrid and automatically entered. However, there are cases (such as Astrid crashing) where the session number could become incorrect. You can type in the correct session number if needed.

Note that a "Session" in Astrid is equivalent to an "observing period" in the lingo of the Dynamic Scheduling System (DSS). "Session" has a different meaning in the DSS.

Observer's Name

This is a drop–down list where you choose the observer's name. Only the PI's on a project are guaranteed to have their name in this list. If your name is not listed, ask your GBT "friend" or the telescope operator to add it.

Operator's Name

This is a drop–down list from which you pick the current operator's name at the beginning of your observations.

3.3.2.2 Submitting An Observing Script to the Run Queue

In order to execute an Observing Script you must:

Step 1 Select the Observation Management Tab.

Step 2 Select the Run Tab.

Step 3 Make sure that the header information fields all have entries.

Step 4 Select the Observing Script you wish to execute from the list of available Observing Scripts.

Step 5 Hit the Submit button below the list of Observing Scripts.

Your Observing Script is then automatically combined with the header information to produce a Scheduling Block that is then sent to the Run Queue. Note that double-clicking on an Observing Script is the same as selecting the Observing Script and then hitting the Submit button.

3.3.2.3 The Run Queue and Session History

When an Observing Script is submitted for execution it is first sent to the Run Queue. This contains a list of submitted Observing Scripts that will be sequentially executed in the future.

When an Observing Script begins execution it is moved to the Session History list. So the Session History list contains the currently executing Observing Script on the first line and all previously executed Observing Scripts that have been run while the current instance of Astrid has been running on subsequent lines.

If there are not any Scheduling Blocks in the Run Queue when a new Observing Script is submitted for execution it may appear that the Observing Script just shows up in the Session History. However it has indeed gone through the Run Queue - albeit very quickly.

3.3.2.4 The Observing Log

The observing log is always visible at the bottom of the Observation Management Tab. It shows information from the execution of Observing Scripts. The observing log can be saved to a disk file by hitting the Export button that is just above the top right corner of the log display area.

3.4 The Data Display Tab

The Data Display Tab provides a real time display of your GBT data. The Data Display Tab will be discussed in Chapter 4.

3.5 The GbtStatus Tab

The GbtStatus Tab displays various GBT specific parameters, sampled values and computed values. Special care was taken to promote its use for remote observing. Examples of how the GBT Status Display appears in Astrid are shown in Figures 3.7 and 3.8.

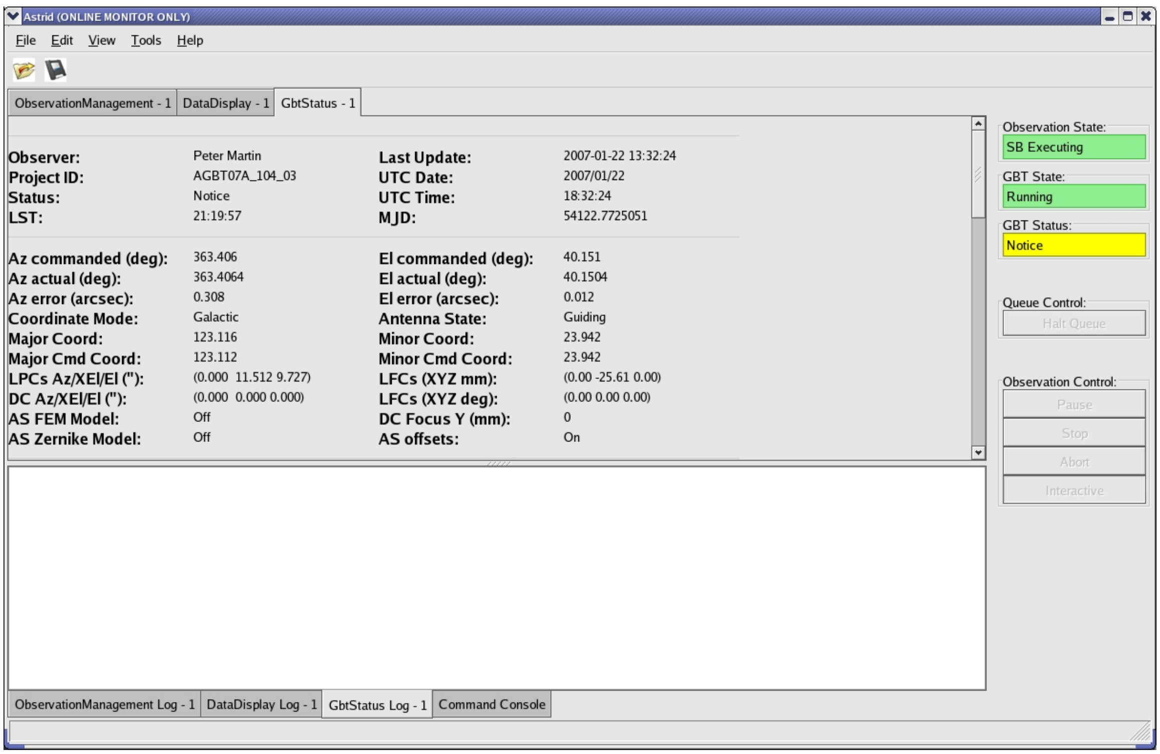


Figure 3.7: The Astrid Gbt Status Tab showing the top portion of the status. To see the rest of the status screen you will need to use the scroll bar.

The default status screen displays all of the currently supported items of the gbtstatus program. These are:

Duration The scan length in seconds.

Remaining The time remaining in the scan.

Scan A derived field: composed of the scan number and PROCNAME, PROCSIZE and PROCSEQN keywords from the GBT Observing (GO) FITS file.

Scan Start Time If scan has started it is the UTC scan start time - if the scan has not started, then it is the countdown until the start of scan.

On Source “Yes” or displays a countdown until the antenna is on source.

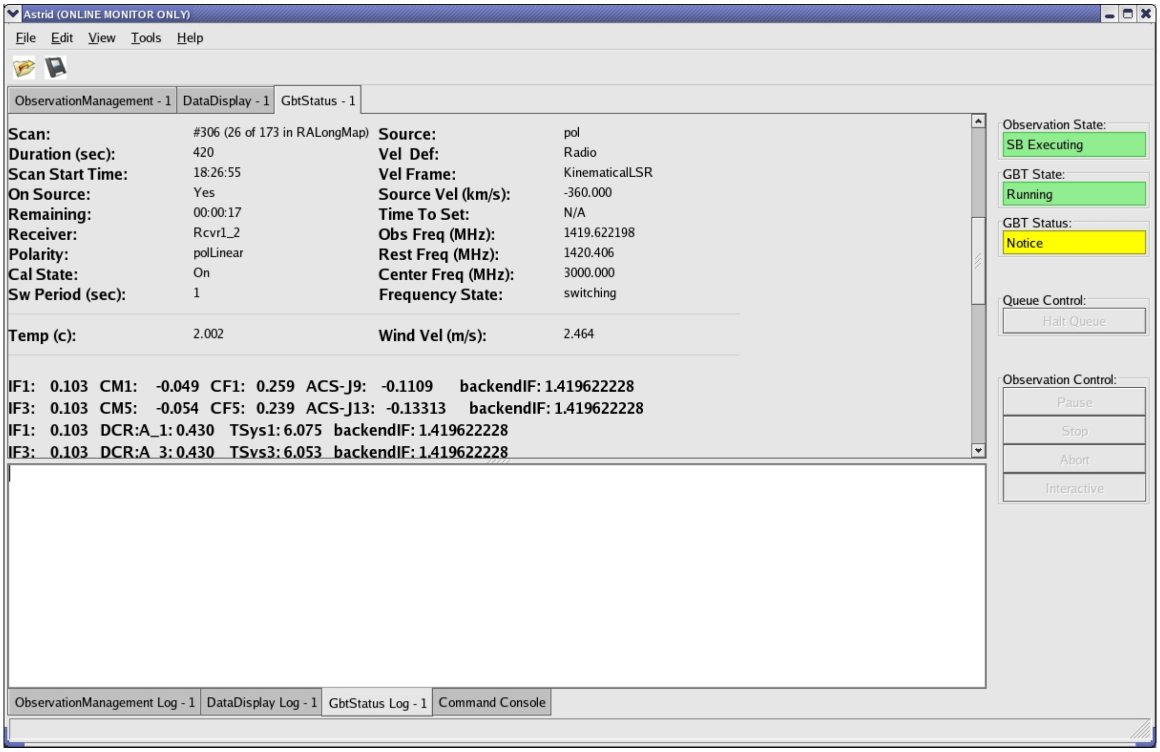


Figure 3.8: The Astrid Gbt Status Tab showing the bottom portion of the status. To see the rest of the status screen you will need to use the scroll bar.

Source The source name.

Source Vel The source velocity ($\mathrm{km\,s^{-1}}$).

Observer The observer name as recorded in the FITS file.

Last Update The local time when the database was last updated.

Operator This field is currently not working.

UTC Time The UTC time of the last update.

UTC Date The UTC date of the last update.

Project ID The data directory of the FITS files.

LST The LST of the last update.

MJD The Modified Julian Date (MJD).

Status The status of the GBT. The GBT status is either:

- Not Connected
- Unknown
- Clear
- Info
- Warning
- Error

- Fault
- Fatal

If the M&C system is not communicating properly with the hardware the status can be “Unknown” or “Not Connected.” If the status is “Clear”, “Info” or “Warning” then there are no significant problems with the GBT. If the status is “Error” then there is potentially something wrong that may need attention. If the status is “Fault” or “Fatal” then something has definitely gone wrong with the observations.

Time To Set The time till the current source sets.

Receiver The receiver being used.

Polarity The receiver polarity.

Obs Freq The observational frequency in MHz.

Rest Freq The rest frequency in MHz.

Cal State ’ON’ if the noise diode is firing during the scan.

Sw Period The switching period in seconds.

Center Freq The center frequency in MHz.

Vel Def The velocity definition.

Vel Frame The velocity frame.

Frequency State The switching type: total power or frequency switching.

Coordinate Mode The coordinate mode.

Major and Minor Coord The telescope position in the current Coordinate Mode.

Major and Minor Cmd Coord The telescope position in the current commanded Coordinate Mode.

Antenna State The telescope state. The most common antenna states are:

- Disconnected
- Dormant
- Stopped
- Guiding
- Tracking
- Slewing

If the antenna software is not running the state will be “Disconnected.” If the antenna software is running but with its control of the antenna turned off then the state is “Dormant.” If the antenna is not moving then the state will be “Stopped.” If the antenna is moving and data are being taken then the state is “Guiding” and if data are not being taken the state is “Tracking.” If the antenna is moving to a new commanded position the state is “Slewing.”

Az commanded The commanded azimuth position of the telescope in degrees.

El commanded The commanded el position of the telescope in degrees.

Az actual The actual azimuth position of the telescope in degrees.

El actual The actual elevation position of the telescope in degrees.

Az error The difference between the commanded and the actual azimuth position of the telescope in arc-seconds. This value does not contain a cos (el) correction.

El error The difference between the commanded and the actual elevation position of the telescope in arc-seconds.

LPCs Az/XEl/El The Local Pointing Correction (LPC) offsets in arc-seconds.

DC Az/XEl/El The Dynamic Corrections values in arc-seconds. The GBT has temperature sensors attached at various points on the backup structure and the feed-arm. These are used in a dynamic model for how the GBT flexes with changing temperatures. This model is used to correct for pointing and focus changes that occur from this flexing.

AS FEM Model The state of the Finite Element Model (FEM) correction for the Active Surface (AS). The FEM predicts how the surface changes due to gravitional flexure versus the elevation angle.

AS Zernike Model The state of the AS Zernike model correction model. The Zernike model is a set of Zernike polynomial coefficients determined from Out–Of–Focus holography that improve the shape of the AS versus the elevation angle.

AS Offsets The state of the AS zero offsets. The zero offsets are the default positions for the AS. This should always be "On" if the AS is being used.

LFCs (XYZ mm) The Local Focus corrections for the offset focus position in millimeters. This value is determined from a Focus observation (see Chapter 5).

LFCs (XYZ deg) The Local Focus Tilt offset in degrees.

DC Focus Y (mm) The Dynamic Corrections Y subreflector offset in millimeters. The GBT has temperature sensors attached at various points on the backup structure and the feed-arm. These are used in a dynamic model for how the GBT flexes with changing temperatures. This model is used to correct for pointing and focus changes that occur from this flexing.

Temp The current air temperature in Celsius.

Wind Vel The current wind velocity in $\mathrm{m\,s^{-1}}$.

The Intermediate Frequency paths (IF path) in use are always displayed in the last section of the GBT status screen. An example screen is shown in Figure 3.8. Each line represents the IF path for a single polarization path from the IF Rack to the backend. Each line contains only the devices in use for the listed path. A path may include a subset of the devices and values listed below. For IF paths containing the Spectrometer, the path will be displayed in "RED" when the duty cycle varies 2.5 db from the optimal value. Details on information presented for the IF paths can be found in Appendix A.

Chapter 4

Near–Real–Time Data and Status Displays

4.1 The Astrid Data Display Tab

The Data Display Tab provides a real time display of your GBT data so that you can check that you are getting valid data. The Data Display is actually running an application called GBT Fits Monitor (GFM). This application provides sub-scan-based display and analysis of GBT data, either in real-time as the data is being collected, or in an offline mode where it can be used to simply step through the sub-scans from an observation.

Some of the features of GFM are:

- Automatically detects the type of scan/observing (e.g. Focus, Pointing, Spectral Line) and calls the relevant analysis modules.
- Knows how to handle groups of scans properly, for example, the four scans within a pointing observation.
- Supports multi-beam/dual-polarization, multiple IF s, and multiple phases.
- Supports dynamic focus corrections.
- Graphics export to Portable Network Graphics (PNG) and Postscript (PS and EPS) formats is supported.
- Playback feature allows you to quickly review the plots within a range of sub-scans.

4.1.1 Working Online

If you are using Astrid's "online" or "monitor" mode and have selected the "DataDisplay" tab, then the data display will update as new data are obtained. Continuum and Spectral Line data are only updated when these displays are being viewed. Pointing and Focus data are always automatically updated whether or not their displays are being shown or not. The list of scans will always automatically update.

4.1.2 Working Offline

You can look at data that have already been taken with the GBT by running Astrid in its "offline" mode. To view data in this mode you need to follow these steps:

Step 1 Change the Astrid mode to "offline" (see § 3.2.4).

Step 2 Select the Data Display Tab.

Step 3 Go to the menu bar, click on "File" and then "Open". A dialog window will appear, containing all of the project directories in /home/gbtdata. Select your project from the list. Once you have entered the directory for that project, double click on ScanLog.fits to access your data.

Step 4 Depending upon how much data you have in your project, it will take several seconds to a few minutes to access all of your scans. The load process is complete when you see the list of your scans displayed sequentially on the left hand side of the GFM display.

Step 5 Click on a scan in the scan list window to process it.

4.1.3 Pointing and Focus Data Display

We will describe the details of pointing and focus observations in Chapter 5 (§ 5.2.2.1).

Pointing scans (from Peak, AutoPeak and AutoPeakFocus – see below) will appear under the Pointing Tab. The data display will automatically process the pointing scans. It will calibrate the data, remove a baseline and fit a Gaussian to the data. After the two azimuth scans it will then automatically update the GBT M&C system with the new azimuth pointing offset values that it determined. It will then automatically update the elevation pointing offset after the two elevation scans, unless certain criteria are not met (see § 4.1.3.1). A sample of the Data Display Application after a pointing is shown in Figure 4.1.

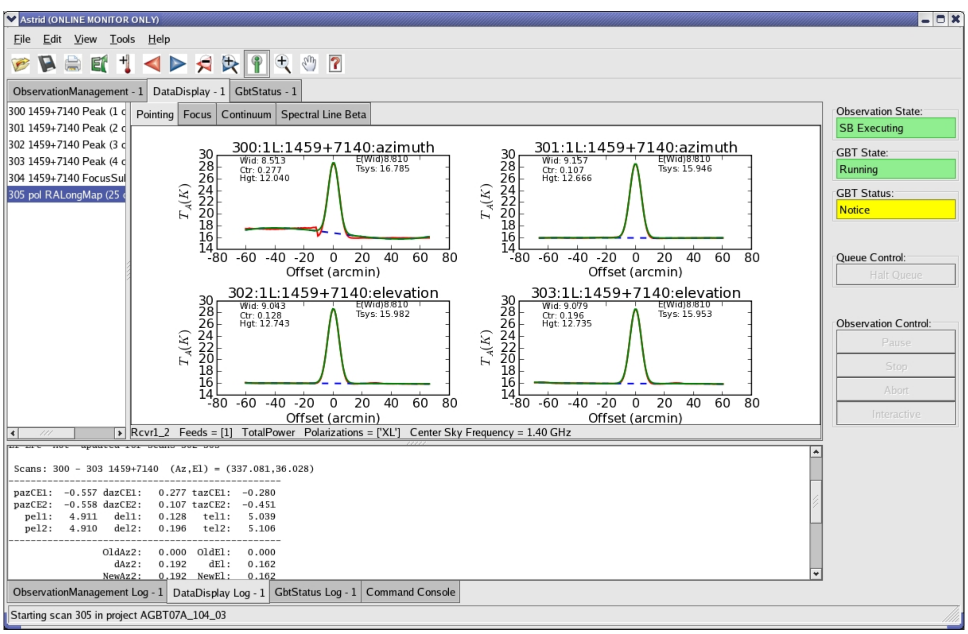


Figure 4.1: The Astrid Data Display Tab showing pointing data.

The focus scan data will appear under the Focus Tab; see Figure 4.2. Again the data will be processed automatically. They will be calibrated, have a baseline removed and a Gaussian will be fit to the data. The focus offset will automatically be sent to the M&C system.

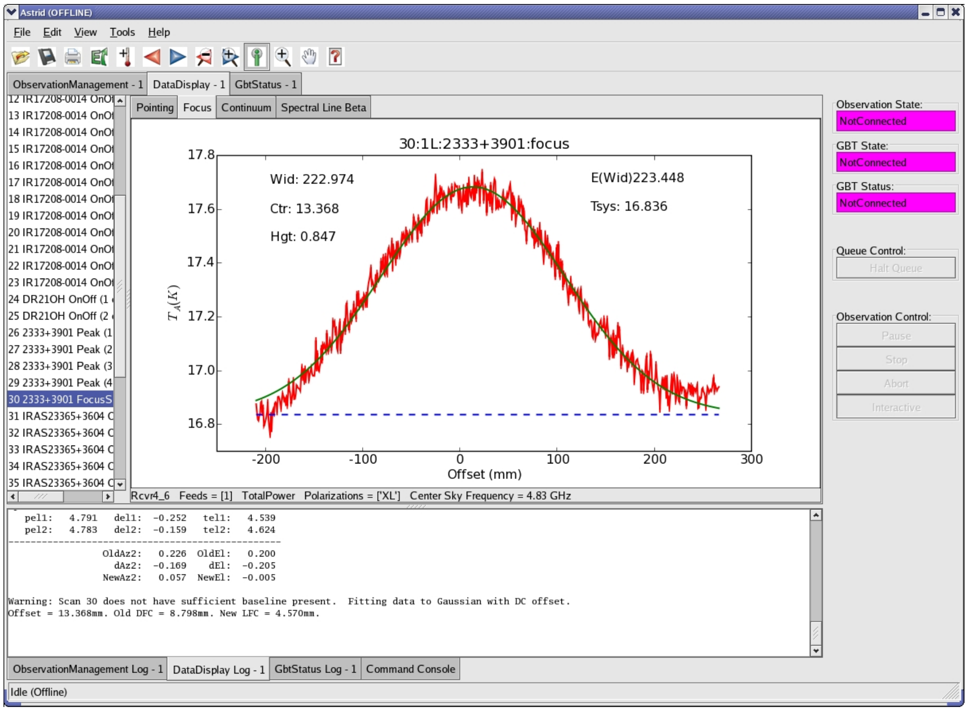


Figure 4.2: The Astrid Data Display Tab showing focus data.

4.1.3.1 Fitting Acceptance Options

GFM has several levels of determining whether or not the pointing and focus solutions will be updated in the M&C system. The expected Full Width at Half the Maximum (FWHM) of the Gaussian fitted to the observed pointing data as the GBT slews across the source should be $\sim 740/\nu_{\rm GHz}$ arc–seconds where $\nu_{\rm GHz}$ is the observing frequency in GHz. For a focus scan the resulting data should approximate a Gaussian with a FWHM of $3.6\lambda_{\rm mm}$ where $\lambda_{\rm mm}$ is the observing wavelength in millimeters. The default behavior is to asssume that a pointing fit is bad if the FWHM differ from the expected value by more than 30% or if the pointing correction is more than twice the FWHM in magnitude. The default for a bad focus scan is if the FWHM is more than 30% from the expected value.

GFM uses the "Automatically accept good fits, automatically reject bad fits" criteria as the default. The user may change fitting acceptance criteria by:

Step 1 Select the Data Display.

Step 2 Select the Pointing Tab or the Focus Tab (see note below).

Step 3 Click on the Tools pull–down menu.

Step 4 Select Options.

Step 5 Select the new mode in the pop–up window (see Figure 4.3).

The options dialog is available only for the Pointing and Focus data displays. Please note that the values are set independently for the pointing data reduction and the focus data reduction. Therefore, the Pointing and Focus can have different option values.

GFM recognizes the following fitting acceptance criteria only when Astrid is in one of its on-line modes:

- Do not apply corrections: Local pointing/focus corrections are never applied, even when the fit is good.
- Automatically accept good fits, automatically reject bad fits: This is the default for Astrid.
- Automatically accept good fits, interactively accept bad fits.
- Interactively accept good and bad fits.
- Accept all automatically: A very dangerous mode that should only be used by experts.

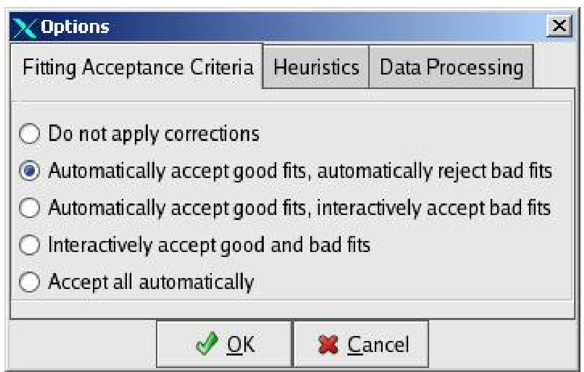


Figure 4.3: The pop–up menu to change the pointing and focus fitting acceptance criteria.

4.1.3.2 Heuristics Options

Heuristics is a generic term used at the GBT to quantify the "goodness of fit" of the pointing and focus data reduction solutions. Based on the known properties of the GBT parts of the solution, such as the beam–width in pointing data, should have certain values within measurement errors. The Heuristics define how large these errors can be. The user may change the Heuristics by:

Step 1 Select the Data Display.

Step 2 Selecd the Pointing Tab or the Focus Tab (see not below).

Step 3 Click on the Tools pull–down menu.

Step 4 Select options.

Step 5 Select the Heuristics tab in the pop–up window.

Step 6 Select the new Heuristics mode in the pop–up window. (see Figure 4.4).

GFM uses "standard" Heuristics as the default upon initialization. The options dialog is available only for the Pointing and Focus data displays. Please note that the values are set independently for the pointing data reduction and the focus data reduction. Therefore, the Pointing and Focus can have different option values.

GFM allows the observer to switch between "standard", "relaxed", and "user-defined" heuristics. The "standard" and "relaxed" heuristic values are predefined and cannot be changed by the user. Under normal observing conditions the observer should expect to use the "standard" values. Under marginal weather conditions the "relaxed" heuristics may be appropriate. The "user-defined" heuristic values should only be used by experts. If you wish to use "user-defined" heuristics then you should contact your GBT support scientist.

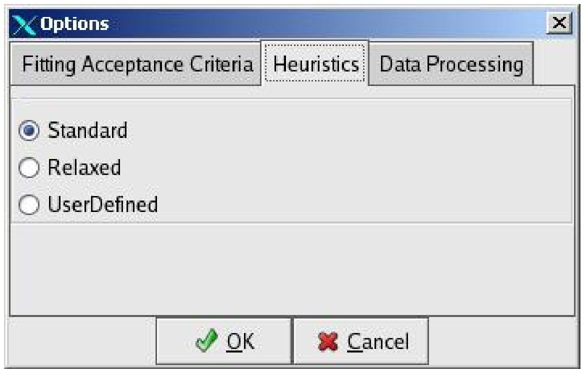


Figure 4.4: The pop–up menu to change the pointing and focus fitting heuristics.

The "standard" heuristics expect that the fitted Gaussians have a FWHM within 30% of the expected values and that the pointing solution is within twice the FWHM of the nominal location of the source. For the "relaxed" heuristics this becomes within 50% of the expected FWHM of the Gaussian fits and three times the FWHM for the pointing correction.

4.1.3.3 Data Processing Options

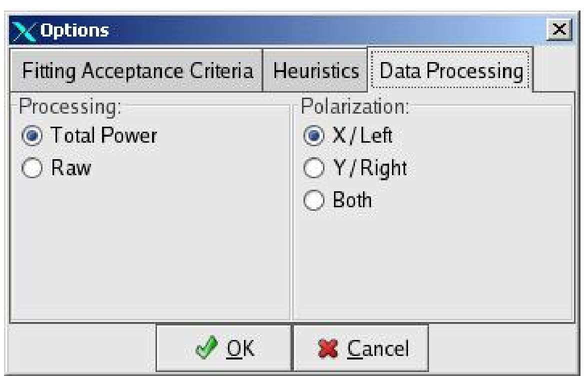


Figure 4.5: The pop–up menu to change the polarization and calibration used in pointing and focus fitting.

The user may change the data processing strategy and/or polarizations used by GFM in reducing pointing or focus scans. This should be done only under extreme circumstances while observing. For example, if the X polarization channel is faulty for some reason, one can use the Y channel instead. This can be done by:

Step 1 Select the Data Display.

Step 2 Selecd the Pointing Tab or the Focus Tab (see not below).

Step 3 Click on the Tools pull–down menu.

Step 4 Select Options.

Step 5 Select the Data Processing tab in the pop–up window.

Step 6 Make the new data processing selections in the pop–up window. (see Figure 4.5).

This options dialog is available only to for Pointing and Focus plugins. Please note that the values are set independently for the pointing data reduction and the focus data reduction. Therefore, the Pointing and Focus can have different option values.

4.1.4 OOF Data Display

"OOF" (Out-Of-Focus holography) is a technique for measuring large-scale errors in the shape of the reflecting surface by mapping a strong point source both in and out of focus. The procedure derives surface corrections which can be sent to the active surface controller to correct surface errors. The procedure is recommended for high-frequency observing at frequencies of 30 GHz and higher.

The AutoOOF procedure will obtain three on-the-fly maps, each taken at a different focus position. Processing will occur automatically upon completion of the third map, and the result will be displayed in the OOF plugin tab of Astrid (see Figure 4.6).

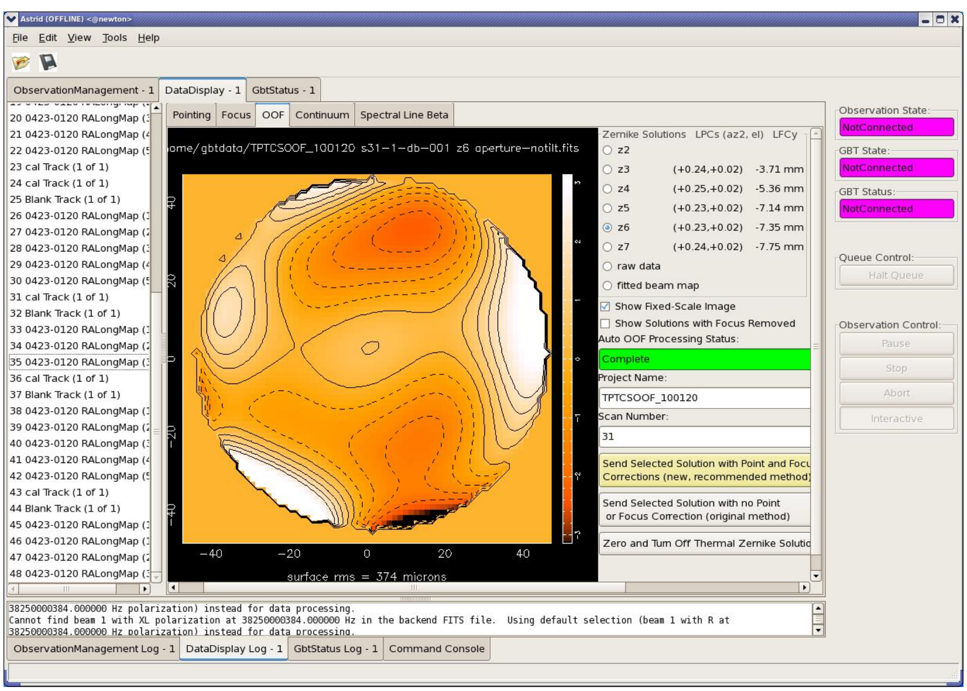


Figure 4.6: The Astrid Data Display Tab showing OOF data.

Solutions must be manually sent to the active surface. The default solution displayed in Astrid is the sixth-order Zernike fit (z6). The most aggressive fit is z7, while z5 is less aggressive. A reasonable solution should contain broad features of less than ±1.5 radians in early to mid-morning to a few radians in the afternoon. If the z6 solution shows large excursions over a significant area at the edge of the dish, then you may wish to drop back to z5. If the calibrator is strong then you may try to use z7, unless the solution begins to show regularly spaced features around the circumference of the dish.

The values of reduced χ^2 for the z5, z6 and z7 fits are shown in the "Fitted beam map" plot. It is important to check the quality of the beam map images before sending the solutions. To do this click the "Show raw data" box. The top row is the raw timestream from the receiver in the three maps. The second row has the baselines removed and the bottom row shows the corresponding beam maps. You should see several detections of the source in all three timestreams and a symmetric right/left positive/negative pattern in all three images. If you do not then the source was not bright enough. Examples of these plots are shown in (Figures 4.7 and 4.8).

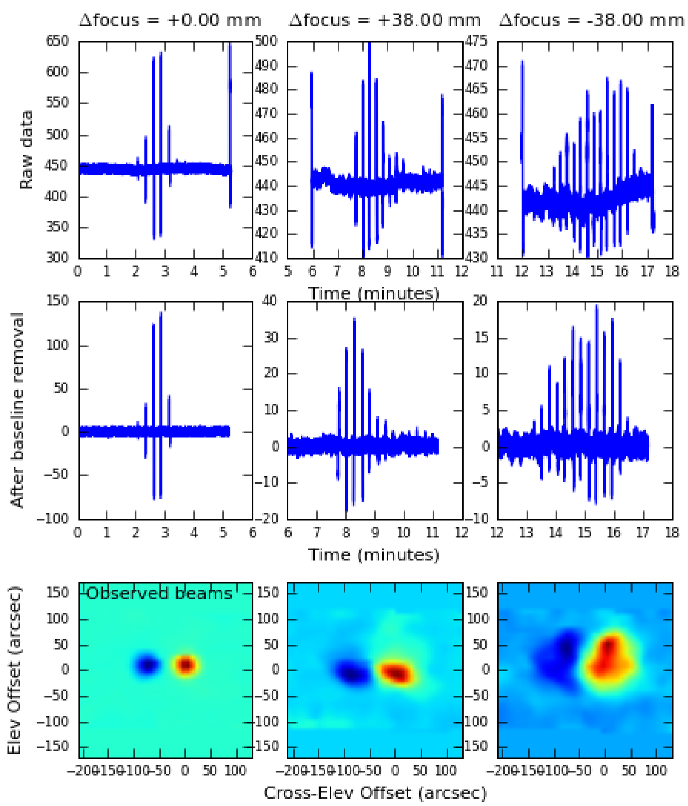


Figure 4.7: A plot of the raw OOF data on a fairly clean Ka/CCB dataset.

When you are ready to accept the solution being displayed it will need to be manually sent to the active surface. A new feature added in October 2009 is the ability to compute the new local pointing and focus offsets from the OOF map. It is recommended that when sending the solutions, you use the yellow button labeled "Send Selected Solution with Point and Focus Corrections (new, recommended method)". If you use this option, you do not have to perform a peak/focus after the OOF map. While it is still good practice to run a peak/focus at the beginning of your observing run (particularly during the day), you can let AutoOOF compute subsequent point/focus corrections.

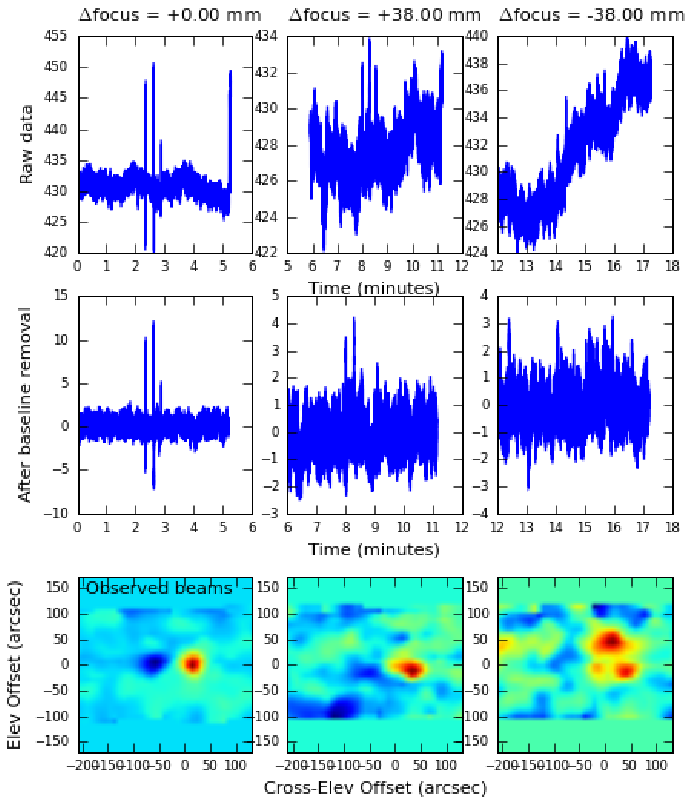


Figure 4.8: A plot of raw OOF data on a source which is too faint.

4.1.5 Continuum Data Display

Continuum data taken with the GBT that are not part of pointing and focus scans will show up in plots under the Continuum Tab (see Figure 4.9). This will show the uncalibrated continuum data as a function of time only.

4.1.6 Spectral Data Display

The Spectral Line Display is a tool for browsing spectral line data. When you are offline, one integration at a time can be selected, while, in online mode, the most recent integration is automatically plotted. See Figure 4.10 for a screen capture of the spectral line data display. The spectra displayed are raw data and no calibration has been applied to them.

All user interaction for this plugin occurs in the right-hand side options panel. The check boxes allow selection of spectra to plot via astronomical variables: Beams, Polarizations, IF Numbers, and Phases. For offline usage, the desired integration can be selected either using the up/down arrows, or by typing in a value in the edit box. For online usage, the latest integration is always shown.

As spectra are plotted, information about each plot is printed in the console window. Each line is color coded to match the color of that spectra in the plotting window. In addition, some of the

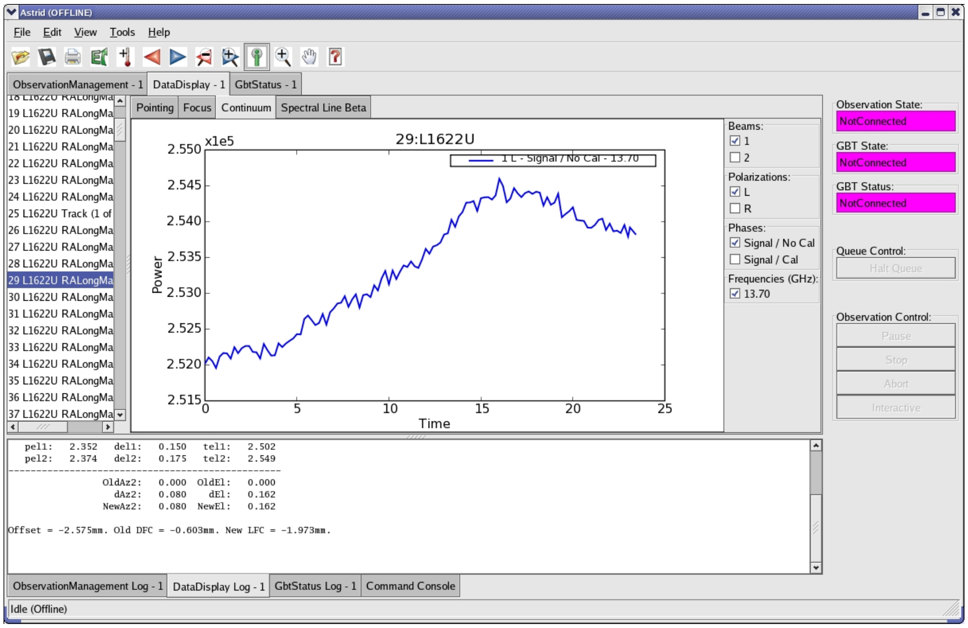


Figure 4.9: The Astrid Data Display Tab showing continuum data.

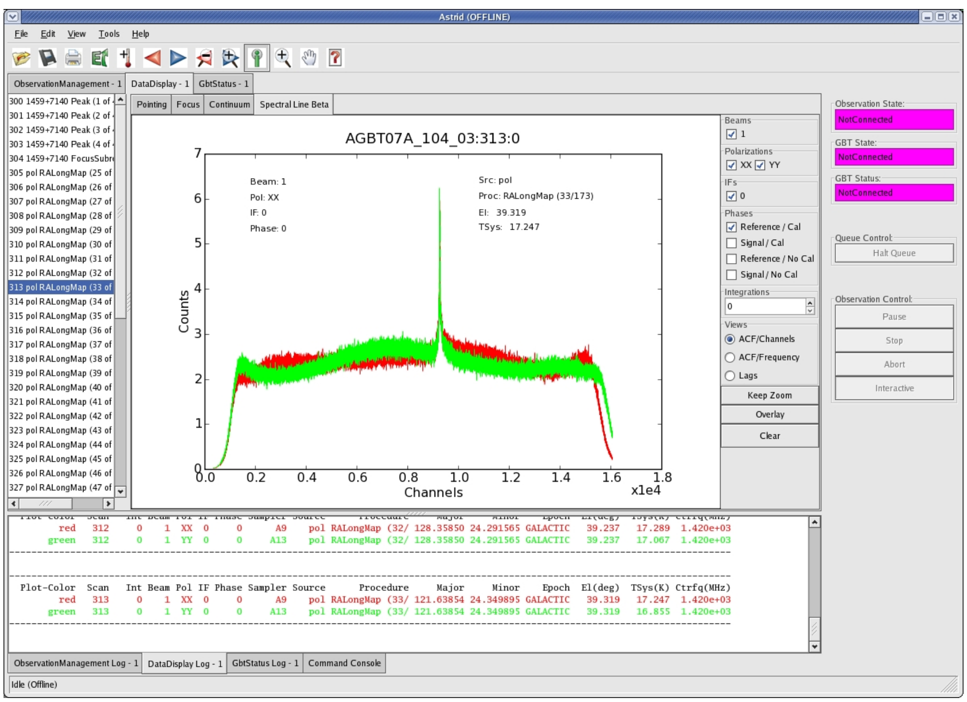


Figure 4.10: The Astrid Data Display Tab showing spectral line data.

information for the very first spectra are used to annotate the plot. The plot title is parsed as project_name:scan_number:integration_number.

The options panel also includes three buttons and a radio box for plot viewing. The "Views" radio box offers options for plotting the bandpass vs. Channels and the bandpass vs. Sky Frequency for all backends. In addition, Spectrometer data will include the option to display the raw auto-correlation lags. The “Keep Zoom” toggle button will maintain the current zoom, even as new spectra are plotted. Using the unzoom command (mouse right-click, or viathe tool bar) will return the plot to its original scale. The “Overlay” toggle button can be used to overplot spectra from different integrations or scans. Finally, the “Clear” button erases the plot.

4.1.6.1 Spectrometer Problems

Common ways that the ACS Spectrometer can malfunction are illustrated in Figure 4.11. The upper spectrum has a strong high-frequency ripple that makes the plot seem to fill the area below the upper envelope. The lower spectrum shows strong spikes at regularly spaced intervals in frequency. If the

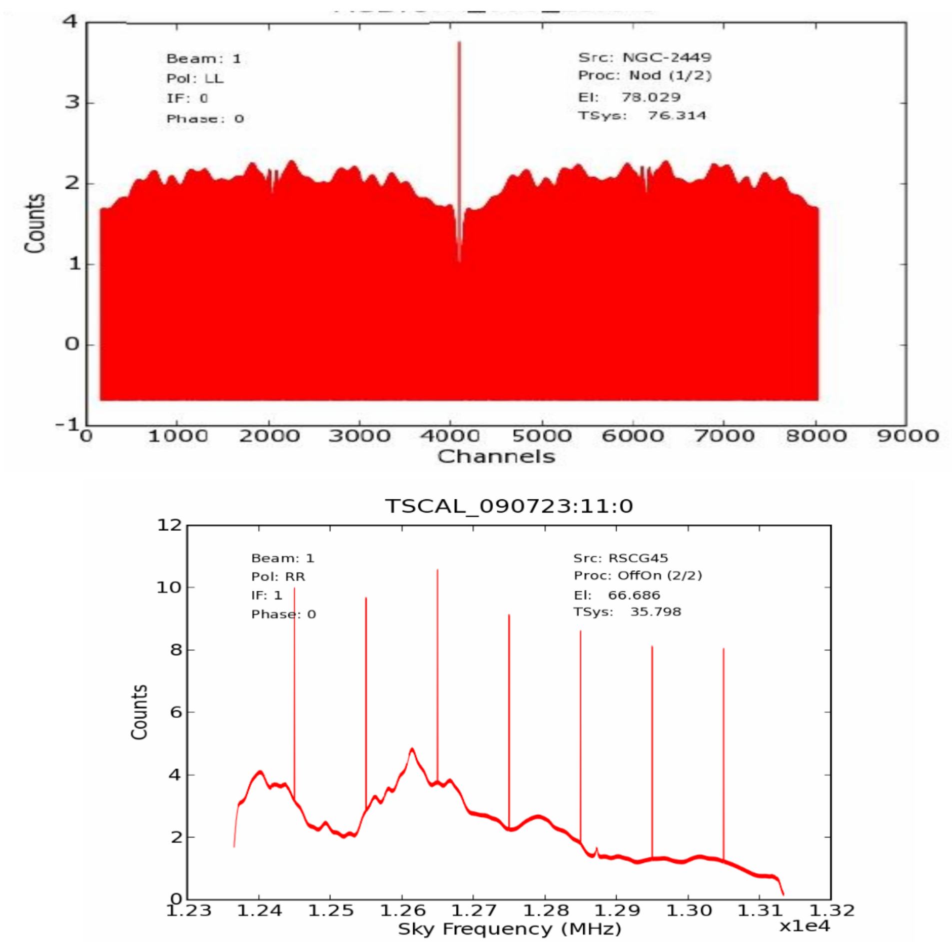


Figure 4.11: Examples of ACS Malfunctions.

ACS appears to be malfunctioning, ask the Telescope Operator to do a ”conform parameters” on the Spectrometer, and then rerun your configuration. If that doesn’t work, ask the Operator to restart the Spectrometer software; if all else fails, call the telescope support scientist.

4.1.7 Creating PNG and Postscript Plots

Note that the "Print" option in the File menu and the printer item in the toolbar do not work. Instead, you select the "export" option or the "E-uparrow" symbol in the toolbar. You can create postscript, encapsulated postscript, or PNG copies of the plots displayed. To do this:

Step 1 Select the Data Display Tab.

Step 2 Display the data for which you would like a plot.

Step 3 Click on the plot button. This button has an "E" followed by and up arrow (see Figure 4.12).

Step 4 Select the directory in which to save the file.

Step 5 In the pop–up window (see Figure 4.13) enter the desired file name for the plot. The name must have an extension of either png, ps or eps.

Step 6 Hit the Save button.

Figure 4.12: The Astrid Data Display plot button.

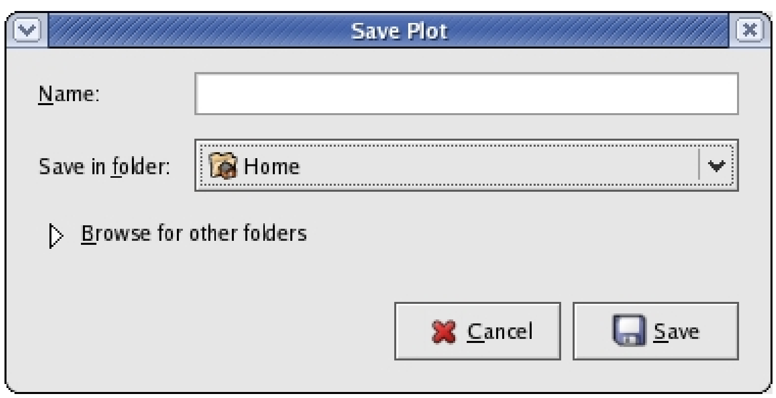


Figure 4.13: The Astrid Data Display pop–up plot window.

4.1.8 Use of Plotting Capabilities

A User Manual is available at http://deap.sourceforge.net/help/index.html that describes all the plotting functionality available in GFM. There is also a plotting Tutorial that illustrates the plotting capabilities by example which is available at http://deap.sourceforge.net/tutorial/index.html.

4.2 The CLEO Utilities

The CLEO (Control Library for Engineers and Operators) system provides a large number of utilities for monitoring and controlling the GBT hardware systems. Some of these are quite useful for observers, although most are intended for expert users and GBT staff. Here described are just a few CLEO utilities that are very useful for observers.

To start CLEO, log in to any Linux workstation in Greenbank, open a terminal window, and type "cleo". A "CleoLauncher" window will appear. Click on the "Launch" menu to get a list of programs that can be run.

Documentation is available on the following web pages, but is somewhat out of date, so its best to consult your GBT "friend" for details. Useful help messages pop up when you hover the mouse over any CLEO widget for a few seconds.
http://www.gb.nrao.edu/ rmaddale/GBT/CLEOManual/index.html
and http://www.gb.nrao.edu/ rmaddale/GBT/CLEOManual/tableofcontents.html.

4.2.1 Talk and Draw

$Launch \rightarrow Utilities/Tools \rightarrow TalkandDraw$

"Talk and Draw" brings up a window for communication with the GBT Telescope Operator. Anything you type in the window will be seen by the Operator, and he can type replies which will show up in your window. Any number of users can open "Talk and Draw" windows at the same time. Everyone running "Talk and Draw" can send messages which will be seen by everyone else. This is a great convenience when doing remote observing. One can also use it for communicating with other members of an observing team who are in remote locations.

4.2.2 Scheduler and Skyview

$Launch \rightarrow Utilities/Tools \rightarrow Scheduler\&Skyview$

This displays a plot of the sky in Az/El coordinates as viewed from Green Bank. One can import a catalog of source positions to be displayed, or display one of the lists of standard calibration sources. By default it displays solar system objects.

For example, to display the x-band pointing sources, start with the "Catalog" button:
$Catalog \rightarrow Add/Select/DeSelectCatalogs \rightarrow xband\_pointing \rightarrow OK$

If one selects "Schedule" (button at upper right), one may enter a date and time and display the sky for that time. It shows the corresponding LST, and moving the cursor on the plot displays the RA/DEC and Az/EL under the cursor. This is very useful for planning observations.

There is also a "Real Time" option in which the location of objects and the direction the GBT is pointed are displayed for the current time.

4.2.3 Status

$Launch \rightarrow Status$

This displays the status of many GBT systems all on one screen. While very useful, it is not recommended for use remotely because it is a heavy user of computing resources. For remote observing, it is recommended to use the Astrid GbtStatus display (See Section 3.5).

4.2.4 Weather

Launch → Weather

This displays the current temperature, humidity, pressure, and wind speed.

4.2.5 CLEO Clock

Launch → Utilities/Tools → Clock

A simple display of both the UTC and LST times, and the Julian Date.

4.2.6 Messages

Launch → Messages

This shows all system status messages. Its often useful to identify problems that might arise with any of the GBT devices.

4.2.7 Other screens

Launch → Receivers → Mustang/PAR

Mustang users will need to bring up the Mustang CLEO screen, as explained in Chapter 16.

Chapter 5

Introduction To Scheduling Blocks And Observing Scripts

Here is an example of what a simple Observing Script looks like:

```
# load the configurations file
execfile("/mypath/myconfigurations.txt")

# load catalogs file
Catalog("/mypath/mycatalog.txt")

# configure the GBT
Configure(myconfig)

# slew to the source
Slew("B0329+54")

# balance the IF system
Balance()

# now observe the source for ten minutes
Track("B0329+54",None,600)
```

The first thing that the script does is to load in the definitions for configuring the GBT's receivers, Intermediate Frequency system (IF) and backends for the observations. This is described in § 5.2.1. Next a catalog that contains information on the sources to observe (containing the source positions and radial velocities, etc.) is loaded. This is described in § 5.2.3. The "Configure" runs the configuration that was defined in "myconfigurations.txt" to select the requested receiver and backend and set switches and frequencies. The telescope is then moved to the desired source with the Slew() procedure (see § 5.2.2) and the power levels in the IF and the backends are balanced so that they should be in their linear regimes (see § 5.2.4). Finally the desired observations are performed using one of the pre-defined scans from § 5.2.2.

5.1 What Are Scheduling Blocks and Observing Scripts?

At the GBT, we use Scheduling Blocks to perform astronomical observations. A Scheduling Block is defined by metadata (the header information provided in the Astrid Run Tab) and an Observing Script. The Observing Script is a list of commands that are "executed" in order to acquire observational data. The Observing Script can contain information for configuring the telescope, balancing the IF, and other commands to "tweak" the telescope system (observing directives) along with the commands (scan types) to collect observational data.

An Observing Script is only part of a Scheduling Block. The Observing Script does not include the observing metadata, such as observer name, etc; and it does not include constraints on when the Scheduling Block should be executed, such as weather required, or the sequence in which Scheduling Blocks must be run.

Astrid interprets Observing Scripts via Python. Thus Observing Scripts should follow Python syntax rules (such as indentation for loops) and can also contain or make use of any Python commands.

5.1.1 Making An Observing Script

Observing Scripts must be created well prior to your telescope time. We suggest that you review Observing Scripts with your project's contact support scientist.

Observing scripts can be written using Astrid's "Observation Management" Edit tab, which contains a simple text editor reminiscent of Notepad (MS Windows), or you can choose to write your script outside of Astrid and use the "Observation Management" Import facility in Astrid to upload it into the database; see § 3.3.1.2 for details.

For the database, you should choose a descriptive name for your Observing Script, such as map_G11.0 or pointfocus, which will remind you of the science you are trying to accomplish by running that particular script. Names such as "test" or "turtle.p" are not descriptive and should be avoided. The name you choose can be up to 96 characters long, and can contain white spaces, so you may have an Observing Script name that consists of a few words (such as "K-band frequency-switched spectroscopy"). You do not need to add a suffix to your Observing Script name (*.sb or *.py).

5.2 Components of an Observing Script

A typical Observing Script will include: a) a configuration for the system; b) specification of sources via a catalog; c) a slew to a source and then balancing the IF and maybe other Observing Directives; and d) the observational scan type commands. It is highly recommended that the source catalog and the configuration definitions reside in files external to the Observing Script. This will allow quick changes/refinements without re-validation and saving of the Observing Scripts.

In the following sections we discuss each of these components.

5.2.1 Configuration of the GBT IF System

5.2.1.1 Overview

The routing of signals through the GBT system is controlled by many electronic switches which eliminate the need to physically change cables by hand. The GBT's electronically configurable IF allows many, and more complicated, paths for the signals to co-exist at all times. Experience has shown us that manual configuration of the GBT IF is not practical due to setup times which typically lasted 30 minutes to 1 hour. We have thus developed a software tool which will configure the GBT IF based on the values of about one dozen input parameters. This suite of software can typically configure the GBT IF in under one minute.

The basic syntax for a configuration definition is:

```
myconfiguration = """
# This is a comment and is ignored by the software
primarykeyword1 = your_primarykeyword_value
primarykeyword2 = your_primarykeyword_value
...
...
...
primarykeywordN = your_primarykeyword_value
"""
```

5.2.1.2 Example Configurations

The best way to learn about how to define and perform configurations is through examples. We will discuss how to use the example configuration definition shown below in an Observing Script in § 5.2.1.3. All of the keywords available for use in a configuration definition will be discussed in § 5.2.1.4.

Continuum Observations

```
# this file contains the configuration definitions

# configuration definition for continuum observations
continuumconfig="""
receiver  = 'Rcvr1_2'
beam      = 'B1'
obstype   = 'Continuum'
backend   = 'DCR'
nwin      = 1
restfreq  = 1400
bandwidth = 80
swmode    = 'tp'
swtype    = 'none'
swper     = 0.2
tint      = 0.2
vframe    = 'topo'
vdef      = 'Radio'
noisecal  =  'lo'
pol       = 'Linear'
"""
```

The above configuration definition has been given the name "continuumconfig" and can be used for pointing and focusing observations or for continuum mapping. For these observations we have selected continuum observations [obstype='Continuum'] using the single beam L–band (1 to 2 GHz) receiver [receiver='Rcvr1_2'; beam='B1'] and the DCR as the backend detector [backend='DCR']. We wish to take data with a single band [nwin=1] which has an 80 MHz bandwidth [bandwidth=80] centered on 1400 MHz [restfreq=1400]. We wish to do total power observations [swmode='tp'; swtype='none'] with the time to go through a full switching cycle being 0.2 seconds [swper=0.2]. We want the DCR to record data every 0.2 seconds [tint=0.2]. We do not wish to Doppler track the rest frequency since these are continuum observations [vframe='topo'; vdef='Radio']. We would like to use the low-power noise-diode [noisecal='lo']. Finally, we wish to take the data using linear polarization [pol='linear'].

If you are unsure as to what is meant by these keywords then you should see § 5.2.1.4.

Spectral Line, Frequency Switching Observations

```
# configuration definition for spectral line observations
# using frequency switching
spectral_fs_config="""
receiver = 'Rcvr1_2'
beam = 'B1'
obstype = 'Spectroscopy'
backend = 'SpectralProcessor'
nwin = 1
restfreq = 1420
bandwidth = 10
swmode = 'sp'
swtype = 'fsw'
swper = 1.0
swfreq = 0, -2.5
tint = 2.0
vframe = 'lsrk'
vdef = 'Radio'
nchan = 'high'
sp.mode = 'Square'
noisecal = 'lo'
pol = 'Linear'
"""
```

The second configuration definition will be named "spectral_fs_config" and can be used for spectral line observations [obstype='Spectroscopy'] using frequency switching [swmode='sp'; swtype='fsw']. For these observations we wish to use the single beam L–band (1 to 2 GHz) receiver [receiver='Rcvr1_2'; beam='B1'] and the Spectral Processor as the backend detector without cross-polarization products [backend='SpectralProcessor'; sp.mode='Square']. We wish to take data with a single band [nwin=1] which has a 10 MHz bandwidth [bandwidth=10] centered on 1420 MHz [restfreq=1420] with the highest number of spectral channels available [nchan='high']. We wish the cycle time to go through a full switching cycle to be 1 second [swper=1.0] while we wish the frequency switching states to be centered on the line and then shifted by -2.5 MHz [swfreq=0,-2.5]. We want the Spectral Processor to record data every 2 seconds [tint=2.0]. We wish to Doppler track the spectral line in the commonly used Local Standard of Rest velocity frame [vframe='lsrk'; vdef='Radio']. We would like to use the noise-diode with the lower equivalent system temperature [noisecal='lo']. Finally, we wish to take the data using linear polarization [pol='linear'].

Multiple Spectral Lines, Position Switching Observations

```
# configuration definition for multiple spectral line observations
```

```
# using position switching
spectral_ps_config="""
receiver   = 'Rcvr8_10'
obstype    = 'Spectroscopy'
backend    = 'Spectrometer'
nwin       = 4
restfreq   = 9816.867, 9487.824, 9173.323, 8872.571
deltafreq = 0, 0, 0, 0
bandwidth = 12.5
swmode     = 'tp'
swtype     = 'none'
swper      = 1.0
tint       = 30
vlow       = 0
vhigh      = 0
vframe     = 'lsrk'
vdef       = 'Radio'
noisecal   =  'lo'
pol        = 'Circular'
nchan   =   'medium'
spect.levels = 9
"""
```

The third configuration definition to be considered has the name "spectral_ps_config" and can be used for spectral line observations [obstype='Spectroscopy'] using position switching [swmode='tp'; swtype='none']. For these observations we wish to use the single beam X–band (8 to 10 GHz) receiver [receiver='Rcvr8_10'; beam='B1'] and the Spectrometer as the backend detector without cross-polarization products [backend='Spectrometer']. We wish to take data on multiple spectral lines [nwin=4], each having a 12.5 MHz bandwidth [bandwidth=12.5] with the middle value for the number of spectral channels available [nchan='medium'] and with 9-level sampling [spect.levels=9]. Each spectral window will be centered on the rest frequencies of the lines at 9816.867, 9487.824, 9173.323, and 8872.571 MHz [restfreq=9816.867, 9487.824, 9173.323, 8872.571]. We wish the cycle time to go through a full switching cycle to be 1 second [swper=1.0]. We want the Spectrometer to record data every 30 seconds [tint=30]. We wish to Doppler track the spectral line in the commonly used Local Standard of Rest velocity frame [vframe='lsrk'; vdef='Radio']. We would like to use the low-power noise-diode [noisecal='lo']. Finally, we wish to take the data using circular polarization [pol='Circular'].

Multiple Spectral Lines, Multi-beam Nodding Observations

```
# configuration definition for spectral line observations
# using a multi-beam receiver for nodding observations
spectral_nod_config="""
receiver   = 'Rcvr22_26'
beam       = 'B34'
obstype    = 'Spectroscopy'
backend    = 'Spectrometer'
nwin       = 4
restfreq   = 23694.495, 23722.633, 23870.129, 25056.025
deltafreq = 0,0,0,0
bandwidth = 50
swmode     = 'tp'
swtype     = 'none'
```

```
swper      = 1.0
tint       = 30
vlow       = 0
vhigh      = 0
vframe     = 'lsrk'
vdef       = 'Radio'
noisecal   =  'lo'
pol        = 'Circular'
nchan   =   'low'
spect.levels = 3
"""
```

The final example configuration has the name "spectral_nod_config" and can be used for spectral line observations [obstype='Spectroscopy'] using nodding observations [swmode='tp'; swtype='none'] with the upper half of the multi-beam K–band (22 to 26 GHz, beams 3 and 4) receiver [receiver='Rcvr22_26'; beam='B34'] and the Spectrometer as the backend detector without cross-polarization products [backend='Spectrometer']. We wish to take data on multiple spectral lines [nwin=4], each having a 50 MHz bandwidth [bandwidth=50] with the lowest value for the number of spectral channels [nchan='low'] and with 3-level sampling [spect.levels=3]. Each spectral window will be centered on the rest frequencies of the lines at 23694.495, 23722.633, 23870.129, and 25056.025 MHz [restfreq=23694.495, 23722.633, 23870.129, 25056.025]. We wish the cycle time to go through a full total power switching cycle to be 1 second [swper=1.0]. We want the Spectrometer to record data every 30 seconds [tint=30]. We wish to Doppler track the spectral line in the commonly used Local Standard of Rest velocity frame [vframe='lsrk'; vdef='Radio']. We would like to use the low-poer noise-diode [noisecal='lo']. Finally, we wish to take the data using circular polarization [pol='Circular'].

5.2.1.3 Executing A Configuration

Although one can put any number of configuration definitions into an observing script, the standard practice is to put all the configuration definitions into a separate file that is then "included" in the Observing Script. This strategy: a) keeps the Observing Scripts simple and without clutter; and b) allows changes to be made to the configuration without having to validate and re-save the Observing Scripts.

With the configuration definitions defined in a separate file from the Observing Script, you need to use the python execfile() function to bring the configuration definitions into the Observing Script. Note that you do not want to use the python import command to do this because it will not reread the file and will miss any changes that you may have made.

We have placed all four example configurations in § 5.2.1.2 within a single configuration file with the name,

`/home/astro-util/projects/6D01/configurations.py`

An Observing Script for continuum observations would look like

```
# observing script to make a continuum map of 3C 286

# load the source catalog
c = Catalog("/home/astro-util/astridcats/fluxcal.cat")

# load the configuration definitions
execfile("/home/astro-util/projects/6D01/configurations.py")
```

```
# peform the configuration
Configure("continuumconfig")

# slew the telescope to 3C286
Slew("3C286")

# Make an on-the-fly  map with 6 rows each 120' long,
# using a spacing of 6' and scan rate of 720'/min:

RALongMap("3C286",
          Offset("J2000", 2.0, 0.0, cosv=True),
          Offset("J2000", 0.0, 0.5, cosv=True),
          Offset("J2000", 0.0, 0.1, cosv=True),
          10.0)
```

The configurations are: a) all contained in one external file; and b) brought into the Observing Script using the execfile() command. The desired configuration is then executed using the name of the configuration definition as an argument to the Configure() command.

5.2.1.4 Configuration Keywords

So far we have shown you how to use the configuration. Now we need to discuss what keywords and values are allowed in a configuration definition.

Keywords That Must Always Be Present

The following keywords do not have default values and must be present in all configuration definitions.

receiver This keyword specifies the name of the GBT receiver to be used. The names and frequency ranges of the receivers can be found in Table 5.1. The value of the receiver keyword is a string and should therefore be placed within quotes when used.

obstype This keyword specifies the type of observing to be performed. The allowed values are one of the following strings: “Continuum”, “Spectroscopy”, “Pulsar”, “Radar”, “VLBI”.

backend This keyword specifies the name of the backend (data acquisition system) to be used. The value for this keyword is a string. Valid backends are listed in Table 5.2.

restfreq This keyword specifies the rest frequencies for spectral line observations or the center frequencies for continuum observations. Up to eight different frequencies can be given in a comma separated list. Values are floats and are given in MHz. Note that only the first value given will be properly Doppler tracked. All other frequencies may need to have Doppler corrections applied during data reduction.

bandwidth This keyword gives the bandwidth in MHz to be used by the specified backend. The value of the keyword should be a float. Possible values depend on the receiver and backend that are chosen (see Table 5.3).

Table 5.1: GBT receivers and their nominal frequency ranges.

Name	Frequency Range (GHz)	Notes
Rcvr_342	.290–.395	
Rcvr_450	.385–.520	
Rcvr_600	.510–.690	
Rcvr_800	.680–.920	
Rcvr_1070	.910–1.23	
Rcvr1_2	1.15–1.73	
Rcvr2_3	1.73–2.60	
Rcvr4_6	3.95–5.85	
Rcvr8_10	8.00–10.0	
Rcvr12_18	12.0–15.4	
Rcvr18_22	18.0–22.4	
Rcvr22_26	22.0–26.5	
RcvrArray18_26	18.0–26.5	7-beam focal plane array
Rcvr_26_40	26.0–31.0, 30.5–37.0, 36.0–40.0	
Rcvr40_52	40.5–47.0	
Rcvr_PAR	80.0—100.00	Mustang Bolometer Array
NoiseSource	N/A	

Table 5.2: GBT backends.

Name	Notes
DCR	The Digital Continuum Receiver directly from the IF Rack. Four/two frequencies maximum for single/dual beam receivers.
DCR_AF	The Digital Continuum Receiver from the Analog Filter Rack. Eight/four frequencies maximum for single/dual beam receivers.
SpectralProcessor	Spectral line backend with up to 1024 channels and 4 frequencies with narrow bandwidths.
Spectrometer	Spectral line backend with up to 32768 channels and 8 frequencies with large bandwidths.
VLBA_DAR	Very Long Baseline Array Data Acquisition Recorder.
Radar	For bi-static radar observations. Private backend.
GUPPI	Green Bank "Ultimate" Pulsar Processor.
CCB	CalTech Continuum Backend
Zpectrometer	Wide-band Spectrometer

Table 5.3: Bandwidths for different receiver/backend combinations.

Backend	Receiver	Possible Bandwidths (MHz)
Spectrometer	Any	12.5, 50, 200, 800
Spectral Processor	Any	40, 20, 10, 5, 2.5, 1.25, 0.625, 0.3125, 0.15625, 0.078125
DCR, VLBI, or Radar	Prime Focus	20, 40, 80, 240
DCR, VLBI, or Radar	Rcvr1_2, Rcvr4_6, Rcvr8_10, Rcvr12_18	20, 80, 320, 1280
DCR, VLBI, or Radar	Rcvr2_3, Rcvr18_26, Rcvr40_52	80, 320, 1280
DCR_AF	Any	12.5, 50, 200, 800

Keywords With Default Values

swmode This keyword specifies the switching mode to be used for the observations. This keyword's values are given as a string. Values are "tp" (total power with cal), "tp_nocal" (total power without cal) , "sp" (switched power with cal), "sp_nocal" (switched power without cal). The default value is "tp".
The switching schemes are:

Total Power With Cal The noise diode is periodically turned on and off for equal amounts of time.

Total Power Without Cal The noise diode is turned off for the entire scan.

Switched Power With Cal The noise diode is periodically turned on and off for equal amounts of time while another component is in a signal state and then again in a reference state. This is used in frequency switching where the signal state is one frequency and the reference state is another frequency. Similarly beam switching and polarization switching change the beams or polarizations so that their signals are sent down two different IF paths.

Switched Power Without Cal The noise diode is turned off while another component is switched between a signal and reference state.

swtype This keyword is only used when swmode="sp" or swmode="sp_nocal", and specifies the type of switching to be performed. This keyword's values are "none", "fsw" (frequency switching), "bsw" (beam switching) and "psw" (polarization switching). **Default values are "fsw" for all single beam receivers except receiver="Rcvr_26_40". The default for receiver="Rcvr_26_40" is swtype="bsw".**

swper This keyword defines the period in seconds over which the full switching cycle occurs. The value is a float. Default values are 0.2 for obstype="continuum", 0.04 for obstype="pulsar", and 1.0 for any other value for the obstype keyword.

swfreq This keyword defines the frequency offsets used in frequency switching (swtype="fsw"). The value consists of two comma separated floats which are the pair of frequencies in MHz. Default values are swfreq=-0.25*Bandwidth, +0.25*Bandwidth for swtype="fsw", and swfreq=0,0 otherwise.

tint This keyword specifies the backend's integration (dump) time. The value is a float with units of seconds. Default values are 10.0 for obstype="continuum", tint=swper for obstype="spectroscopy" and 30.0 of any other value for the obstype keyword.

beam This keyword specifies which beams are to be used for observations with multi-beam receivers. The keyword value is a string. Possible values are "B1", "B2", "B3", "B4", "B12" (both beams 1 and 2), "B34" (both beams 3 and 4). This has a different meaning from the "beamName" in the observing scans. This **beam** indicates which signals from the receiver are to be recorded – e.g., "B12" means to record data from both feeds 1 and 2. The default value is "B1".

nwin This keyword specifies the number of frequency windows that will be observed. The number of values given for the restfreq and deltafreq keywords must be the same as nwin. The value for this keyword is an integer. Possible values are 1,2,3,4,6, and 8. The default value is 1. nwin is backend and receiver dependent, see § 2.1.4.

deltafreq This keyword specifies offsets in MHz for each spectral window so that the restfreq is not centered in the middle of the spectral window. The values for this keyword consist of a comma separated list of floats. Default values are 0.0. See Appendix B for more details on the use of deltafreq.

vlow and vhigh These keywords specify the minimum and maximum velocity to be observed from a group of sources. The value is a float and is in km s^{-1} for velocities. The default value is 0.0. See Appendix C for more details on the use of vlow and vhigh. The use of vlow and vhigh is not recommended for frequencies where there can be large amounts of Radio Frequency Interference (RFI).

vframe This keyword specifies the velocity frame (the inertial reference frame). The keyword value is a string. Allowed values are "topo" (topocentric, i.e. Earth's surface), "bary" (Barycenter of solar system), "lsrk" (Local Standard of Rest kinematical definition , i.e. normal LSR definition), "lsrd" (Local Standard of Rest dynamical definition – rarely used), "galac" (center of galaxy), "cmb" (relative to Cosmic Microwave Background). The default value is "topo".

vdef This keyword specifies which mathematical equation (i.e. definition) is used to convert between frequency and velocity. The keyword value is a string. Allowed values are "Optical", "Radio", "Relativistic".

$$\mathrm{v_{optical}} = \mathrm{c}\left[1 - \frac{\nu}{\nu_\mathrm{o}}\right]$$

$$\mathrm{v_{radio}} = \mathrm{c}\left[\frac{\nu_\mathrm{o}}{\nu} - 1\right]$$

$$\mathrm{v_{relativistic}} = \mathrm{c}\left[\frac{\nu_\mathrm{o}^2 - \nu^2}{\nu_\mathrm{o}^2 + \nu^2}\right]$$

The default value is "Radio".

iftarget This keyword specifies the target voltage level to use when balancing the IF Rack. The keyword value is a float. The nominal range of the IF Rack is 0.0–10.0 and the linear range is 0.1–5.0. Default values are listed in Table 5.4.

Table 5.4: The default IF target levels. The receiver categories A, B, and C are actually based on the nominal IF center frequencies for these receivers (3, 6 and 1.08 GHz, respectively). The Receiver Groups are as follows: A) L–band, C–band, X–band, Ku–band; B) S–band, K–band, Ka–band, Q–band; and C) any prime focus receiver (P–band).

Receiver Group	IF Rack Bandwidth (MHz)	ACS 50 & 12.5 MHz 9 Level (Volts)	ACS 50 & 12.5 MHz 3 Level (Volts)	ACS 800 MHz & 200 MHz (Volts)	All Other backends (Volts)
A	20	0.1	0.5	1.0	1.0
A	80	0.1	0.5	1.0	1.0
A	320	1.0	1.0	1.0	1.0
A	1280	1.0	1.0	1.0	1.0
B	80	1.0	1.0	1.0	1.0
B	320	1.0	1.0	1.0	1.0
B	1280	1.0	1.0	1.0	1.0
A or B	AllPass	3.0	5.0	1.0	1.0
C	≤ 80	0.1	0.5	1.0	1.0
C	> 80	1.0	1.0	1.0	1.0

Backend and Receiver Dependent Keywords

Some configuration keywords depend on which backends and receivers are being used. Some observations may require one of these keywords while for other observations none may be needed.

sp.mode This keyword specifies the mode of use for the Spectral Processor. The keyword value is a string. Possible values are "Square" (RR and LL or XX and YY terms only), "Cross" (RL and LR or XY and YX terms only), and "SqrCross" (RR, LL, RL and LR or XX, YY, XY and YX terms for full Stokes observations). This keyword does not have a default value.

nchan This keyword is used to determine the number of spectral channels that the spectrometer or Spectral Processor will use when more than one choice is available. This keyword is a string. Available values are "high", "medium" and "low". They respectively choose the highest, medium or lowest resolution available for a given mode for the spectrometer or Spectral Processor. Please see the "GBT Proposers Guide" for possible modes and number of channel options. This keyword does not have a default value.

spect.levels This keyword specifies the number of sampler levels in the Analog to Digital (A/D) signal conversion that is desired in the GBT Spectrometer. This keyword value is an integer that is either 3 or 9. For 800 and 200 MHz bandwidth modes only 3 level sampling is available. For 50 and 12.5 MHz bandwidth modes both 3 and 9 level sampling are available. This keyword does not have a default value.

pol Each of the prime focus receivers, L–band, S–band and C–band receivers have a hybrid that can output either linear or circular polarization. This keyword specifies whether linear or circular polarization is desired for these receivers. The keyword value is a string. Allowed values are "Linear" and "Circular". The default value is "Circular" for the Very Long Baseline Interferometer (VLBI)and Radar back ends, and "Linear" otherwise.

noisecal All receivers below 12 GHz have two noise diodes for calibration signals – one with an equivalent brightness temperature at roughly one tenth the system temperature (lo value) and one nearly equal to the system temperature (hi value). This keyword is a string which specifies which noise diode is to be used. Allowed values[1] are "lo", "hi" and "off". The default value is "lo" except for the Radar backend for which the default values is "off".
For the Ka–band receiver there are three additional choices. These are 'L', 'R', or 'LR'. The Ka–band receiver has two "lo" noise diodes, one for each polarization for each of the two beams. The 'L', 'R', and 'LR' options specify which of these noise diodes are to be used with the Ka–band receiver.

notchfilter There is a notch filter covering roughly 1200–1310 MHz in the L–band receiver that filters out an FAA radar signal. This keyword determines if this notch filter is in place and used by the system or is removed from the receiver's RF path. The keyword value is a string with allowed values of "In" or "Out". The default value is "In".

spect.crosspol This keyword determines whether the spectrometer will create cross polarization products (i.e. RR, LL, RL and LR or XX, YY, XY and YX correlations). The keyword values is a string. To turn on cross polarization products the value should be "y". To turn off the cross polarization products the value should be "n". The default value is "n".

guppi.obsmode GUPPI-specific keyword (see Chapter 14). Obsmode can be 'search', 'fold', or 'cal'.

guppi.numchan GUPPI-specific keyword. Numchan can be a power of two from 64 to 4096.

guppi.polnmode GUPPI-specific keyword. Polnmode is 'full_stokes' or 'total_intensity'.

guppi.scale GUPPI-specific keyword. (see Chapter 14 for details.)

guppi.outbits GUPPI-specific keyword. Currently only outbits=8 is available.

[1]There are expert values of "on-mcb", "on-ext", "lo-mcb", "hi-mcb", "lo-ext" and "hi-ext" whose use is beyond the scope of this document. Please contact a support person about the use of these values.

guppi.fold_dumptime GUPPI-specific keyword. Fold-specific parameters are not needed for cal or search. For fold or cal observations, this is how much we will integrate the pulsar (or cal) before dumping a set of profiles to disk. It must be shorter than the scan duration that you set via the Track() command.

guppi.fold_bins GUPPI-specific keyword. Number of bins in profile.

guppi.fold_parfile GUPPI-specific keyword. Pulsar ephemeris parameter file – make sure that it exists!

guppi.datadisk GUPPI-specific keyword. This is the top-level directory (i.e. RAID array, 'data1' or 'data2') where your data will be stored. It will go in a subdirectory called **/guppi.datadisk/observername/projectID/date/**. Since the data will be owned by the "monctrl" computer account, you will not be able to remove it – that means Scott Ransom will bug you mercilessly until you process your data!

Expert Keywords

These keywords should only be used by very experienced observers who have expert knowledge of how a given backend works or in how the GBT IF works.

sp.polyco This expert keyword sets the polyco file name for the Spectral Processor when it is used in the pulsar mode. The keyword value is a string that contains the full path plus filename of the polyco file.

spect.numbanks This is an optional, expert keyword that can be used to set the number of banks that the spectrometer uses. In most cases there is only one choice and the config tool chooses the default. There are a few cases in which one may choose an alternate number of banks. The keyword value is a string and can be "None", "1", "2", or "4".

vlbi.phasecal This expert keyword turns the Very Long Baseline Interferometer (VLBI)phase cals on or off. The phase cals can can run at 1 MHz ("M1") or 5 MHz ("M5"). The keyword value is a string. Allowed values are "off", "M1" or "M5".

xfer This expert keyword sets the beam switch for the Ku–band, K–band and Ka–band receivers. The keyword is a string. Allowed values are "ext", "thru", or "cross". The default values are "ext" when swtype="bsw" and "thru" otherwise.

polswitch This expert keyword sets the polarization switch for the L–band and X–band receivers. The keyword value is a string. Allowed values are "ext", "thru", and "cross". The default value is "ext" if swtype="psw" and "thru" otherwise.

ifbw This expert keyword sets the minimum IF bandwidth to be used in filters within the receiver and in the IF Rack. The keyword value is float with units of MHz.

if0freq This expert keyword is used to set the center frequency of the IF after the mixing the RF signal with the first Local Oscillator (LO). The keyword value is a float with units of MHz.

lo1bfreq This expert keyword is used to set the frequency of the synthesizer used for the alternative First LO (LO1), LO1B. This keyword is only to be used with the Ka–band receiver. The keyword value is a float with units of MHz.

lo2freq This expert keyword is used to set the frequency values of the eight Second LO (LO2) synthesizers within the Converter Rack. The keyword values are a comma separated list of floats with units of MHz. There should be nwin entries for this keyword value.

if3freq This expert keyword is used to set the IF input frequency of the backend. The keyword value is a comma separated list of floats with units of MHz. There should be nwin entries for this keyword value.

5.2.2 Scan Types

A Scan is a pattern of antenna motions that when used together yield a useful scientific dataset. This section describes the various scan types that are available for use within GBT Observing Scripts. Each scan type consists of one or more scans, which are the individual components of the antenna's motion on the sky. The scan types listed below are the functions within your Observing Script where data will be obtained with the GBT.

In Table 5.5 we show the available, "built-in" Scan Types for the GBT along with a short description of what each Scan Type does. More details on each Scan Type can be found in the subsequent text after the table.

Please note that the syntax for all Scan Types is case-sensitive. Location, Offset, Horizon, and Time objects are defined in § 5.2.5 while Catalogs are defined in § 5.2.3. Seldom used scan types are discussed in Appendix F.2.

Location
: Most Scan commands require a "location" parameter. This may be either a Location object (see § 5.2.5 and Appendix D), or it may be the name of a radio source given in a Catalog (see § 5.2.3).

beamName
: Most Scan commands use a "beamName" parameter. This should not be confused with the **beam** keyword in Configurations (see Section 5.2.1.4). This indicates the "tracking beam" i.e., the beam that is pointed at the specified location. It may have values "1", "2", "3", up to the maximum beam number for the specified receiver. The beam numbers and their relative locations depend on the receiver. A value of "C" means the center of the receiver box, where there may or may not be a feed. Syntax such as "MR12" or "MR34" means halfway between beams 1 and 2 or 3 and 4, respectively. These are used for subreflector nodding, (see "SubBeamNod" in Section 5.2.2.3). For multi-beam and array receivers, such as the KFPA 7-beam receiver, one may use constructions such as "MR7,11" for halfway between beams 7 and 11, and **beamname="arraycenter"** to track the center of the array.

5.2.2.1 Utility Scans

AutoPeakFocus

The intent of this scan type is to automatically peak and focus the antenna for the current location on the sky and with the current receiver. Therefore it should not require any user input. However, by setting any of the optional arguments the user may partially or fully override the search and/or procedural steps as described below.

AutoPeakFocus() should not be used with Prime Focus receivers. The prime focus receivers have pre-determined focus positions and there is not enough travel in the feed to move them significantly out of focus.

Syntax: AutoPeakFocus(source, location, frequency, flux, radius, balance, configure, beamName, gold)

The parameters to AutoPeakFocus() are:

source A string. It specifies the name of a particular source in the pointing catalog or in a user-defined Catalog. The default is None. Specifying a calibrator bypasses the search process. Please note that NVSS source names are used in the pointing catalog. If the name is not located in the pointing catalog then all the user-specified catalogs previously defined in the scheduling block are searched. If the name is not in the pointing catalog or in the user defined catalog(s) then the procedure fails.

Table 5.5: The most commonly used Scan Types available for the GBT.

Observing Type	Scan Type	Description
Continuum	AutoPeakFocus	Selects and observes a nearby calibration source and updates the pointing and focus corrections.
Continuum	AutoPeak	Selects and observes a nearby calibration source and updates the pointing corrections.
Continuum	AutoFocus	Selects and observes a nearby calibration source and updates the focus correction.
Continuum	AutoOOF	Selects and observes a nearby calibration source with different focus settings to create an out-of-focus holography map to update the surface.
Continuum	Peak	Performs a pointing observation.
Continuum	Focus	Performs a focus observation.
Continuum	Tip	Performs an observation to derive T_{sys} vs. elevation.
Continuum, Line, Pulsar, VLB	Slew	Slews the telescope to the specified source or Location.
Continuum, Line, Pulsar, VLB	Track	Follows a single position or moves with a constant velocity while taking data.
Continuum, Line	OnOff	Observe a source and then a reference position.
Continuum, Line	OffOn	Observe a reference position and then the source.
Continuum, Line	OnOffSameHA	Observe a source and then a reference position using the same Hour Angle as the source observations.
Continuum, Line	Nod	Observe a source with one beam and then with the other beam for a dual-beam receiver.
Continuum, Line	DecLatMapWithReference	Make an on-the-fly raster map by moving along the minor axis of the coordinate system and making periodic reference observations.
Continuum, Line	DecLatMap	Make an on-the-fly raster map by moving along the minor axis of the coordinate system.
Continuum, Line, Pulsar	PointMapWithReference	Make a map using individual pointings with periodic reference observations.
Continuum, Line, Pulsar	PointMap	Make a map using individual pointings.
Continuum, Line	RALongMapWithReference	Make an on-the-fly raster map by moving along the major axis of the coordinate system and making periodic reference observations.
Continuum, Line	RALongMap	Make an on-the-fly raster map by moving along the major axis of the coordinate system.
Continuum, Line	BalanceOnOff	Move to the source and then a reference position and then balance the IF system to the mid-point of the two power levels.
Continuum, Line	SubBeamNod	Moves the subreflector alternately between two beams of the reciever.

location A Catalog source name or Location object (see Appendix D). It specifies the center of the search radius. The default is the antenna's current beam location on the sky.

frequency A float. It specifies the observing frequency in MHz. The default is the rest frequency used by the standard continuum configurations, or the current configuration value if **configure** is False (see Table 5.6).

flux A float. It specifies the minimum acceptable calibration flux in Jy at the observing frequency. The default is 20 times the continuum point-source sensitivity.

radius A float. The routine selects the closest calibrator within the radius (in degrees) having the minimum acceptable flux. The default radius is 10 degrees. If no calibrator is found within the radius, the search is continued out to 180 degrees and if a qualified calibrator is found the user is given the option of using it [default], aborting the scan, or continuing the scheduling block without running this procedure.

balance A boolean. Controls whether after slewing to the calibrator the routine balances the power along the IF path and again to set the power levels just before collecting data. Allowed values are True or False. The default is True.

configure A boolean. This argument causes the scan type to configure the telescope for continuum observing for the specified receiver. The default is True. **Note: because AutoPeakFocus() is self-configuring, one must re-configure the GBT IF path for your normal observing after the pointing and focus observations are done, unless the configure parameter is set to False.** Also be aware that setting configure to False means the observer must insure the DCR is properly configured and included in the Scan Coordinator, as the AutoPeakFocus() procedures will not check the configuration of the GBT.

beamName A string. It specifies which receiver beam will be the center of the cross-scan. beamName can be "C", "1", "2", "3", "4", etc, up to "7" for the KFPA receiver. The default value is the recommended value for the receiver. If you configure for one beam, and point with another (using the beamName parameter) you can have very, very bad data. Make sure that if you choose "Configure=False" and "beamName" that the two are compatible!!! (Note that this parameter has different syntax than the beam configuration keyword.)

gold A boolean. If True then only "Gold standard sources" (.i.e. sources suitable for pointing at high frequencies) will be used by AutoPeakFocus(). This parameter is ignored if the "source" parameter is specified.

The sequence of events done by **AutoPeakFocus** in full automatic mode, i.e, with no arguments are:

Step 1 Get recommended beam, antenna/subreflector motions, and duration for peak and focus scans.

Step 2 Get current receiver from the M&C system.

Step 3 Get current antenna beam location from the control system.

Step 4 Configure for continuum observations with the current receiver.

Step 5 Run a balance (see § 5.2.4.2) to obtain accurate system temperature readings from the DCR.

Step 6 Select a source using computed minimum flux, observing frequency, location, and search radius.

Step 7 If no pointing source is found within the specified radius, then provide the observer the option to use a more distant source (default), and if none found either aborting (second default) or continuing the scheduling block.

Step 8 Slew to source.

Step 9 Run a balance to set scan power levels.

Step 10 Run a scan using Peak

Step 11 Run a scan using Focus.

Setting optional arguments will cause the scan to skip some steps.

These examples demonstrate the expected use of AutoPeakFocus:

```
AutoPeakFocus()          # use all default values
AutoPeakFocus('3C286')   # point and focus on 3C286
AutoPeakFocus(location=Location("J2000", "16:30:00", "47:23:00"))
        # find a pointing source near ra=16:30:00 dec=47:23:00
```

Normally AutoPeak and AutoPeakFocus will choose reasonable scanning rates and lengths. Table 5.6 gives the standard parameters. If you need to use non-standard values for a peak or focus, use the "Peak" or "Focus" commands explicitly.

Table 5.6: Recommended lengths and times for performing peak and focus observations.

Band	ν (MHz)	$\Delta\nu$ (MHz)	Beam FWHM	Beam	Focus FWHM	Peak Length ’	Peak Time sec	Focus Length mm	Focus Time sec	Notes
PF1	340	20	36’	1	3.2m	180	30	—	—	A,C
PF1	415	20	30’	1	2.6m	180	30	—	—	A,C
PF1	680	20	18’	1	1.6m	90	15	—	—	A,C
PF1	770	20	16’	1	1.4m	80	15	—	—	A,C
PF2	970	20	13’	1	1.1m	65	15	—	—	A,C
L–band	1400	80	8.8’	1	76cm	130	30	480	60	B,D,E
S–band	2000	80	6.2’	1	54cm	90	30	480	60	B,D,E
C–band	5000	80	2.5’	1	22cm	40	30	480	60	B,D,E
X–band	9000	80	1.4’	1	12cm	20	30	480	60	B,D,E
Ku–band	14000	320	53”	1	76mm	18	30	320	60	B,D,F
K–band (l)	21500	320	33”	1	50mm	12	30	270	60	B,D,F
K–band (h)	25000	320	30”	3	43mm	10	30	240	60	B,D,F
Ka–band	32000	320	23”	1	32mm	8	30	180	60	B,D,F
Q–band	43000	320	17”	1	25mm	6	30	120	60	B,D,F

A Prime Focus: Peak Lengths are chosen to be 5 x FWHM with a scan time of 15 seconds to have good sampling across the beam.

B Gregorian Focus: Peak Rates are chosen to give 2 seconds across the FWHM, Peak Times to give a scan time of 30 seconds (to allow vibrations to settle).

C Prime Focus: Axial focus measurements are not recommended for prime focus receivers since the gain changes only slightly over the entire focus range.

D Gregorian Focus: The optimal focus length is 2 x FWHM, but to allow for varying baselines we currently recommend $\sim$ 3 x focus FWHM, plus 40mm at each end to allow for the fact that focus measurement is done with respect to to focus tracking curve, not last offset. The Focus Rate is then chosen to give a 60sec scan time. This is a trade-off between completing the focus scan quickly, and allowing any potential scan-start anomalies to die away.

E At L through X–band, the focus rates and lengths are conservative limits set by subreflector hardware (the absolute maximum would be 600mm/min and 600mm).

F At Ku–band and higher frequencies the peak length is rather larger to accommodate the beam separation in azimuth for these multi-beam receivers.

AutoPeak

AutoPeak() is the same as AutoPeakFocus() except that it does not perform a Focus scan (Step 11 above).

AutoFocus

AutoFocus() has the same parameters as AutoPeakFocus(). However it does not do a pointing and only does a Focus scan [skipping Step 10 in AutoPeakFocus()].

AutoFocus() should not be used with Prime Focus receivers. The prime focus receivers have pre-determined focus positions and there is not enough travel in the feed to move these receivers significantly out of focus.

Peak

The Peak scan type sweeps through the specified sky location in the four cardinal directions. Its primary use is to determine pointing corrections for use in subsequent scans. Note that the hLength, vLength and scanDuration should be overridden as a unit since together they determine the rate. Peak assumes that the user has executed a continuum configuration.

Syntax: Peak(location, hLength, vLength, scanDuration, beamName)

The only required parameter for Peak is location.

The parameters for peak are

location A Catalog source name or Location object. It specifies the source upon which to do the scan.

hLength An Offset object. It specifies the horizontal distance used for the Peak. The default value is the recommended value for the receiver (see Table 5.6).

vLength An Offset object. It specifies the vertical distance used for the Peak. The default value is the recommended value for the receiver (see Table 5.6).

scanDuration A float. It specifies the length of each scan in seconds. The default value is the recommended value for the receiver (see Table 5.6).

beamName A string. It specifies the receiver beam to use for both scans. beamName can be “C”, “1”, “2”, “3”, “4” or any valid combination for the receiver you are using such as “MR12” and “MR34”. The default value is the recommended value for the receiver. If you configure for one beam, and point with another (using the beamName parameter) you can have very, very bad data. Make sure that if you configure with the same beam with which you Peak!!!

The following example does a peak in encoder coordinates with 90 minute lengths and a 30 second scan duration using beam 1.

```
Peak("0137+3309", Offset("Encoder", "00:90:00", 0),
          Offset("Encoder, 0, "00:90:00"), 30, "1")
```

or with the defaults.

```
Peak("0137+3309")
```

Slew

Slew moves the telescope beam to point to a specified location on the sky.

Syntax: Slew(location, offset, beamName)

The parameters for Slew are

location A Catalog source name or Location object. It specifies the source to which the telescope should slew. The default is the current location in “J2000” coordinate mode.

offset An Offset object. It moves the beam to an optional offset position that is specified relative to the location specified in the location parameter value. The default is None. See § 5.2.5 for inforamtion on Offset objects.

beamName A string. It specifies the receiver beam to use for the scan. beamName can be “C”, “1”, “2”, “3”, “4” or any valid combination for the receiver you are using such as “MR12” and “MR34”. The default is “1”.

Slew does the following based on the arguments provided:

1) If only a location is given the antenna slews to the indicated position.

2) If a location and offset are given, the antenna slews to the indicated position plus the specified offset.

3) If only an offset is given, the antenna slews to the current location plus the specified offset.

The following example slews to 3C 48 using the center of all the receiver's beams:

```
Slew("3C48", beamName="C")
```

Focus

The Focus scan type moves the subreflector or prime focus receiver (depending on the receiver in use) through the axis aligned with the beam. Its primary use is to determine focus positions for use in subsequent scans.

Syntax: Focus(location, start, focusLength, scanDuration, beamName)

The only required parameter for Focus() is location.

The parameters for Focus() are

location A Catalog source name or Location object. It specifies the source upon which to do the scan.

start A float. It specifies the starting position of the subreflector (in mm) for the Focus scan. See Table 5.6 for the recommended value for each receiver.

focusLength A float. It specifies the ending position of the subreflector relative to the starting location (also in mm). See Table 5.6 for the recommended value for each receiver.

scanDuration A float. It specifies the length of each scan in seconds. See Table 5.6 for the recommended value for each receiver.

beamName A string. It specifies the receiver beam to use for the scan. beamName can be “C”, “1”, “2”, “3”, “4” or any valid combination for the receiver you are using such as “MR12” and “MR34”. The default for each receiver is listed in Table 5.6. If you configure for one beam, and focus with another (using the beamName parameter) you can have very, very bad data. Make sure that if you configure with the same beam with which you Focus!!!

In the following example a focus of the subreflector is performed from -200 to +200mm at 400mm/min using beam 1:

```
Focus("0137+3309", -200.0, 400.0, 60.0, "1")
```

Or using the defaults:

```
Focus("0137+3309")
```

Tip

The Tip scan moves the beam on the sky from one elevation to another elevation while taking data and maintaining a constant azimuth.

Syntax: Tip(location, endOffset, scanDuration, beamName, startTime, stopTime)

The parameters for Tip are

location A Catalog source name or Location object. It specifies the start location of the tip scan. The Location must be in AzEl or encoder coordinates.

endOffset An Offset object. It specifies the beam's final position for the scan, relative to the location specified in the first parameter. The Offset also must be in AzEl or encoder coordinates.

scanDuration A float. It specifies the length of each scan in seconds.

beamName A string. It specifies the receiver beam to use for the scan. beamName can be "C" (center), "1", "2", "3", "4" or any valid combination for the receiver you are using such as "MR12" (i.e., track halfway between beams 1 and 2) and "MR34". The default value for beamName is "1".

startTime A time string with the following format: "hh:mm:ss". It allows the observer to specify a start time for the Tip.

stopTime A time string with the following format: "hh:mm:ss". It allows the observer to specify a stop time for the Tip.

Scan timing may be specified by either a scanDuration, a stopTime, a startTime plus stopTime, or a startTime plus scanDuration.

The following example tips the GBT from 6.0 degrees in elevation to 80.0 degrees in elevation over a period of three minutes using beam "1":

```
Tip(Location("AzEl", 1.5, 6.0),
    Offset("AzEl", 0.0, 74.0),
    300.0)
```

BalanceOnOff

When there is a large difference in power received by the GBT between two positions on the sky, it is advantageous to balance the IF system power levels to be at the mid-point of the two power levels. Typically this is needed when the "source position" is a strong continuum source. This scan type has been created to handle this senario; one should consider using it when the system temperature on and off source differ by a factor of two or more.

BalanceOnOff() slews to the source position and then balances the IF system. It then determines the power levels that are observed in the IF Rack. Then the telescope is slewed to the off position and the power levels are determined again. The change in the power levels is then used to determine attenuator settings that put the balance near the mid-point of the observed power range. Note that the balance is determined only to within ± 0.5 dB owing to the integer settings of the IF Rack attenuators.

Syntax: BalanceOnOff(location, offset, beamName)

The parameters for BalanceOnOff() are

location A Catalog source name or Location object. It specifies the source upon which to do the scan.

offset An Offset object. It moves the beam to an optional offset position that is specified relative to the location specified in the location parameter value. The default is None. See § 5.2.5 for inforamtion on Offset objects.

beamName A string. It specifies the receiver beam to use for the scan. beamName can be "C", "1", "2", "3", "4" or any valid combination for the receiver you are using such as "MR12" and "MR34". The default for each receiver is listed in Table 5.6.

The following example balances on 3C48 and remeasures 1 degrees off:

```
Catalog("fluxcal")
BalanceOnOff("3C48", Offset("J2000", 1.0, 0.0))
```

5.2.2.2 AutoOOF

"OOF" (Out-Of-Focus holography) is a technique for measuring large-scale errors in the shape of the reflecting surface by mapping a strong point source both in and out of focus. The procedure derives surface corrections which can be sent to the active surface controller to correct surface errors. The procedure is recommended for high-frequency observing at frequencies of 26 GHz and higher.

AutoOOF() has the same parameters as AutoPeakFocus(). However, the receiver parameter is limited to 'Rcvr26_40', 'Rcvr40_52', and 'Rcvr_PAR' (i.e., MUSTANG). When using the Rcvr_PAR an additional parameter **nseq** can specify the number of OTF maps to be made with AutoOOF. This must be either 3 or 5.

The intent of the AutoOOF scan is to automatically run an Out of Focus holography scan for the current location on the sky and with the current receiver, therefore it should not require any user input. However, by setting any of the optional arguments the user may partially or fully override the source search and/or procedural steps as described below.

AutoOOF() should only be used for observations above 26 GHz.

Details and recommended strategy

It is important to choose a bright calibrator: preferably at least 7 K in the observed band, which is about 4 Jy at Q-band. You should not rely on the catalog flux to be accurate as it is often many years out of date. If you are not sure then run a point/focus scan on the calibrator first in order to confirm its strength. Remember, you need to be able to detect the source when the subreflector is $\pm 5\lambda$ out of focus. which typically reduces its peak intensity by a factor of 8. The AutoOOF procedure will obtain three on-the-fly maps, each taken at a different focus position, and each requiring 5 minutes of observing time plus nearly a minute of initial M&C overhead per map, plus 1 minute for processing, for a total of 19 minutes. The processing is launched automatically upon completion of the third map, and the result is displayed in the OOF plugin tab of Astrid. Because this is a new observing mode, it is incumbent upon the user to examine the solutions, and click the button (in the Astrid DataDisplay tab) to send the selected solution to the active surface.

It is recommended that when sending the solutions, you use the yellow button in the OOF display tab labeled "Send Selected Solution with Point and Focus Corrections (new, recommended method)". By using this method, it is no longer necessary to follow AutoOOF with another AutoPeakFocus.

If you plan to run AutoOOF as the first thing during your observing slot, we recommend running an AutoPeakFocus before the AutoOOF. Subsequent runs of AutoOOF will not need a pre-point/focus as small errors in these values do not harm the results.

How long does the solution remain valid?

- Nighttime
  If the corrections are measured at least an hour after sunset, then they should last for the next few hours as the backup structure cools off. This can take many hours if it was a sunny day. At frequencies below 90 GHz, the corrections can be turned off sometime between midnight and 3AM. If a sidelobe begins to appear on a bright pointing source during this timeframe, then the previous AutoOOF is no longer valid. Turning off the corrections may improve the surface, or a new AutoOOF may be necessary.

- Daytime
  During the daytime, this is a difficult question to answer, as it depends on how much the pose of the telescope is changing with respect to the Sun, cloud cover changes, etc. The answer can be anything from 1-4 hours. In practice, we suggest running an AutoPeak every 30-40 minutes and watching for the reappearance of a sidelobe on the elevation scans. When the sidelobe becomes significant, it is probably a good time for another AutoOOF.

AutoOOF Scheduling blocks

The scheduling block to execute AutoOOF is quite simple. The command normally does not require any arguments, although specifying the source is prudent. We recommend choosing a source with a flux density of at least 3-4 Jy. You may specify a flux density cutoff using the flux argument.

- Q-band with DCR
  If Q-band is the current receiver (Rcvr40_52), then AutoOOF will automatically configure the DCR to use the widest continuum filter (1280 MHz). So you may simply issue:

```
AutoOOF(source=''2253+1608'')
Break(''Examine solutions and send them to active surface'')
```

  The default polarization is “R”. If you want to select the “L” channel, then specify an additional argument: channel=0. If you don’t know the name of a bright nearby calibrator, you may alternatively specify a flux density cutoff: AutoOOF(flux=3.0), but beware that the flux density database is not kept current, and this option has not been tested much.

- Ka-band with CCB (preferred)
  If the Ka-band is the current receiver (Rcvr26_40), then AutoOOF will automatically configure the CCB and will use the second highest frequency channel (34.25 GHz), because it provides significantly better receiver temperature than the highest frequency band. Use the same commands as for Q-band (above).

- Ka-band with DCR (backup)
  In case the preferred backend (CCB) is not functional, the DCR can be used instead. In order to do this, you must configure the DCR prior to calling AutoOOF. As an example, we provide a DCR configuration used by GB.PTCS.

```
execfile(''/home/groups/ptcs/obs/turtle/configs.py'')
Configure(kaband)
AutoOOF(source=''2253+1608'', configure=False)
```

- AutoOOF with MUSTANG.
  Here is an example of setting up and running AutoOOF with Rcvr_PAR (a.k.a MUSTANG). One must set up and tune up the MUSTANG system first; refer to Chapter 16.

```
Configure("/users/bmason/mustangPub/sb/mustangfull.conf")
mySrc=''1159+2914'
Catalog()
Slew(mySrc)
AutoOOF(source=mySrc)
```

More information on AutoOOF can be found at
https://safe.nrao.edu/wiki/bin/view/GB/PTCS/AutoOOFInstructions

5.2.2.3 Observing Scans

Track

The Track scan type follows a sky location while taking data.

Syntax: Track(location, endOffset, scanDuration, beamName, startTime, stopTime)

The parameters for Track are

location A Catalog source name or Location object. It specifies the source which is to be tracked.

endOffset An Offset object. It moves the beam to a new position during the scan which is specified relative to the location specified in the first parameter. If no offset is desired, use None for this parameter.

scanDuration A float. This specifies the length of the scan in seconds.

beamName A string. It specifies the receiver beam to use for the scan. beamName can be “C”, “1”, “2”, “3”, “4” or any valid combination for the receiver you are using such as “MR12” and “MR34”. The default value for beamName is “1”.

startTime A time object. This specifies when the scan begins. If the start time is in the past then the scan starts as soon as possible with a message sent to the scan log. If the start time plus the scan duration is in the past, then the scan is skipped with a message to the observation log. The value may be:

- A time object. Note, if startTime is more than ten minutes in the future then a message is sent to the observation log. See § 5.2.5 for information on time objects.
- A Horizon object. When a Horizon object is used, the start time is implicitly computed, e.g.,

```
Track("3C247", None, 120.0, startTime=Horizon())
```

  If the source never rises then the scan is skipped and if the source never sets then the scan is started immediately. In either case a message is sent to the observation log. See § 5.2.5 for information on Horizon objects.

stopTime A time object. (See § 5.2.5 for information on time objects.) This specifies when the scan completes. If the stop time is in the past then the scan is skipped with a message to the observation log. The value may also be:

- A Horizon Object. When a Horizon object is used, the stop time is implicitly computed, e.g., a complete scheduling block for tracking VirgoA from rise until set and using a horizon of 20 degrees would be

```
horizon = Horizon(20.0)
Track("VirgoA", None, startTime=horizon, stopTime=horizon)
```

  If the source never sets, then the scan stop time is set to 12 hours from the current time. See § 5.2.5 for information on Horizon objects.

Scan timing must be specified by either a scanDuration, a stopTime, a startTime plus stopTime, or a startTime plus scanDuration.

The following example tracks 3C48 for 60 seconds using the center beam

```
Track("3C48", None, 60.0)
```

OnOff

The OnOff scan type performs two scans. The first scan is on source, and the second scan is at an offset from the source location used in the first scan.

Syntax: OnOff(location, referenceOffset, scanDuration, beamName)

The parameters of OnOff are:

location A Catalog source name or Location object. It specifies the source upon which to do the On scan.

referenceOffset An Offset object. It specifies the location of the Off scan relative to the location specified by the first parameter.

scanDuration A float. It specifies the length of each scan in seconds.

beamName A string. It specifies the receiver beam to use for both scans. beamName can be "C", "1", "2", "3", "4" or any valid combination for the receiver you are using such as "MR12" and "MR34". The default value for beamName is "1".

The following example does an OnOff scan with reference offsets of 1 degree of arc in Right Ascension and 1 degree of arc in Declination and a 60 second scan duration, using beam 1:

```
OnOff("0137+3309", Offset("J2000", 1.0, 1.0, cosv=False), 60, "1")
```

OffOn

The OffOn scan type is the same as the OnOff scan except that the first scan is offset from the source location.

Syntax: OffOn(location, referenceOffset, scanDuration, beamName)

The following example does an OffOn scan with reference offsets of 1 degree of arc in Right Ascension and 1 degree of arc in Declination and a 60 second scan duration, using beam 1:

```
OffOn("1831=0949", Offset("J2000", 1.0, 1.0, cosv=False), 60, "1")
```

OnOffSameHA

The OnOffSameHA scan type performs two scans. The first scan is on the source, and the second scan follows the same HA track used in the first scan.

Syntax: OnOffSameHA(location, scanDuration, beamName)

The parameters for OnOff are

location A Catalog source name or Location object. It specifies the source upon which to do the On scan.

beamName A string. It specifies the receiver beam to use for both scans. beamName can be "C", "1", "2", "3", "4" or any valid combination for the receiver you are using such as "MR12" and "MR34". The default value for beamName is "1".

scanDuration A float. It specifies the length of each scan in seconds.

The following example does an OnOffSameHA scan with a 60 second scan duration, using beam 1:

```
OnOffSameHA("0137+3309", 60, "1")
```

Nod

The Nod procedure does two scans on the same sky location with different beams. **Nod should only be used with dual beam receivers.**

Syntax: Nod(location, beamName1, beamName2, scanDuration)

The parameters for Nod are

location A Catalog source name or Location object. It specifies the source upon which to do the Nod.

beamName1 A string. It specifies the receiver beam to use for the first scan. beamName1 can be “C”, “1”, “2” or any valid combination for the receiver you are using such as “MR12” and “MR34”.

beamName2 A string. It specifies the receiver beam to use for the second scan. beamName2 can be “C”, “1”, “2” or any valid combination for the receiver you are using such as “MR12” and “MR34”.

scanDuration A float. It specifies the length of each scan in seconds.

The following example does a Nod between beams 1 and 2 with a 60 second scan duration:

```
Nod("1011-2610", "1", "2", 60.0)
```

SubBeamNod

For two-beam receivers SubBeamNod causes the subreflector to tilt about its axis between the two feeds at the given periodicity. The primary mirror is centered on the midpoint between the two beams. The beam selections are extracted from the scan’s beamName, i.e., MR12 or MR34. The ”first” beam (”1” or ”3”) performs the first integration. The periodicity is specified in seconds (float) per nod (half-cycle). A nod is limited to a minimum of 4.4 seconds.

Syntax: SubBeamNod(location, scanDuration, beamName, nodLength, nodUnit)

The parameters for SubBeamNod() are:

location A Catalog source name or Location object. It specifies the source upon which to do the nod.

scanDuration A float. It specifies the length of each subscan in seconds.

beamName A string. It specifies the receiver beam pair to use for nodding. beamName can be ”MR12” or ”MR34”.

nodLength A number (integer for ”integrations”, and float or integer for ”seconds”). Type depends on value of nodUnit. It specifies the half-cycle time. This is the time spent in one position plus move time to the second position.

nodUnit A string, either ”integrations” or ”seconds”. The default is ”seconds”.

An example:

```
SubBeamNod("3C48",scanDuration=60.0, beamName=''MR12'',
           nodLength=4.4826624)
```

Alternatively, one can specify the nod time in units of the primary backend's integration times (integer) by setting the periodicity units to integrations instead of the default seconds, e.g.,

```
SubBeamNod("3C48",scanDuration=60.0, beamName=''MR12'', nodLength=3,
            nodUnit="integrations")
```

If the backend's actual integration time is obtainable then a warning is issued if the alignment between the integration times and the nod times shift over the duration of the scan by more than 10% of the nod time. A warning is issued in any case if the backend's actual integration time is not obtainable. Attempting to use integrations as the unit when the integration time cannot be obtained from the selected backend will cause a failure.

In either case, when the subreflector is moving the entire integration during which this occurs is flagged. The scan will end at the end of the scanDuration (once the current integration is complete) regardless of the phase of the nod cycle.

The following example does a subreflector nod between beams 1 and 2 for 60 seconds, each nod or half-cycle lasts for three integrations where Rcvr26_40 was selected in the configuration with an integration time of 1.5 seconds:

```
SubBeamNod("1011-2610'', 60.0, ''MR12'', nodLength=3,
            nodUnit=''integrations'')
```

In this example one out of every three integrations will be blanked because the subreflector is moving. So the sequence will be: Beam 1 (2 intg.), Blanked while subreflector moving (1 intg.), Beam 2 (2 intg.), Blanked (1 intg.), Beam 1 (2 intg.), etc.. It takes about 0.5 seconds for the subreflector to move between beams but the entire integration time will be blanked. If nodLength=5, then only one in five integrations would be blanked.

Control of the subreflector may be done with any scan type using the submotion class. This should only be done by expert observers. Those observers interested in using this class should contact their GBT support person.

5.2.2.4 Mapping Scans

DecLatMapWithReference

A Declination/Latitude map with reference observations, or DecLatMapWithReference, peforms a raster scan centered on a specific location on the sky while periodically moving to a reference location. Scanning is done in the declination, latitude, or elevation coordinate depending on the desired coordinate mode. This procedure allows the user to periodically move to a reference position on the sky (please see DecLatMap if no reference is required). The starting point of the map is defined as (-hLength/2, -vLength/2).

Syntax: DecLatMapWithReference(location, hLength, vLength, hDelta, referenceOffset, referenceInterval, scanDuration, beamName, unidirectional, start, stop)

The parameters to DecLatMapWithReference() are

location A Catalog source name or Location object. It specifies the center of the map.

hLength An Offset object. It specifies the horizontal width of the map (i.e., the extent in the longitude-like coordinate).

vLength An Offset object. It specifies the vertical height of the map (i.e., the extent in the latitude-like coordinate).

hDelta An Offset object. It specifies the distance between "columns" in the map. Note that hDelta values must be positive.

referenceOffset An Offset object. It specifies the position of the reference source on the sky relative to the Location specified by the first input parameter (i.e. the center of the map).

referenceInterval An integer. It specifies when to do a reference scan in terms of map columns, e.g. 4 means every fourth column. A reference scan is always done before doing column number 1.

scanDuration A float. It specifies the length of each scan in seconds.

beamName A string. It specifies the receiver beam to use for the scan. beamName can be "C", "1", "2", "3", "4" or any valid combination for the receiver you are using such as "12". Default is "1".

unidirectional A Boolean. It specifies whether the map is unidirectional (True) or boustrophedonically[2] (False). Default is False.

start An integer. It specifies the starting column for the map. The default value for start is 1. "start" and "stop" are useful for doing a portion of a map or restarting a partially observed map.

stop An integer. It specifies the stopping column for the map. The default value for stop is None, which means "go to the end".

This example produces a boustrophedonic (bidirectional or back and forth) map with 21 columns each 12 arc–minutes long (along a great circle of Declination), using a spacing of 0.24 arc–minutes using beam 1. This map goes to a reference point 2 degrees north and 3 degrees east of the map center every 3 columns:

```
DecLatMapWithReference("ORIONKL",                              # center of map
    Offset("J2000", 4.8/60.0, 0.0, cosv = False) ,             # 4.8' wide
    Offset("J2000", 0.0, 12.0/60.0, cosv = False),             # 12' tall
    Offset("J2000", 0.24/60.0, 0.0, cosv = False),             # 0.24' stripe spacing
    Offset("J2000", 3.0, 2.0, cosv = False),                   # 2,2 deg ref offset
    3,                                                         # ref every 3rd column
    10.0)                                                      # 10 seconds per row
```

Note that the above example does not create a map which is "rectangular" when plotted in RA vs. Dec. You would have to use cosv = True to get a rectangular map.

DecLatMap

A Declination/Latitude map, or DecLatMap, does a raster scan centered on a specific location on the sky. It is similar to DecLatMapWithReference() except that it does not make periodic observations of a reference position.

Syntax: DecLatMap(location, hLength, vLength, hDelta, scanDuration, beamName, unidirectional, start, stop)

DecLatMap() does not have referenceInterval as a parameter, otherwise it is the same as DecLatMapWithReference(). See DecLatMapWithReference() for information on the parameters for DecLatMap().

[2]from the Greek meaning "as the ox plows" i.e. back and forth

PointMapWithReference

A PointMapWithReference() constructs a map by sitting on fixed positions laid out on a grid. This scan type allows the user to periodically move to a reference location on the sky. Please see PointMap() if no reference location is required. The starting point of the map is defined as (-hLength/2, -vLength/2).

Syntax: PointMapWithReference(location, hLength, vLength, hDelta, vDelta, referenceOffset, referenceInterval, scanDuration, beamName, start, stop)

The parameters for PointMapWithReference are

location A Catalog source name or Location object. It specifies the center of the map.

hLength An Offset object. It specifies the horizontal width of the map.

vLength An Offset object. It specifies the vertical height of the map.

hDelta An Offset object. It specifies the horizontal distance between points in the map.

vDelta An Offset object. It specifies the vertical distance between points in the map.

referenceOffset An Offset object. It specifies the position of the reference source on the sky relative to the Location specified by the first input parameter (i.e. the center of the map).

referenceInterval An integer. The number of points that should be completed in the map before moving to the reference offset position and to perform a reference scan, e.g. 4 means every four points.

scanDuration A float. It specifies the length of each scan in seconds.

beamName A string. It specifies the receiver beam to use for the scan. beamName can be “C”, “1”, “2”, “3”, “4” or any valid combination for the receiver you are using such as “12”. Default is “1”.

start An integer. It specifies the starting point for the map. The default value for start is 1. Note in PointMap this counts points, not stripes.

stop An integer. It specifies the stopping point for the map. The default value for stop is None, which means “go to the end”.

The following example does a 4x4 point map using beam “C” that moves to a reference source every 2 points:

```
PointMapWithReference("2023+2223",                    # map center location
    Offset("B1950", 1.50, 0.00, cosv=True), # 90 arcmin/cos(dec) size
    Offset("B1950", 0.00, 1.50, cosv=True), # 90 arcmin of deg size
    Offset("B1950", 0.50, 0.00, cosv=True), # 30 arcmin/cos(dec) step
    Offset("B1950", 0.00, 0.50, cosv=True), # 30 arcmin step spacing
    Offset("J2000", 3.00, 3.00, cosv=True), # offset reference dist
    2,                                       # reference every 2nd pnt
    2.0)                                     # 2 seconds per point
```

PointMap

A PointMap() constructs a map by sitting on fixed positions laid out on a grid. It is similar to PointMapWithReference() except that it does not make periodic observations of a reference position.

Syntax: PointMap(location, hLength, vLength, hDelta, vDelta, scanDuration, beamName, start, stop)

PointMap() does not have referenceInterval as a parameter, otherwise it is the same as PointMapWithReference(). See PointMapWithReference() for information on the parameters for PointMap().

RALongMapWithReference

A Right Ascension/Longitude (RALong) map performs a raster scan centered on a sky location. Scans are performed in the right ascension, longitude, or azimuth coordinate depending on the desired coordinate system. This scan type does allow the user to periodically move to a reference location on the sky; please see RALongMap for a map that does not use a reference. The starting point of the map is defined as (-hLength/2, -vLength/2).

Syntax: RALongMap(location, hLength, vLength, vDelta, referenceOffset, referenceInterval, scanDuration, beamName, unidirectional, start, stop)

The parameters for RALongMapWithReference are

location A Catalog source name or Location object. It specifies the center of the map.

hLength An Offset object. It specifies the horizontal width of the map (i.e., the extent in the longitude-like direction).

vLength An Offset object. It specifies the vertical height of the map (i.e., the extent in the latitude-like direction).

vDelta An Offset object. It specifies the distance between map rows. Note that vDelta values must be positive.

referenceOffset An Offset object. It specifies the position of the reference source on the sky relative to the Location specified by the first input parameter.

referenceInterval An integer. It specifies when to do a reference scan in terms of map rows, e.g. 4 means every fourth row.

scanDuration A float. It specifies the length of each scan in seconds.

beamName A string. It specifies the receiver beam to use for the scan. beamName can be "C", "1", "2", "3", "4" or any valid combination for the receiver you are using such as "12". Default is "1".

unidirectional A Boolean. It specifies whether the map is unidirectional (True) or boustrophedonically[3] (False). Default is False.

start An integer. It specifies the starting row for the map. The default value for start is 1. This is useful for doing parts of a map at different times. For example, if map has 42 rows, one can do rows 1-12 by setting "start=1, stop=12", and later finishing the map using "start=13, stop=42".

stop An integer. It specifies the stopping row for the map. The default value for stop is None, which means "go to the end".

This example produces a map with 6 rows each 120' long, using a spacing of 6' and scan rate of 720'/min that moves to a reference position every 7 rows:

```
RALongMapWithReferene("CygA",                       # center of map
    Offset("J2000", 2.0, 0.0, cosv=True),           # 120'/cos(dec) width
    Offset("J2000", 0.0, 0.5, cosv=True),           # 30' height
    Offset("J2000", 0.0, 0.1, cosv=True),           # 6' vertical spacing
    Offset("J2000", 4.0, 4.0, cosv=True),           # 4 deg/cos(dec) ref offset
    7,                                              # ref every 7th column
    10.0)                                           # 10 seconds per row
```

[3] from the Greek meaning "as the ox plows" i.e. back and forth

RALongMap

A Right Ascension/Longitude map, or RALongMap, does a raster scan centered on a specific location on the sky. It is similar to RALongMapWithReference() except that it does not make periodic observations of a reference position.

Syntax: RALongMap(location, hLength, vLength, vDelta, scanDuration, beamName, unidirectional, start, stop)

RALongMap() does not have referenceInterval as a parameter, otherwise it is the same as RALongMapWithReference(). See RALongMapWithReference() for information on the parameters to RALongMap().

Daisy

The Daisy scan type performs a scan around a central point in the form of daisy petals, also sometimes called a rose curve. Examples of these curves are shown in Figure 5.1.

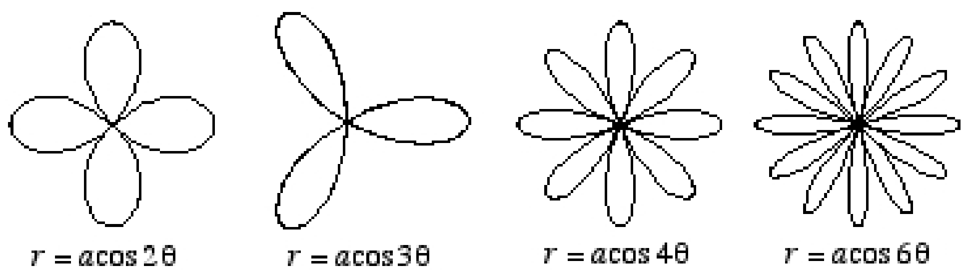


Figure 5.1: Examples of rose curves.

The area of the sky covered will be circular, with a diameter equal to twice the specified radius. For map radii of a few arc-minutes, a radial oscillation period of 60 seconds or longer is recommended; a scanDuration of 20 radial oscillation period's will result in an approximately closed pattern. For beam-sizes of 20 arcsec (FWHM) or so, the circular area mapped will be fully sampled if the map radius is less than 6'. It is not an especially useful observing mode for general-purpose single-beam mapping, since the largest "hole" in the map is approximately 0.3 x map_radius. However it is useful for focal-plane arrays.

Syntax: Daisy(location, map_radius, radial_osc_period, radial_phase, rotation_phase, scanDuration, beamName, cos_v, coordMode, calc_dt)

This command scans around a central point in the form of a daisy petal or a rose curve.

The following explanation is not as complete as one might like. Please refer to the MUSTANG chapter (Chapter 16, section 16.3.1, and section 16.6), or consult your GBT "friend".

The parameters to Daisy() are:

location A Catalog source name or Location object. It specifies the center of the map.

map_radius A float which specifies the radius of the map's "daisy petals" in arc-minutes. This parameter is equivalent to a in Figure 5.1.

radial_osc_period A float which specifies the period of the radial oscillation in seconds.

radial_phase A float which specifies the radial phase in radians.

rotation_phase A float which specifies the rotational phase in radians.

scanDuration A float. It specifies the length of the scan in seconds.

beamName A string. It specifies the receiver beam to use for both scans. beamName can be "C", "1", "2", "3", "4" or any valid combination for the receiver you are using such as "MR12" and "MR34". The default value for beamName is "1".

cos_v A boolean. It specifies whether secant minor corrections should be used for the major axis of the coordinate system. The default is True.

coordMode A string. It specifies the coordinate mode for the radius that generate the map. The default is "AzEl".

calc_dt A float. It specifies time sampling and should be between 0.1 and 0.5. The default is 0.1.

This example produces a three leaf map about "3C123":

```
Catalog("fluxcal")
Daisy(location="3C123", map_radius=5, radial_osc_period=60, radial_phase=0,
        rotation_phase=0, scanDuration=1200)
```

5.2.3 Catalogs

The Source Catalog system in Astrid provides a convenient way for the user to specify a list of sources to be observed, as well as a way to refer to standard catalogs of objects. At a minimum for each source there must be a name and a location (Ra/Dec or Glat/Glon, etc). Other parameters may be set, such as radial velocity. An example of a simple Catalog is:

```
# My source list
format=spherical
coordmode=J2000
HEAD = NAME    RA           DEC
Object1   09:56:16.98   +49:16:25.5
Object2   10:56:16.98   +50:16:25.5
Object3   11:56:16.98   +51:16:25.5
Object4   12:56:16.98   +52:16:25.5
```

There are four formats of catalogs:

SPHERICAL A fixed position in one of our standard coordinate systems, e.g., RA/DEC, AZ/EL, GLON/GLAT, etc.

EPHEMERIS A table of positions for moving sources (comets, asteroids, satellites, etc.)

NNTLE NASA/NORAD Two-Line Element (TLE) sets for earth satellites.

CONIC Orbital elements for solar system objects, such as comets and asteroids.

In addition, major solar system bodies may be referred to by name ("Sun", "Moon", "Mercury", "Venus", "Mars", "Jupiter", "Saturn", "Uranus", "Neptune", "Pluto"). These names are case-insensitive. They may be given to any Scan Type function ("Track", "RALongMap", etc). No catalog needs to be invoked for the system to understand planet names (and Pluto).

To use the catalog system, the user invokes the Catalog() command in her Observing Script, and passes the name of the desired object to any of the scan functions. All sources named in all the catalogs that have been invoked are available within an Observing Script. If the same name appears in two or more catalogs, the name from the most recently invoked catalog will prevail. Name comparisons are case-insensitive, hence "b2322+16" and "B2322+16" are equivalent.

5.2.3.1 Getting Your Catalog Into Astrid

Although one can include any number of Catalogs into an observing script, the standard practice is to put all the Catalogs into separate files that are then brought into the Observing Script via the "Catalog()" command. This: a) keeps the Observing Scripts simple and without clutter; and b) allows changes to be made to the Catalog without having to validate and re-save the Observing Scripts.

The best way to learn about how to bring Catalogs into the Observing Script is through an example. Lets suppose that there are two Catalogs that you need for your observations. These two catalogs are in the following files:

```
/home/astro-util/projects/6D01/sources.cat
/home/astro-util/projects/6D01/pointing.cat
```

To bring these Catalogs into your Observing Script, you would need the following lines in your Observing Script:

```
#first load the catalog with the flux calibrators
cata=Catalog("/home/astro-util/projects/6D01/sources.cat")
#
#now load the catalog with the pointing source list
catb=Catalog("/home/astro-util/projects/6D01/pointing.cat")
```

All sources from all catalogs are available and referenced by name within the scope of the Observing Script, with the exception that for duplicate source names only the last entry of that name will be recognized. After loading a Catalog any scan function may be run by giving it the source name, for example:

```
Track("TMC-1", 60)
Nod("ORI-KL", 120)
Slew("SGR-B")
```

and so forth.

5.2.3.2 The Format of the Catalog

A Catalog typically has two sections: a header section followed by a table of information for all the sources. The header section consists of the "KEYWORD = VALUE" pairs. The "KEYWORD = VALUE" pairs tell the Scheduling Block interpreter how to read the information in the table section of the Catalog. Once a keyword value is given, its value will persist until re-set or the end of the Catalog is reached. The keywords are case-insensitive. The values for a keyword must not contain any embedded blanks (except source names in NNTLE and CONIC formats).

A Catalog can contain comments with the beginning of a comment being denoted by the hash symbol, "#". All information on a line after the hash symbol is considered to be part of the comment. After the header, each source in the Catalog occupies a single line. You should not use the hash symbol, "#", in source names. Here is an example of a simple catalog:

```
# My source list
format=spherical
coordmode=J2000
HEAD = NAME    RA           DEC
Object1   09:56:16.98   +49:16:25.5
Object2   10:56:16.98   +50:16:25.5
Object3   11:56:16.98   +51:16:25.5
Object4   12:56:16.98   +52:16:25.5
```

Catalog Header Keywords

Catalog Header Keywords are used to define how the catalog entries should be read. The keywords and their values are case insensitive.

The following example will be used to describe some of the Catalog Header Keywords.

```
# My source list
format=spherical
HEAD = NAME  COORDMODE  RA          DEC       RESTFREQ    VELDEF      VEL    type
Object1   J2000   09:56:16.9   +49:16:25   1420.405   VRAD-LSR   -25.3   HII
"Src A"   B1950   10:56:16.9   +50:16:25   1665.401   VOPT-BAR   100.9   Gal
```

FORMAT This tells the type of catalog and must be the first line in any catalog. Possible values are "spherical", "ephemeris", "nntle" and "conic". For the SPHERICAL format, the first line would contain "FORMAT=SPHERICAL". This is the default format, hence the "FORMAT=SPHERICAL" may be omitted.

HEAD This gives the header for tabular data, and consists of a list of any keywords. This should appear as the last line in the header before lines giving information about the sources in the catalog. You can also create your own header keyword, such as the "type" column in the above example. The default header is "HEAD = NAME RA DEC VELOCITY". In the above example we have added more entries than the default. We have also created a new keyword named "type".

NAME The source name is any string up to 32 characters long. The name as given in the catalog does not have to be surrounded by quotes unless it contains embedded blanks or hashes (#). In the above example we needed to use quotes around "Src A" because of the space in the name.

COORDMODE The default is J2000. Possible values are: J2000, B1950, JMEAN (mean coordinate of date given by EQUINOX), GAPPT (geocentric apparent coordinates of date), GALACTIC, HADEC, AZEL, ENCODER. In the above example we put the COORDMODE keyword in the HEAD line since we have sources whose positions are given in different coordinate modes (J2000 and B1950).

VEL or VELOCITY The radial velocity in km/sec. The Default is to use any previous setting or 0.0 if there is none.

VELDEF Velocity definition in the FITS convention (see https://safe.nrao.edu/wiki/bin/view/GB/Data/VelDefFits), e.g. "VOPT-BAR", "VRAD-LSR", etc. The default is the velocity definition or reference frame that was previously set. In the above example we put the VELDEF keyword in the HEAD line since we have sources whose velocity definitions are different.

RESTFREQ The rest frequency, in MHz. The default is to use the previous setting. Again we put the RESTFREQ keyword in the HEAD line since we are defining two different spectral line rest frequencies for each source.

RA, HA, DEC, AZ, EL, GLON, GLAT A pair of coordinates must be given : RA/DEC, HA/DEC, AZ/EL, or GLON/GLAT. Angle formats may be either in sexegesimal with colons (e.g. dd:mm:ss.ss) or in decimal format. RA and HA are in hours, all other angles are in degrees.

EQUINOX Used if the Coordmode is “JMEAN”. The value is a float (e.g. 2006 December 1, 12:00 UT would be 2006.919178082192).

An example of the use of the EQUINOX Catalog Header Keyword is:

```
# My source list
format=spherical
coordmode=jmean
equinox=2007.123456
HEAD = NAME      RA          DEC
Object2   10:56:16.98   +50:16:25.5
```

Additional keywords used when the Ephemeris format is active are (see § 5.2.3.6 for examples):

DATE The UTC date, either “2005-06-23” or “2005-Jun-23” form.

UTC The UTC time in the form “hh:mm:ss”.

DRA, DHA, DDEC, DAZ, DEL, DLON, DLAT The coordinate rate keywords given in arcseconds per hour.

DVEL The radial velocity rate in km/sec/hour.

Additional keywords used by NNTLE and CONIC formats are (see § 5.2.3.7 and § 5.2.3.8 on NNTLE and CONIC formats below for examples):

FILE For use in NNTLE and CONIC formats only. This keyword value may refer to a file or a URL containing a 2-line element set.

USERADVEL For use in the NNTLE format only. If this is set to 1, then the radial velocity tracking will be performed. Otherwise, if this is set to 0 or is missing then radial velocity tracking will not be performed.

5.2.3.3 SPHERICAL format Examples

Here is an example of a simple catalog.

```
# My source list
format=spherical
coordmode=J2000
HEAD = NAME    RA          DEC
Object1   09:56:16.98   +49:16:25.5
Object2   10:56:16.98   +50:16:25.5
Object3   11:56:16.98   +51:16:25.5
Object4   12:56:16.98   +52:16:25.5
```

Because all the keyword values use the defaults, the following is equivalent:

```
# My source list
Object1   09:56:16.98   +49:16:25.5
Object2   10:56:16.98   +50:16:25.5
Object3   11:56:16.98   +51:16:25.5
Object4   12:56:16.98   +52:16:25.5
```

Here is an example catalog that specifies the radial velocities of the sources.

```
# My source list with radial velocities
format=spherical
coordmode = B1950
head = name    ra           dec          velocity
Object1   09:56:16.98   +49:16:25.5   27.23
Object2   08:56:16.98   +48:16:25.5   28.24
Object3   07:56:16.98   +47:16:25.5   29.25
Object4   06:56:16.98   +45:16:25.5   30.26
```

Here is an example Catalog where one may omit the "format=" line, but not the "coordmode=" line.

```
# A list of HII regions
coordmode=Galactic
head= NAME  GLON        GLAT        vel       restfreq
G350+.07    350.107     +0.079      42.2235   9816.867
G351+.17    351.613      0.172     -15.553    9487.824
G352-.17    352.393     -0.176     -52.227    9173.323
G352-.36    353.4219    -0.3690     22.335    9487.824
```

Warning: setting the velocity or rest frequency in a catalog only changes the values in the First LO (LO1) manager. If either value is changed by a large amount, the receiver selection or bandpass filters or the frequency spacing between spectral windows may change. Thus one should re-configure the IF for a large change in velocity or frequency. The user should be wary of how much the velocity or rest frequency can change for a particular configuration.

Finally we show an example Catalog with user-defined keywords. The user may make up arbitrary keywords (or equivalently column headings). These are available within an Observing Script, but are otherwise ignored.

```
# a list of pointing references
format=spherical
coordmode=j2000
head= name      ra              dec            BMIN BMAX  S20    S6
 0011-1434  00:11:40.40     -14:34:04.7     15   45    0.17   0.20
 0012-3321  00:12:17.96     -33:21:57.8     15  180    0.85   0.18
 0012+6551  00:12:37.80     +65:51:10.5     15  360    1.20   0.55
 0012+2702  00:12:38.14     +27:02:40.7     15  180    0.60   0.21
 0012+3353  00:12:47.3826   +33:53:38.459    0   45    0.08   0.08
 0012-3954  00:12:59.9080   -39:54:25.836    0   45    0.49   1.5
```

5.2.3.4 Standard Catalogs

Several "standard" catalogs are available for use within the Green Bank computing system. They are all ASCII files in the directory /home/astro-util/astridcats/ and are listed in Table 5.7.

Table 5.7: The following Catalogs are present as of 07 Nov 2006. For the pointing catalogs, users should be warned that the flux densities at higher frequencies are extrapolated from measurements at 8 GHz and below.

Catalog	Description
fluxcal.cat	Calibrators with well-determined flux densities. U. S. Government Printing Office (Usgpo) 2006, The Astronomical Almanac for the year 2006, Washington: U.S. Government Printing Office (USGPO), 2006, U.S. Naval Observatory (USNO), Royal Greenwich Observatory (RGO).
pointing.cat	Condon's master pointing catalog for the GBT. https://safe.nrao.edu/wiki/bin/view/GB/PTCS/PointingFocusCatalog
lband_pointing.cat	Extracted from pointing catalog for 21 cm band (1.4GHz).
cband_pointing.cat	Extracted from pointing catalog for the 6 cm band (6GHz).
xband_pointing.cat	Extracted from pointing catalog for the 3.5 cm band (9GHz).
kuband_pointing.cat	Extracted from pointing catalog for the 2 cm band (14GHz).
kband_pointing.cat	Extracted from pointing catalog for the 1.5 cm band (20GHz).
kaband_pointing.cat	Extracted from pointing catalog for the 9 mm band (32GHz).
qband_pointing.cat	Extracted from pointing catalog for the 7 mm band (43GHz).
HI_strong.cat	Galaxies with strong HI lines, extract from Rich Fisher's database. http://www.gb.nrao.edu/~rfisher/GalaxySurvey/galaxy_survey.html
pulsars_all.cat	All 1533 pulsars in the ATNF database as of 26 Aug 2005.
pulsars_all_GBT .cat	All 1054 pulsars visible from Green Bank.
pulsars_brightest_GBT .cat	The brightest pulsars, visible from Green Bank.
pulsars_bright_MSPs_GBT .cat	Bright millisecond pulsars visible from Green Bank.

5.2.3.5 Catalog Functions

Two useful catalog functions are available.

c.keys() Acts like a python function that returns a list of all the source names in the Catalog loaded into the variable "c" [i.e. via c=Catalog("mycatalog")].

c["sourcename"]["keyword"] Returns the value of the keyword for the named source in the Catalog loaded into the variable "c". This function can be used to pass information in the Catalog on to the Observing Script (e.g. specifying different map sizes for different sources/directions).

The c.keys() function can be used so that the Observing Script will automatically loop through all the sources in a Catalog. Here is an example of how to do this:

```
c = Catalog("/home/astro-util/astridcats/HI_strong.cat")
sourcenames = c.keys()
for s in sourcenames :
    Nod(s, 120)
```

The c["sourcename"]["keyword"] function can be used to get information out of the "keyword" column of the Catalog for use within the Observing Script. In the following example we get the source's Declinations and only observe those sources above 20° Declination (note that the coordinates are always returned in degrees):

```
c = Catalog("/home/astro-util/astridcats/lband_pointing.cat")
sourcenames = c.keys()
for s in sourcenames :
    print c[s]["dec"]
    if c[s]["dec"] > 20 :
        Nod(s, 120)
```

The c[“sourcename”][“keyword”] function can also be used to execute more complicated observing strategies. In the following example we have many sources to observe and we desire a different amount of total integration time for each source. To accomplish this we add two new columns to the Catalog. We will call these columns “sourcetime” and “status”. A few lines of the Catalog (lets call it mycatalog.cat) would look like:

```
head= name ra dec velocity sourcetime status
SrcA  00:01:02  -03:04:05  -22.0  300  done
SrcB  06:07:08  +10:11:12  +56.3  120  waiting
```

The Observing Script would look like:

```
c = Catalog("mycatalog.cat")
sourcenames = c.keys()
for s in sourcenames :
    if c[s]["status"] == "waiting"
        dwelltime = float(c[s]["sourcetime"])
        Track(s,None,dwelltime)
```

Note that we have to convert the value that c[s][“sourcetime”] returns from a string into a float before it can be used in the Observing Script.

5.2.3.6 EPHEMERIS format : Tables for moving objects

A Catalog can also be used as an Ephemeris for the position of a moving object, such as a comet or asteroid. To make the Catalog into an Ephemeris the first non-comment line of the Catalog must contain:

```
FORMAT = EPHEMERIS
```

The header of the Catalog for an Ephemeris can also contain the NAME, COORDMODE, VELDEF and HEAD keywords. The “data lines” in the Catalog must contain at least the date, the time, and a pair of coordinates for an Ephemeris. Optional parameters are coordinate rates, radial velocity and radial velocity rate. User-defined parameters may also be added. The dates and times are required to be in UTC. The dates and times can be specified in any legal python form, for example: a) ’YYYY-MM-DD hh:mm:ss’ where MM is month number (e.g August = 09); or b) ’YYYY-MMM-DD hh:mm:ss’ where MMM is the abbreviated month name such as Jan, Feb, etc.

The ephemeris table should contain enough entries to cover a period longer than that required by a particular observing session. The observing system selects the portion of the table needed for the current scan start time and duration.

Here is an example of a valid ephemeris file:

```
#-----------------------------------------------------------------
# Richard's sample ephemeris catalog
#-----------------------------------------------------------------
```

```
#
FORMAT = EPHEMERIS
#
NAME     = MyMovingObject
COORDMODE = J2000
VELDEF = VRAD-LSR
2004-07-16 00:10:00  09:56:16.98  +49:16:25.5  27.234234
2004-07-16 00:20:00  09:56:17.76  +49:16:36.2  27.456345
2004-07-16 00:30:00  09:56:18.55  +49:16:46.9  27.568233
2004-07-16 00:40:00  09:56:19.32  +49:16:57.6  27.623423
2004-07-16 00:50:00  09:56:20.10  +49:17:08.3  27.723456
#------------------------------------------------------------------------
```

Note that the “HEAD=” line has been omitted because the default is “DATE UTC RA DEC VEL”.

Here is a more complicated example for a comet that specifies the coordinate rates:

```
FORMAT=EPHEMERIS
NAME=C2002T7
COORDMODE=GAPPT
VELDEF=VRAD-TOP
HEAD= date utc ra dec dra ddec vel
2004-Jun-13 16:15:00 09:46:23.35 -09:15:41.80 26.73 8.02  63.757540
2004-Jun-13 16:20:00 09:46:23.89 -09:15:39.40 26.71 8.01  63.758070
2004-Jun-13 16:25:00 09:46:24.44 -09:15:36.90 26.69 8.01  63.758760
2004-Jun-13 16:30:00 09:46:24.98 -09:15:34.50 26.68 8.01  63.759620
2004-Jun-13 16:35:00 09:46:25.52 -09:15:32.10 26.66 8.01  63.760650
2004-Jun-13 16:40:00 09:46:26.06 -09:15:29.70 26.64 8.01  63.761850
2004-Jun-13 16:45:00 09:46:26.60 -09:15:27.30 26.62 8.01  63.763220
2004-Jun-13 16:50:00 09:46:27.14 -09:15:24.90 26.61 8.00  63.764740
```

Here is an example for tracking a satellite:

```
# PRN14 tracking table (angles in degrees)
# visible 01:30 to 3:00 UT
format=ephemeris
name=PRN14
coordmode=azel
head=date      utc       az         el
2004-05-16 01:30:06  103.1822   43.0174
2004-05-16 01:30:14  103.2464   42.9721
2004-05-16 01:30:22  103.3105   42.9268
2004-05-16 01:30:30  103.3745   42.8814
```

5.2.3.7 NNTLE : tracking earth satellites

“NNTLE” stands for NASA/NORAD Two-Line Elements. This refers to a standard NASA format for orbital elements for Earth satellites (see e.g. http://ghrc.msfc.nasa.gov/orbit/tleformat.html or http://www.amsat.org/amsat/keps/formats.html)

The first non-comment line of the Catalog must contain:

```
FORMAT = NNTLE
```

If the FILE keyword is used then one should only give the name of the object in the Catalog as the elements of the orbit are retrieved from the file or URL. Note that the full path name of the file must be given, and the file must have world read permission.

The remainder of the non-comment lines contain the names for one or more satellites and their orbital elements in the NASA/NORAD Two-Line Element format.

An example of a valid file is as follows (data taken from the AMSAT URL listed above):

```
# Richard's sample nntle catalog
#------------------------------------------------------------
FORMAT = NNTLE
USERADVEL = 1      # optional keyword
#
OSCAR10
1 14129U          88230.56274695 0.00000042           10000-3 0   3478
2 14129  27.2218 308.9614 6028281 329.3891    6.4794  2.05877164 10960
GPS-0008
1 14189U          88230.24001475 0.00000013                   0   5423
2 14189  63.0801 108.8864 0128028 212.9347 146.3600  2.00555575 37348
#----------------------------------------------------------------------------
```

When implementing a NNTLE catalog, the scantype function will pass the 3 lines to a program that will calculate positions for the antenna, given the scan start time and duration. The source name is the string that appears on the first of the three lines, and that is what one would pass to the scan function.

Because it may be convenient to download a Two-Line Element (TLE) file from NASA[4], we provide an option to use such a TLE file "as is" by using the "FILE" keyword as shown in the following example.

An example of this format follows:

```
# NNTLE catalog referring to an unedited TLE file
#-------------------------------------------------------
FORMAT=NNTLE
USERADVEL=0
#
FILE= /users/fghigo/tlecatalogs/goes.txt
Name = "GOES 6"
Name = "GOES 10"
Name = "GOES 11"
```

The first set of orbital elements whose name matches the name listed in the file will be used for calculating the satellite position.

Note that the generation of tracks for satellites is based on "pyephem", an implementation of xephem in Python.

5.2.3.8 CONIC : tracking solar system objects

The CONIC format is used for solar system objects whose elements are given in "xephem" format. For examples, see http://www.minorplanetcenter.org/iau/Ephemerides/Comets/Soft03Cmt.txt.

The first non-comment line of the catalog must contain:

```
FORMAT = CONIC
```

[4]One might load a TLE catalog through the web site http://www.celestrak.com/NORAD/elements/

The use of the FILE keyword is similar to that for NNTLE and may refer to a file or a URL containing orbital elements in "xephem" format. In this case only the name of the object should be given. Note that the full path name of the file must be given, and the file must have world read permission.

Here is an example where the orbital elements are given in the catalog:

```
#------------------------------------------------------------------------
FORMAT=CONIC
# From MPC 44030
P/2000 Y3 (Scotti),e,2.251,354.896,87.424,5.0129,0.08781,0.194783,154.80,08/18.0/2005,2000,g  9.0,4.0
# From MPC 44182
C/2001 B2 (NEAT),e,150.657,145.157,305.014,3436,0.0000049,0.998449,0.0000,08/31.7355/2000,2000,g  4.0,4.0
# From MPC 45961
C/2001 C1 (LINEAR),h,03/29.1087/2002,68.9555,33.6590,220.0235,1.001431,5.107184,2000,6.0,4.0
# From MPC 45334
C/2001 G1 (LONEOS),h,10/11.7349/2001,45.3242,203.8746,343.4583,1.004918,8.245451,2000,3.5,4.0
#------------------------------------------------------------------------
```

In this example the orbital elements are found in a file retrieved from the web.

```
FORMAT=CONIC
FILE=http://www.minorplanetcenter.org/iau/Ephemerides/Comets/Soft03Cmt.txt
name="8P/Tuttle"
name="103P/Hartley"
NAME = "P/2000 Y3"
NAME= "C/2001 B2"
```

5.2.4 Utility Functions

Utility functions are used in Observing Scripts to control various aspects of the GBT other than data-taking scans. This includes such things as balancing the IF, pausing the Observing Script, or waiting for a source to rise. Please note that the syntax for all utility functions is case-sensitive.

Advanced utility functions are found in Appendix F.

5.2.4.1 Annotation()

The Annotation() function allows you to add any keyword and value to the GBT Observing (GO) FITS file. This could be useful if there is any information you would like to record about your observation for later data processing, or for record keeping. The syntax is

```
Annotation("KEYWORD","Value")
```

where KEYWORDS must be written completely in capital letters and can be no longer than eight characters in length.

An example use of the Annotation() function is if you wish to specify what type of source you are observing. Your sources might include H II regions and Planetary Nebulae for example. You could specify each type with

```
Annotation("SRCTYPE ","HII")
Annotation("SRCTYPE ","PNe")
```

The information in a FITS KEYWORD created via the Annotation() function will be ignored by the standard GBT data reduction package GBT IDL Data Reduction Package (GBTIDL).

5.2.4.2 Balance()

Balancing changes the various attenuator levels and gain levels in the GBT receivers, the IF and the back-ends so that each device is operating in its linear response regime. The Balance() function is used to balance the electronic signal throughout the GBT IF system. The Balance() function will work for any device with attenuators and for a particular backend. Indivdual devices can be balanced, such as the Prime Focus receivers, the IF, the GBT Spectrometer, and/or the Spectral Processor. (The Gregorian receivers lack attenuators and do not need to be balanced.) If the argument to Balance() is blank, then all devices for the current state of the IF system will be balanced.

Without any arguments, the Balance() command uses the last executed configuration to decide what hardware will be balanced. Strategies for balancing the hardware in the GBT IF are discussed in § 6.1.

An simple example of the use of Balance() is

```
Configure(myconfiguration)
Balance()
```

For more details about the Balance command, refer to Section 6.1 and Appendix G.

5.2.4.3 Break()

The Break() function inserts a breakpoint into your Observing Script and gives the observer the choice of continuing or terminating the Observing Script. When a breakpoint is encountered during execution, your Observing Script is paused and a pop-up window is created. The Observing Script remains paused for a set amount of time or until you acknowledge the pop-up window and tell Astrid to continue running your script.

The Break() function can take two optional arguments, a message string and a timeout length. Why have a timeout? If an observer walks away from the control room during his or her observing session (e.g. to go to lunch or the bathroom) and a breakpoint is reached, it would be counterproductive to pause the observation indefinitely. This will help to save valuable telescope time.

The full syntax for the Break function is Break(message, timeout) where "message" is a string that is displayed in the pop-up dialog with a default of "Observation paused" and timeout is a float that determines how long the system waits (in seconds) to get user-input before continuing the Observing Script. The default for timeout is 300 seconds, or 5 minutes. If you wish for the timeout to last forever then use None [i.e. Break("Wait Forever", None) or Break(timeout=None)].

Here are examples of the use of Break():

```
Break("This will time out in 5 minutes, the default.")
Break("This will time out after 10 minutes.",600)
Break("This will never time out.", None)
```

5.2.4.4 Comment()

The Comment() function allows you to add a comment into the Astrid observing process which will be echoed to the observation log during the observation. The syntax is Comment("Text to display during the observation"). What's the difference between this, and just writing comments with the pound (#) sign in your Observing Script? When you use the pound sign to write your comments, they will not appear in the observation log when your Observing Block is run. Using the Comment() function directs your comment to the output in the observation log.

Here is an example of the use of Comment():

```
# now slew to the source
Comment("Now slewing to 3C 286")
Slew("3C286")
```

5.2.4.5 GetUTC()

The GetUTC() function returns a float representing the current time in UTC. The returned value is the decimal hours since midnight.

An example of the use of GetUTC() is

```
while GetUTC() < 12.0:
    Track("0353+2234",None,600.)
```

which will repeatedly perform Track scans until the UTC time is past 12.0 hours.

5.2.4.6 GetLST()

The GetLST() function returns a float representing the current Local Sidereal Time at the time of execution. The returned value is the decimal hours.

An example of the use of GetLST() is

```
while GetLST() < 13.5:
    Track("1153+1107",None,600.)
Track("1712+036",None,600.)
```

which will repeatedly observe the source "1153+1107" until the LST is past 13.5 hours when the source "1712+036" will be observed once.

5.2.4.7 Now()

The Now() function returns the current time as a UTC time object (see § 5.2.5.4) containing the UTC time and date. An example of the usage of Now() is

```
while Now() < "2006-03-12 09:54:12" and Now() != None:
    Track("1153+1107",None,600.)
```

which repeatedly performs Track scans of the source 1153+1107 until 09:54:12 UTC on 12 March 2006. Note that the while statement also checks that the returned value of Now() is not None. This is to ensure that Astrid does not get stuck in an infinite loop during validation when Now() will always return a value of None.

5.2.4.8 WaitFor()

The WaitFor(time) function pauses the scheduling block until the specified time is reached, e.g.,

```
WaitFor("15:13:00 LST")
```

The expected wait time is printed in the observation log including a warning if the wait is longer than 10 minutes. WaitFor() will immediately return if the specified time has already passed and is within the last 30 minutes. While WaitFor has the Observing Script paused, it does not prevent the user from aborting a script. However if the user chooses to continue once the abort is detected, then the WaitFor abandons the wait and returns immediately.

Using a Horizon object (defined in § 5.2.5.3), WaitFor() can be used to have an Observing Script wait for a source to rise above an elevation of 10°,

```
WaitFor(Horizon(10.0).GetRise("1532+3421"))
```

or to wait for a source to set

```
WaitFor(Horizon(10.0).GetSet("sun"))
```

If WaitFor's argument is None then it aborts with an error message to the observation log. This can occur when the commands GetRise() or GetSet() detect an event which will never occur, such as the rise time for a circumpolar source.

5.2.4.9 ChangeAttenuation()

ChangeAttenuation() allows the observer to change all the attenuators in the IF Rack or the Converter Rack by the same ammount.

ChangeAttenuation takes two arguments.

devicename A string that can be either "IFRack" or "ConverterRack". This specifices the device in which the attenuators will be changed.

attnchange A float. This specifies how much the attenuators should be changed. This value can be either positive or negative.

It should be noted that if any new attenuator setting is less than zero or exceeds the maximum value, 31 for the IF Rack and 31.875 for the Converter Rack, then the attenuator settings is made to be the appropriate limiting value.

Examples of the usage of ChangeAttenuation() are:

```
ChangeAttenuation('IFRack', 1.0)
ChangeAttenuation('ConverterRack','-0.5')
```

5.2.5 Observing Script Objects

Observing Script Objects are python objects that are used to contain multiple pieces of information within a single variable. These are used with positions (requiring a major and minor axis value along with an epoch), times (requiring the date and the time of day), and for defining a horizon for the minimum elevation below which you would not want to observe.

5.2.5.1 Location Object

A Location object is used to represent a particular location on the sky. Locations can be specified in the following coordinate modes: “J2000”, “B1950”, “RaDecOfDate”, “HaDec”, “ApparentRaDec”, “Galactic”, “AzEl”, and “Encoder”. A Location is specified by two values, the meanings of which are dependent on the coordinate mode chosen. e.g. For J2000, the two values are time and degrees. location = Location(”J2000”, ”16:30:00”, ”47:23:00”)

Coordinates may be given in sexagesimal as a quoted string (i.e. ”hh:mm:ss.ss”) or as a floating point number in degrees.

For more information about Locations, see Appendix D. Also see Appendix E for angle formats and units.

5.2.5.2 Offset Object

An Offset is a displacement from the position of a source or from the center position of a map. Offsets can be specified in the following coordinates: “J2000”, “B1950”, “RaDecOfDate”, “HaDec”, “ApparentRaDec”, “Galactic”, “AzEl”, and “Encoder.” An Offset is specified by two values – generically called h and v with h being the value of the offset for the major axis and v being the offset value for the minor axis. For example h refers to Right Ascension and v refers to Declination in J2000 coordinates. Values can be entered in sexegesimal notation in quotes, or decimal degrees. Also, the user can specify whether the cos(v) correction should be taken into account for the Offset. The correction is applied when “cosv = True” (i.e. $h/\cos(v)$ is the offset value used in the direction of h), “cosv = False” means that the correction is not applied. The default value for cosv is True. Offsets may be added together and may also be added to Locations if their coordinate modes are the same. See Appendix D and Appendix E for more information about Locations and Offsets.

Here is an example of how to specify an Offset:

```
myoffset = Offset("J2000", "00:30:00", "05:00:00", cosv = False)
```

This offset is defined in J2000 coordinates. The offset is 30 minutes in Right Ascension and 5 degrees in Declination. See Appendix E for more information on angle formats and units in defining an Offset object.

5.2.5.3 Horizon Object

Observing Scripts allow an observer to specify a definition of the horizon. The user defined horizon can be used to begin an observation when an object “rises” and/or end the observation when it “sets” relative to the specified elevation of the “horizon”. The Horizon object may be used to obtain the initial time that a given source is above the specified horizon (including an approximate atmospheric refraction correction). The horizon object is created via

```
myhorizon = Horizon(10.0)
```

In this example a horizon that is at 10 degrees elevation is created. If no argument is given to Horizon, it will assume a default of elevation of 5.25 degrees (the nominal GBT horizon limit).

Any Horizon object may be substituted as a start or stop time in scan types, such as Track(). The rise and set times for any sky location in spherical coordinates may be obtained as a UTC time and date (see § 5.2.5.4). horizon.GetRise(source) will return the nearest rise time and horizon.GetSet(source) will return the next set time of the source. For example, to display the rise time of “0616-1041” (the previous rise time if the source is above the horizon or the next rise time if the source is below the horizon):

```
print Horizon().GetRise("0616-1041")
```

For northern circumpolar sources which never set, horizon.GetRise(source) returns the current time and horizon.GetSet(source) returns "None". For southern circumpolar sources which never rise, horizon.GetRise(source) returns "None" and horizon.GetSet(source) returns the current time.

Note that Horizon only works for objects defined in catalogs with spherical coordinates. Horizon will not work with planets and ephemeris tables.

5.2.5.4 Time Object

The Time Object is primarily used for defining scan start or stop times. The time may be represented as either a sexegesimal string or in a python mxDateTime object. You can learn more about mxDateTime at http://www.egenix.com/files/python/mxDateTime.html [5].

The Time Object can be expressed in either UTC or LST. The time can be either absolute or relative. An absolute or dated time specifies both the time of day and the date. An absolute time may be represented by either a sexegesimal string, i.e., "yyyy-mm-dd hh:mm:ss" or by a DateTime object. Relative or dateless times are specified by the time of day for "today". "WaitFor" will treat a dateless time that is more than 10 minutes in the past as being in the future, i.e., the next day. Relative times may be represented by either a sexegesimal string, i.e., "hh:mm:ss" or a DateTimeDelta object.

For UTC times, the sexegesimal representation may include a "UTC" suffix. Note that mxDateTime objects are always UTC. LST time may only be used with relative times and the sexegesimal representation must include a "LST" suffix.

Time Objects can have slightly varying formats and can be created in a few different ways. Some examples are:

"2006-03-22 15:34:10" Absolute time in UTC represented by a string.

DateTime.TimeDelta(12, 0, 0) Relative time in UTC as a mxDateTime object.

"2006/03/22 15:34:10 UTC" Absolute time in UTC represented by a string.

"22:15:48 LST" Relative time in LST as a string.

DateTime.DateTime(2006, 1, 21, 3, 45, 0) Absolute time in UTC as a mxDateTime object.

In this example we will continue to do one minute observations of srcA until Feb 12, 2007 at 13:15 UTC when we will then do a ten minute observations of srcB.

```
from mx import DateTime
emptyline
switchTime=DateTime.DateTime(2007,2,12,13,15,0) # Feb 12, 2007, 13:15 UTC
emptyline
while Now() < switchTime and Now() != None:
    Track(srcA,None,60)
emptyline
Track(srcB,None,600)
```

[5] Note, one must access the python DateTime module directly from an observation script to generate time objects, i.e., using mx import DateTime.

5.2.6 Example Observing Scripts

For the following observing script examples we will use the configuration examples from § 5.2.1.3. We will use a Catalog that is as follows:

```
# My source list with radial velocities
format=spherical
coordmode = B1950
head = name   ra          dec          velocity
Object1  09:56:16.98  +49:16:25.5  27.23
Object2  08:56:16.98  +48:16:25.5  28.24
Object3  07:56:16.98  +47:16:25.5  29.25
Object4  06:56:16.98  +45:16:25.5  30.26
```

The configuration file will be in /home/astro-util/projects/6D01/configurations.py and the Catalog file will be in /home/astro-util/projects/6D01/sources.cat (these files exist and are available within the Green Bank computing environment).

5.2.6.1 Frequency Switched Observations Looping Through a List of Sources

In this example we perform frequency switched observations of the HI 21 cm line towards several different sources. This example is available as /home/astro-util/projects/6D01/example-one.py

```
# Frequency Switched Observations where we loop through a list of sources
#
# first we load the configuration file
execfile('/home/astro-util/projects/6D01/configurations.py')
#
# now we load the catalog file
c = Catalog('/home/astro-util/projects/6D01/sources.cat')
#
# now we configure the GBT IF system for frequency switch HI observations
Configure(spectral_fs_config)
#
# now we balance the IF system
Balance()
#
# now we use a Break() so that we can check the IF system
Break('Check the Balance of the IF system')
#
# get the list of sources
sourcenames = c.keys()
#
# now loop the sources and observe each one for 10 minutes
for srcs in sourcenames:
    Track(srcs,None,600.)
```

5.2.6.2 Position Switched Observations Repeatedly Observing the Same Source

In this example we perform position switched observations of a single source. We observe the source for two minutes and the off position for two minutes. This is repeated twenty times. This example is available as /home/astro-util/projects/6D01/example-two.py from any computer within the Green Bank network.

```
# Position Switched Observations where we repeatedly observe the same source
#
# first we load the configuration file
execfile('/home/astro-util/projects/6D01/configurations.py')
#
# now we load the catalog file
Catalog('/home/astro-util/projects/6D01/sources.cat')
#
# now we configure the GBT IF system for frequency switch HI observations
Configure(spectral_ps_config)
#
# now we balance the IF system
Balance()
#
# now we use a Break() so that we can check the IF system
Break('Check the Balance of the IF system')
#
# specify which source we wish to observe
srcs = 'Object1'
#
# specify how far away from the source the off position should be
# offset two minutes of time in Right Ascension
myoff=Offset("J2000","00:02:00",0.0)
#
# specify how many times to observe the source
numobs = 20
#
# observe ``on'' source for 2 minutes and ``off'' source for 2 minutes
# and then repeat
for i in range(numobs):
    OnOff(srcs,myoff,120.)
```

5.2.6.3 Position Switched Observations of Several Sources and Using the Horizon Object

In this example we perform position switched observations of three sources. We observe the first source until the second source rises above 20 degrees elevation. Then we observe the second source until it goes below 20 degrees elevation at which point we observe a third source. This example is available as /home/astro-util/projects/6D01/example-three.py from any computer within the Green Bank network.

```
# Position Switched Observations where we observe the first source
# until the second source rises, and then we observe a third source
# after the second source sets
#
```

```
# first we load the configuration file and the catalog file
execfile('/home/astro-util/projects/6D01/configurations.py')
Catalog('/home/astro-util/projects/6D01/sources.cat')
#
# now we configure the GBT IF system for frequency switch HI observations
Configure(spectral_ps_config)
#
# now we balance the IF system and use a Break() to check the IF system
Balance()
Break('Check the Balance of the IF system')
#
# specify that the off position should be offset 2 min of time in RA
myoff=Offset("J2000","00:02:00",0.0)
#
# define the horizon to use - 20 degrees elevation in this case
h=Horizon(20.0)
#
# define which sources to observe
srcA = 'Object4'
srcB = 'Object3'
srcC = 'Object1'
#
# now get rise and set times of srcB
riseSrcB = h.GetRise(srcB)
setSrcB = h.GetSet(srcB)
#
# observe srcA until srcB has risen above 20 deg elevation
while Now() < riseSrcB and Now() != None:
    OnOff(srcA,myoff,120.)
#
# now observe srcB until it sets
while Now() < setSrcB and Now() != None:
    OnOff(srcB,myoff,120.)
#
# now observe srcC five times
numobs=5
for i in range(numobs):
    OnOff(srcC,myoff,120.)
```

To print the rise and set times in the above example you would just need to add

```
# print the rise and set times to the log using Comment()
risesetstring = "20 deg elev. rise = %s and set = %s" % (riseSrcB,setSrcB)
Comment(risesetstring)
```

to the script.

5.2.6.4 Frequency Switched On-The-Fly Mapping

In this example we perform frequency switched observations of the HI 21 cm line to map a 5 by 5 degree region of the sky. We use pixels that are 3 arc–minutes in size and have an integration time of 2 seconds per pixel. We do not observe the whole map in this example. This example is available as /home/astro-util/projects/6D01/example-four.py

```
# Frequency Switched Observations where we loop through a list of sources
#
# first we load the configuration file
execfile('/home/astro-util/projects/6D01/configurations.py')
#
# now we load the catalog file
Catalog('/home/astro-util/projects/6D01/sources.cat')
#
# now we configure the GBT IF system for freq switched HI observations
Configure(spectral_fs_config)
#
# now we balance the IF system
Balance()
#
# now we use a Break() so that we can check the IF system
Break('Check the Balance of the IF system')
#
# now we set the parameters for the map
#
# which source
srcs = 'Object2'
#
# the size of the map along the major axis
majorSize = Offset("Galactic",5.0,0.0)  # 5 degrees in galactic longitude
#
# the size of the map along the minor axis
minorSize = Offset("Galactic",0.0,5.0)  # 5 degrees in galactic latitude
#
# the size between two rows of the map
rowStep = Offset("Galactic",0.0,0.05) # 3 arcminutes expressed in degrees
#
# the time to scan each row
# time = majorSize / rowStep * integration time per pixel
scanTime = 5.0/0.05*2. # 2 seconds per pixel
#
# row start and stop number
# only do part of the map here
rowStart = 10
rowStop = 20
#
#
# now observe for the map
RALongMap(srcs,majorSize,minorSize,rowStep,scanTime,
          start=rowStart,stop=rowStop)
```

5.2.6.5 Position Switched Pointed Map

In this example we perform position switched observations to map a 0.5 by 0.5 degree region of the sky. We use pixels that are 3 arc–minutes in size and have an integration time of 120 seconds per pixel. We observe the reference Off position after every second point in the map. This example is available as /home/astro-util/projects/6D01/example-five.py

```
# Position Switched Observations where we repeatedly observe the same source
#
# first we load the configuration file and the catalog file
execfile('/home/astro-util/projects/6D01/configurations.py')
Catalog('/home/astro-util/projects/6D01/sources.cat')
#
# now we configure the GBT IF system for freq switched HI observations
Configure(spectral_ps_config)
#
# now we balance the IF system and use a Break to check the IF system
Balance()
Break('Check the Balance of the IF system')
#
# specify which source we wish to observe
srcs = 'Object1'
#
# now we set the parameters for the map
# the size of the map along the major axis
majorSize = Offset("J2000",0.5,0.0)  # 0.5 degrees in RA
# the size of the map along the minor axis
minorSize = Offset("J2000",0.0,0.5)  # 0.5 degrees in Dec
# the size between two points in a row of the map
pointStep = Offset("J2000",0.05,0.0) # 3 arcminutes expressed in degrees
# the size between two rows of the map
rowStep = Offset("J2000",0.0,0.05) # 3 arcminutes expressed in degrees
# specify how far away from the map center that the off position should be
# two degrees of arc in Right Ascension direction
myoff=Offset("J2000",2.0,0.0)
# how many points to observe between off observations
refInterval=2
# the time to scan each point in the map
scanTime = 120. # seconds
# which beam to use
mybeam='2'
#
# now observe for the map
PointMapWithReference(srcs,majorSize,minorSize,pointStep,rowStep,myoff,
                      refInterval,scanTime,mybeam)
```

5.3 What Makes a Good Observing Script

Rarely does an observing session exactly follow one's plans. A useful philosophy is to consider the work that would be involved in editing an Observing Script if something were to go wrong during its execution and you wanted to resume its execution where you left off. You should break apart any long scripts into smaller individual scripts to reduce the need for edits.

During your observing, you will make decisions as to how to proceed with the next observations. You should break apart large scripts to increase your flexibility in being able to react to the circurmstances that arise during your observing.

Choose the Optimal Size for your Observing Script

When preparing your science program, you should construct several Observing Scripts. An Observing Script should ideally contain from 5 to 30 minutes worth of observations. A good example is the following Observing Script, which sets up a source list, configures for the observation, slews to the source, balances the IF power levels, and then does the observations:

```
# this is the set-up for MYPROJECTID

#first load the catalog with the flux calibraters
Catalog("/home/astro-util/astridcats/fluxcal.cat")

#now load the catalog with the L-band pointing source list
Catalog("/home/astro-util/astridcats/lband_pointing.cat")

# which source to use
myCalibrator = "3C123"
calOffset = Offset("J2000", -0.58, 0.0, cosv = True)
galOffset = Offset("J2000", -1.45, 0.0, cosv = True)

# set things up for a spectral line frequency switched observation
# of the calibrator source
Comment("Configuring for spectral line with L-band")
execfile("/home/astro-util/projects/6D01/configurations.py")
Configure(spectral_fs_config)

# now slew to the source and balance the IF system
Slew(myCalibrator)
Balance()

#run a track scan for one minute
Track(myCalibrator, None, 60)
```

We recommend that the following should be avoided within a single Observing Script, as it will make the block too long:

Multiple configurations Multiple configurations (peak/focus and science observations should optimally be done in separate Observing Scripts).

Changing Receivers You should use only a single receiver within an Observing Script.

Multiple Maps You should perform only a single map within any Observing Script.

Chapter 6

Observing Strategies

6.1 Balancing Strategies

The GBT IF has many ways to add gain and/or attenuation in the IF path, depending upon the desired configuration. Before taking data with the GBT, the observer must ensure that all components along the IF path have optimum input power levels; this process is referred to as “balancing”. This will ensure for example that no components saturate and that amplifiers are in the most linear part of their dynamic range.

The system automatically adjusts power levels to optimum values when you issue the ”Balance” command in an Astrid script, i.e., *Balance*() The following discussion gives guidelines for when and how often to use the ”Balance” command.

Strategies for balancing the IF power levels depend upon the backend, the observing frequency, the observing strategy, the weather and the objects being observed. The DCR has a dynamic range of about 10 in its ability to handle changes in the brightness of the sky as seen by the GBT[1] The Spectrometer has the dynamic range to handle up to a factor of 4 change in the sky brightness.[2] The spectral processor can handle changes of about a factor 15 in the observed sky brightness. The sky brightness can change because of continuum emission of a source or a maser line as you move on and off the source. It can also change due to changes in the atmosphere’s contribution to the system temperature as the elevation of the observations change.

There are not any set–in–stone rules for when an observer should balance the GBT IF. However there are some guidelines which will allow you to determine when you should balance the IF. Here are the guidelines:

1 You should balance the IF after performing a configuration.

2 You should minimize the number of times you balance when observing.

3 If you know $\mathrm{T_{sys}} + \mathrm{T_{src}}$ will change by more than a factor of two (3 dB[3]) when you change sources (not between and on and off observation) you should consider balancing.

4 If the spectrometer reports errors in excess of 2 dB that cannot be explained by changes in antenna position (such as for on/off observations) then you should consider balancing.

5 Try to avoid balancing while making maps.

[1]From about 0.5 to 5 Volts of IF power in the IF Rack.
[2]A factor of 2 from its optimal balance point in each direction.
[3]A change in power from $\mathrm{P_1}$ to $\mathrm{P_2}$ can be represented in dB by $10\,[\log \mathrm{P_1} - \log \mathrm{P_2}]$

6 Never balance between "signal" and "reference" observations (such as during an on/off observation).

7 If you are observing target sources and calibration sources then try not to balance between observations of the targets and calibrators.

Whenever you balance the GBT IF you almost always change variable attenuator settings. Each attenuator setting has a unique bandpass shape. So if you change attenuators then you will likely see changes in the bandpasses and baselines of the raw data.

If during your observing you expect to see a change in power levels on the sky that are roughly equivalent to the GBT system temperature, then you should contact your GBT support person to discuss balancing strategies. There are no global solutions or formulae to follow and each specific case must be treated independently.

6.2 Active Surface (AS) Strategies

If you are observing at a frequency of 8 GHz or higher then you should use the AS. At frequencies below 8 GHz the AS does not provide any improvements to the efficiency of the GBT. Due to Radio Frequency Interference (RFI) considerations the AS may be turned off for lower frequency observations.

You do not need to do anything to turn on or off the use of the AS. The GBT telescope operator performs these tasks.

6.3 AutoOOF Strategy

AutoOOF is recommended for observing at frequencies of 26 GHz and higher. For a description of this procedure and strategies, refer to Section 5.2.2.2. For the associated data display, see Section 4.1.4.

6.4 Strategies For Pointing and Focusing

How often you need to point and focus the GBT depends on the frequency of your observations, the weather conditions, whether or not it is day or night-time, and the amount of flux error that your experiment can tolerate from pointing and focus errors. We will use "good" to refer to situations where the flux errors from pointing and focusing are less than 5% and "usable" for when these errors are between 5-10%. **Note that this is not the total flux error limit of the system - only the contribution from pointing errors.**

In Table 6.1 the pointing and focus accuracies required to achieve "usable" and "good" performance as a function of observing wavelength are shown. The approximate wind limits at which these flux accuracies can be achieved are shown along with the recommended observing strategy for performing peak and focus measurements. The pointing error due to wind can be approximated by

$$\sigma_{\rm w}(wind) = 0.23 \cdot {\rm c_w}^2 (m\ s^{-1})\ arcsec, \tag{6.1}$$

where c_w is the wind speed.

If you are looking at extended sources then the pointing requirements can be relaxed from those in Table 6.1. You can use Equation 6.1 to estimate what the wind requirements should be when observing extended sources.

The recommended observing strategies are:

Table 6.1: Requirements, limits and observing strategies for "usable" performance (10% rms flux errors from pointing alone) and "good" performance (5% rms flux errors from pointing alone).

Receiver	Frequency	wind limit		Performance	Observing
	GHz	(m/s)	(mph)	Level	Strategy
PF1	0.340	43.3	96.9	Good	A
PF1	0.415	39.6	88.6	Good	A
PF1	0.680	30.6	68.5	Good	A
PF1	0.770	28.9	64.6	Good	A
PF2	0.970	26.0	58.2	Good	A
L–band	1.4	21.4	47.9	Good	A
S–band	2.0	18.0	40.3	Usable	A
C–band	5.0	11.4	25.5	Usable	B
C–band	5.0	9.5	21.3	Good	B
X–band	10.0	8.5	19.0	Usable	B
X–band	10.0	7.0	15.7	Good	C
Ku–band	15.0	6.7	15.0	Usable	C
Ku–band	15.0	5.5	12.3	Good	C
K–band - lower	20.0	5.1	11.4	Usable	C
K–band - lower	20.0	4.0	8.9	Good	D
K–band - higher	25.0	4.8	10.7	Usable	C
K–band - higher	25.0	3.7	8.3	Good	D
K–band Focal Plane Array	25.0	4.8	10.7	Usable	C
K–band Focal Plane Array	25.0	3.7	8.3	Good	D
Ka–band	32.0	4.0	8.9	Usable	C
Ka–band	32.0	2.6	5.8	Good	D
Q–band	45.0	2.6	5.8	Usable	D
Q–band	45.0	2.2	4.9	Good	D
MUSTANG	90.0	4.5	10.	Usable	D

Strategy A (Appropriate for Prime Focus, L–band and S–band observing.) The antenna should deliver "good" pointing and focus performance under all allowed wind conditions and in the presence of any thermal gradients. We always recommend at least one peak (for all receivers) and one focus (except when using the Prime Focus Recievers) at the start of a new observing program, if only to ensure that the antenna has not been left misconfigured (e.g. well out of focus because the previous observer was performing out-of-focus beam maps!). After this initial check, the blind pointing/focus performance of the antenna should provide sufficient accuracy.

Strategy B (Appropriate for C–band or "usable" performance at X–band.) Ensure that the wind speeds do not exceed the limits listed in Table 6.1. Extreme thermal gradients (typically only encountered during the daytime with particularly unfavorable solar illuminations) may produce pointing and axial focus errors which unless corrected will approach the limits for successful observations. We recommend you perform peak and focus measurements every few hours during night-time, increasing to perhaps once per hour around local noon and into the afternoon.

Strategy C (Appropriate for "good" X–band, all Ku–band, and "useable" performance at K–band.) Ensure that the wind speeds do not exceed the limits listed in Table 6.1. Daytime thermal gradients may easily produce pointing and axial focus errors which unless corrected will approach or exceed the limits for successful observations; under some conditions these gradients may also extend well into the evening. We recommend that you perform peak and focus measurements at least once an hour initially. The spacing between peak/focus checks may be extended during the night-time if the results appear stable. Remember to increase the frequency of pointing and focus checks again after dawn.

Strategy D (Appropriate for "good" K–band, Ka–band or any Q–band or MUSTANG observing.) Ensure that the wind speeds do not exceed the limits listed in Table 6.1. Uncorrected thermal gradients will certainly cause unacceptably large pointing and focus corrections. Perform peak and focus checks at least every half hour initially. The spacing between peak/focus checks may be extended during the night-time if the results appear stable, but we recommend performing a check at least once every sixty minutes in any event.

6.5 Calibration Strategies

For best flux density calibration of spectra, it is recommended that you should observe continuum flux density calibration sources at least once during an observing session. To do this, do a Peak/Focus on the calibrator, followed by an observation in the same spectral line setup used for the program sources. This will give the relation of flux density to antenna temperature as a function of frequency that can be applied to the program spectra.

If you can observe a calibrator both at the beginning and end of a session, it will also indicate if there have been any changes in the GBT system which need to be taken into account during the observing run.

Of course, there may be other outstanding reasons to perform calibration observations more often. If you have concerns over how often you should observe a calibrator you should get into contact with your GBT support person.

6.6 Balancing The Converter Rack

When using the Spectral Processor, pulsar, or Radar backends it may be necessary to set attenuation levels in the Converter Rack. This currently does not happen automatically. This can be accomplished in one of two ways. The first is to specify the target power level (in volts) of the Converter Rack modules. This is accomplished using the BalConvRack() function. Here is an example of how you would use this function in an Observing Script:

```
# make the function available
execfile("/users/tminter/astrid/BalConvRack.py")
#
# to use:  BalConvRack( module_list, target_values)
#
#   module_list is an array with a list of converter module numbers.
#   target_values is a corresponding array with a list of RF values.
#
# For example if one wants to adjust CM1 to an RF value of 2.5,
#   and CM5 to 3.5,
#   then one says:
#
BalConvRack( [1,5], [2.5, 3.5] )
#
#
# This adjusts the converter module attenuators such that
# the target power levels are attained, to within a threshhold
# of 0.15 volts as read by the RF power samplers.
```

The second method is to specify the attenuation settings of the Converter Rack attenuators. All of the Converter Rack attenuators can be set to the same value using:

```
# make the function available
execfile("/users/tminter/astrid/SetConverterRackAttenuation.py")
#
# set all the attenuators to 15.5 dB
SetConverterRackAttenuation(15.5)
```

6.7 Observing Strategies For Strong Continuum Sources

Spectral line observations of strong continuum sources leads to a great amount of structure (i.e. ripples) in the observed spectra. So observations of strong continuum sources requires careful consideration of the observing setup and the techniques used.

If you are trying to observe broad spectral lines (wider than a few MHz) toward a source with strong continuum emission (more than 1/10th the system temperature), then you should consider using double position switching. This technique is discussed in an Arecibo memo by Tapasi Ghosh and Chris Salter which can be found at http://www.naic.edu/ astro/aotms/performance/2001-02.ps.

Another issue is finding a proper IF balance that allows both the "on" and "off" source positions to remain in the linear range of the backend being used. This means that one must find the IF balance in both the "on" and "off" position and then split the difference – assuming that the difference in power levels between the "on" and "off" do not exceed the dynamic range of the backend. The BalanceOnOff() (see § 5.2.2.1) can be used to accomplish this type of balancing.

6.8 High Frequency Observing Strategies

When observing at frequencies above 10 GHz you should be aware that additional calibration measurements may be necessary. The telescope efficiency can become elevation dependent, atmospheric opacities are important and the opacities can be time variable. You should contact your GBT support person to discuss these issues.

All the GBT high frequency receivers have at least two beams (pixels) on the sky. You should make use of both of these during your observations if possible. For example, if you are doing position switched observations and your source is not extended then you can use the Nod procedure to observe.

6.9 VLBI

The GBT is different from the standard VLBA stations. 1) Changing between Gregorian receivers in the receiver turret takes 1-2 minutes; and 2) Changing between Gregorian and prime focus requires about 10 minutes. Changing from one prime focus receiver to another requires about 4 hours, because one feed must be physically removed and replaced with another.

It is recommended to allow for pointing and focus touch-ups when observing at frequencies of 8 GHz and higher. Table 6.2 shows the recommended maximum intervals between pointings. At the higher frequencies (18-26 GHz and 40-50 GHz) also do a pointing check when the source elevation has changed by 15 degrees or more.

The observer should select as a pointing source a strong continuum source (flux density > 0.5 Jy) within about 15 degrees and at similar elevation as the program source. Include the pointing calibration

Table 6.2: Recommended maximum intervals between pointing observations during VLBI observerations.

Frequency Band	Interval between pointing scans
8–10 GHz	4–5 hours
12–16 GHz	3–4 hours
18–26 GHz	1.5–2 hours
40–50 GHz	30–45 minutes

source in the Very Long Baseline Interferometer (VLBI) observing schedule at the recommended intervals. Allow about 8 minutes for the pointing and focus check in the schedule. Note also that significant pointing errors at 7mm can happen when the wind speed is greater than 3 m/sec (7 miles per hour). For 1.3 cm significant pointing errors can occur for wind speeds greater than 6 m/sec (14 miles per hour). Refer to § 6.4 for details.

To include a pointing and focus scan in your schedule, put commands into your ".key" file similar to the following:

```
comment='GBT pointing scan.'  peak=1
stations =  gbt_vlba
source = 'J0920+4441'  dwell = 08:00  vlamode='VA'  norecord /
nopeak
```

It is important to specify only the GBT ("stations=gbt_vlba") when putting in "PEAK=1". Otherwise it may do a reference pointing for the whole VLBA, and if the selected pointing source is under about 5 Jy for the VLBA, it could produce bad results.

VLBI observers should see http://www.gb.nrao.edu/ fghigo/gbtdoc/vlbinfo.html for more details.

Chapter 7

GBT IF System

In this chapter we provide a general outline of the GBT IF. Figures 7.1 and 7.2 give a simplified overview of the GBT IF path and will guide our discussion. We will not cover the MUSTANG, KFPA or Ka–band receiver IF paths. Note that during each frequency mix, each polarization pair is mixed with a signal from the same synthesizer. All synthesizers are locked to our H-maser frequency standard.

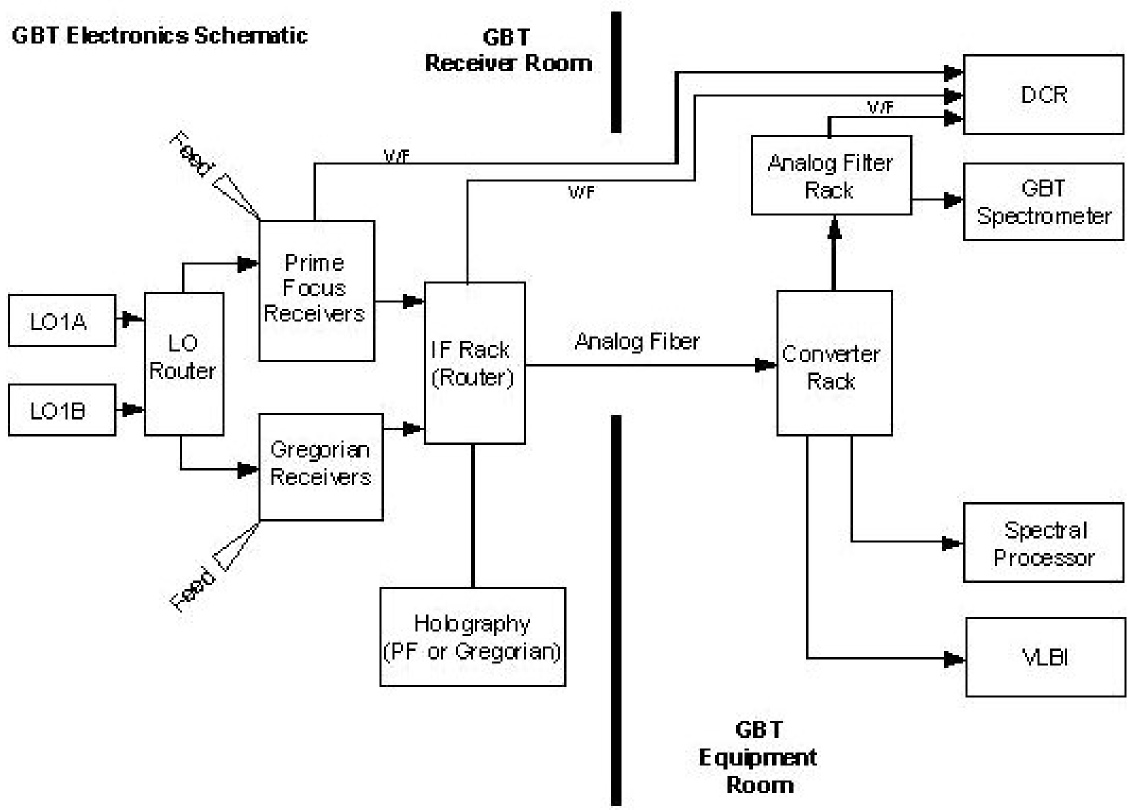


Figure 7.1: A simplified flow diagram of the GBT IF routing.

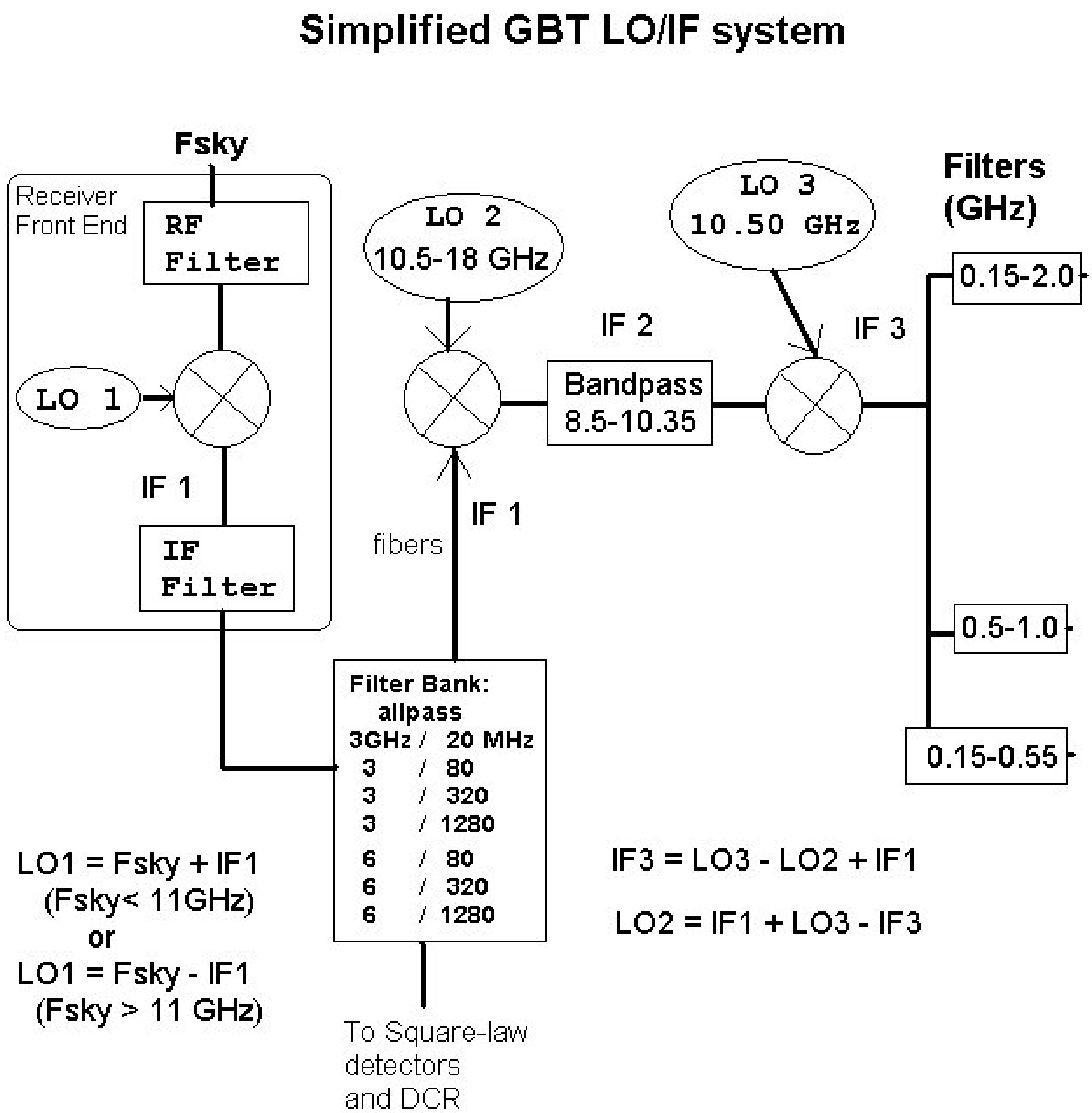


Figure 7.2: A simplified flow chart of the GBT IF.

7.1 From the Receiver to the IF Rack

The frequency that is observed is given by F_{sky}. Within the receiver the detected signal at F_{sky} is mixed with the LO1 signal. The LO1 frequency is derived from a synthesizer and can vary in time when Doppler tracking the velocity of a spectral line. The result of the mixing of F_{sky} and LO1 is the IF frequency, IF 1. The allowed IF 1 center frequencies are 1080, 3000 and 6000 MHz. Filters limit the bandwidth in the receivers both before and after the LO1 mix. There are also filters in the IF Rack that limit the bandwidth. The resulting allowed bandwidths are 20, 80, 320, 1280 MHz and "All Pass" (i.e. no filtering other than the response of the receiver).

In the IF Rack each signal is split into two (single beam receivers only) copies of the original signal. Each signal in the IF Rack is detected and then sent to the DCR (as used during pointing and focus observations). Each signal is also sent as an analog signal over optical fiber to the Jansky Lab to the Converter Rack.

7.2 From the Converter Rack to the Backend

When the signal reaches the Converter Rack it is split into four separate copies. This allows up to eight different copies of the received signal for single beam receivers and four copies of each received signal for dual beam receivers.

In the Converter Rack the signal is mixed with the Second LO (LO2) signal. Each copy of the signal can be mixed with a different LO2 since there are eight different LO2 synthesizers. The resultant signals are then sent through a filter to make sure it has a bandpass of no more than 1.85 GHz. A final mix with a fixed frequency of 10.5 GHz then gets the signal within the input bandpasses of the backends. There is a final set of filters that ensures the signal has the correct bandwidth for the backend.

Chapter 8

Radio Frequency Interference

Radio Frequency Interference (RFI) can be a significant problem for some observations. The most up to date information on the RFI environment at the GBT can be found at
http://www.gb.nrao.edu/IPG/.

Useful resources, referenced from the above web page include a list of known sources of RFI:

https://safe.nrao.edu/wiki/bin/view/GB/Projects/RFIReportsTable

and plots of RFI monitoring data:

http://www.gb.nrao.edu/IPG/rfiarchivepage.html

Every observer should check for known RFI around their observing frequencies. If you suspect that this could have a significant impact on your observations you should contact your scientific support person to decide on an appropriate course of action.

Mitigation of known RFI signals.
In some cases, it is possible to turn off a known RFI source. For example, there is an amateur transponder at about 432 MHz, which we can request to be shut down. If there are known RFI signals, the user should discuss them with the scientific support person. Given enough advance warning (days to weeks), we may be able to have them shut down during the observing.

Chapter 9

Introduction to the Dynamic Scheduling System

by Jim Braatz and Dana Balser, July 15, 2010

This document gives an introduction to the Dynamic Scheduling System (DSS) for the Robert C. Byrd Green Bank Telescope (GBT). The GBT has been scheduled with the DSS since October 1, 2009. Observers can access the DSS through this site:

https://dss.gb.nrao.edu

Additional information on the DSS can be found here:

http://science.nrao.edu/gbt/scheduling/dynamic.shtml

9.1 Overview of the DSS

The primary goal of the DSS is to improve the efficiency of GBT observations by matching the observing schedule to predicted weather conditions, while allowing each observer to retain interactive control of the telescope. Each day the DSS will examine the weather forecast, equipment availability, observer availability, and other factors, and set an observing schedule for the 24-hour period beginning the next day. Observers will therefore get about 24-48 hours notice before their project will observe. Observers will have the opportunity to pause their observing program, set blackout dates indicating when they are unavailable for observing, and back out of current observations if they find the observing conditions are not suitable to their science goals.

The DSS readily accommodates remote observing, but by being on site in Green Bank observers increase their likelihood of being scheduled during the period of their visit. Visits to Green Bank should be arranged in advance with the project's "Friend," and observers should expect to spend five or more days in Green Bank to give enough opportunity for their project to get scheduled at least once. Projects observing at high frequencies (20 GHz and higher) typically require staying in Green Bank for a week or longer.

9.2 Resources for Getting Help

The main resource for help with the DSS is the email helpdesk:
helpdesk-dss@gb.nrao.edu
There is a link to the helpdesk on the main DSS web site. Also on the web site is a link to online documentation, including a FAQ and a glossary of DSS terms.

9.3 DSS Terminology

The process of scheduling GBT observations begins with the preparation of the proposal using the NRAO Proposal Submission Tool (PST). Proposals accepted by the selection committee become GBT projects that appear in the DSS system and are identified by an assigned project ID (e.g., GBT09C-001).

Projects are divided into sessions, which have associated parameters that define how the observation should be scheduled. These parameters include sky position, time allocated, observing frequency, and minimum and maximum durations preferred for a single, contiguous block. Sessions for monitoring observations have additional parameters describing how often to repeat the observation. The project investigators initially define the session parameters in the proposal, but the parameters may be modified during the refereeing process. Observers can see the most critical session parameters on the DSS web pages.

Completing the observations for a session may require scheduling multiple segments. Each contiguous block of scheduled time is called a *telescope period*. (Note a telescope period is known as a "session" in Astrid with a unique session code.)

As telescope periods are completed, the project will be "billed" the time. If any time is lost to weather or an equipment failure, the observer may consult with the telescope scheduler (via email to the DSS helpdesk) and request that the project not be billed for the lost time.

9.4 Controlling the Scheduling of a Project

Users can access their DSS account by logging in to the system at https://dss.gb.nrao.edu. The DSS username and password are the same as those used for NRAO Interactive Services (i.e., the Proposal Submission Tool).

From the DSS web site, users can view and manage the scheduling information for their projects. Users can control when their project is scheduled by enabling or disabling sessions, individually. Sessions are enabled for observing simply by clicking a check box. Once enabled, an observing session enters the pool of sessions eligible for scheduling. Note that astronomers intending to observe remotely must be trained and approved by GB staff before the project can be authorized and made eligible for scheduling.

Observers can enter personal blackout dates. Blackouts can be entered either as onetime events (e.g., May 1, 20:00 to May 4, 05:00 UT) or as repeating events (e.g., every Monday from 15:30 to 17:30 ET). If all observers for a given project are blacked out at a given time, that project will not get scheduled. If at least one observer is not blacked out, the project is eligible for scheduling. The default time zone used for entering blackouts is set on the *Preferences* tab, which is linked at the top of every DSS web page. Observers can also override the default by selecting a time zone when making a blackout entry. Observers with more than one project will find that they need to enter blackout dates only once, and the dates will be applied to all their projects. Those visiting Green Bank to observe should use blackout dates to mark the periods of their travel before and after the run to ensure they are scheduled only when available and ready on-site.

Guidelines for the use of blackouts: While blackout dates give observers control of the scheduling process, efficient GBT operation requires that not too much time be blacked out or disabled. It is especially important that projects with large observing allocations not have too much time unavailable for scheduling because of blackouts. As a guideline, projects with more than 20 hours of allocated observing should limit time that cannot be scheduled to no more than 20% of the total eligible observing time over the course of a trimester. If a project cannot meet this guideline, the PI is encouraged to increase observing opportunities by enlisting additional observers who are qualified for remote observing. Projects that require observers to visit Green Bank for training are excluded from this guideline until the observers are trained for remote observing.

Caution Regarding Blackouts: If a project has only one observer, that observer should be particularly conscientious of blackouts. It can be easy for an observer to inadvertently hamper observing opportunities too much by setting blackout dates too freely, particularly repeating blackouts. So repeating blackouts should be used with care. Targets with low declinations, such as the Galactic Center, have tightly constrained observing opportunities to begin with, so observers on such tightly constrained projects should be particularly careful with blackouts that would further limit their observing opportunities. Consider, as an example, a project that has a session with a 4-hour minimum duration to observe the Galactic Center. If the observer has a repeating 1-hour blackout date that intersects the window, the entire session becomes ineligible each time the blackout intersects the 4-hour window.

When entering blackouts, keep in mind, too, that projects do expire, so it is in the interest of the observer to keep the projects eligible for scheduling as much as possible.

9.5 Target Positions

The DSS keeps track of a project's scheduling requirements via the session parameters, which can be viewed on the project page. The PI should check that session parameters properly reflect the needs of the project. The project Friend assigned by NRAO can also offer advice on optimizing session parameters, where appropriate. In some cases, a session's target position may be representative of a group of objects clustered on the sky. As the project progresses and some of these targets are observed, this representative position may need to be updated. In this case, the PI should send an email request to the DSS helpdesk.

The DSS can automatically update the sky coordinates of common, fast-moving solar system objects, including comets. The position is updated each day prior to scheduling. On the project page under "Project Sessions", an asterisk next to the coordinates indicates that the position for that session is automatically updated in this manner.

Many observers find it helpful to use a sky-plotting tool to help plan their observations and keep track of target locations on the sky. The cleo "scheduler", which runs on Linux systems in Green Bank and can be run remotely through vnc, is one such tool that allows a GBT user to plot target locations on the sky for any date and time. This application can read target coordinates from a standard astrid catalog file. Observers will find this tool handy for identifying the time of day a project may get scheduled, as well as helping to plan observations in detail after they are scheduled. To run the program, type:

```
unix% cleo scheduler
```

9.6 Contact Information and Project Notes

Observers can specify how they should be contacted, prior to and during their observations. The GBT operations staff stress that it is critical to keep contact information current. Each observer can provide "dynamic contact" information in a free-format text box. Here the observer should provide home and cell phone numbers and any other relevant contact information. Observers can specify the order in which

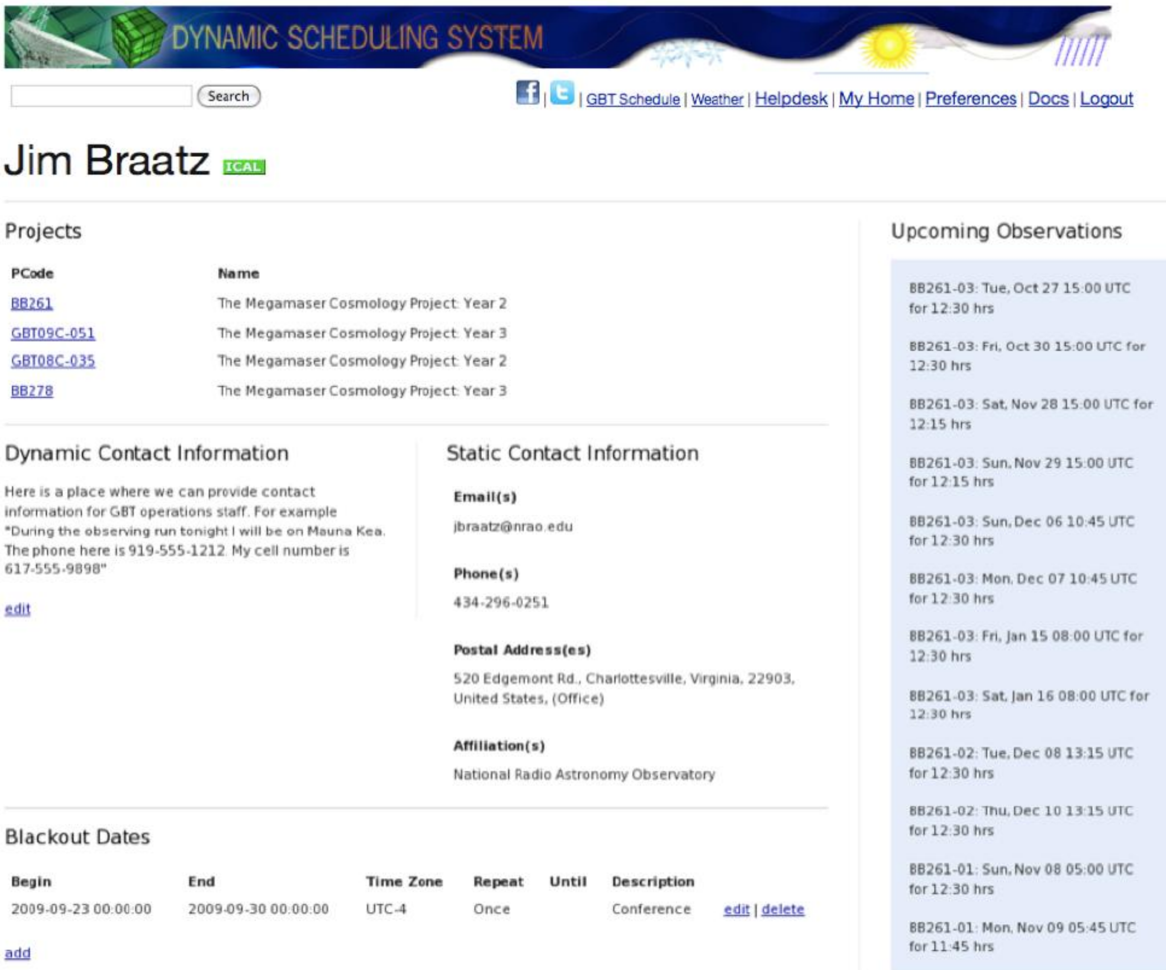


Figure 9.1: A sample DSS home page.

they should be contacted by GBT operations, in the event of any schedule changes or in case there is need to contact the observer for any reason prior to the scheduled start time. Specify the order by clicking the arrow icons next to the list of team members, on the DSS project page.

Finally, observers can record "Project Notes" on the DSS project web page. Project notes provide observers a place to store and share observing instructions. The notes are visible to all project team members as well as the GBT operations staff and GBT schedulers. Observers who need to share instructions or other information with the GBT operator prior to the start of an observation can provide these instructions in the project notes area. Project notes are not intended to be a log for observations, but rather a place to store brief instructions or news that should be shared among observers and the GBT operator.

9.7 The DSS Software

Upon logging in to the DSS system, users arrive at their DSS home page (Figure 9.1) where they see a list of active projects on which they appear as co-investigator. From the DSS home page, users can:

- Access the project page for each of their affiliated projects

- See a list of upcoming observations
- See a list of upcoming Green Bank room reservations
- See their "static" contact information, as entered in the NRAO services system (http://my.nrao.edu)
- Set "dynamic" contact information
- Set blackout dates
- use a link to the current GBT fixed schedule
- use a link to the weather forecasts page
- Send email to the helpdesk
- Set the default time zone via the Preferences link
- Access DSS documentation
- Establish an iCalendar subscription. Instructions for using iCalendar are available by hovering the mouse cursor over the iCal icon on the DSS Home Page.

By selecting a project ID, observers are presented with the project page, where they can:

- See a project calendar
- Inspect session parameters
- Enable or disable individual sessions
- Specify observers from the project team, and set the order they should be contacted by GBT operations
- See a list of blackout dates for all observers on the project
- See a list of completed telescope periods
- Store and share project notes

The project calendar gives observers an idea when their project is eligible for scheduling. Regardless of the weather, there will be times when a project is not eligible for scheduling, for example because of no receiver availability, observer blackouts, fixed telescope maintenance periods, and other fixed projects appearing on the GBT schedule. Times not eligible for scheduling will be grayed out on the project calendar.

The project calendar helps with planning in a number of ways. For example, by identifying days when they are not likely to be scheduled, observers can "tune out" of the DSS system for a while. However, it is important to understand that a session's eligibility is based on ever-changing constraints, and can change from "not eligible" to "eligible" at any time. Therefore, if observers wish to take a break from observing based on the calendar outlook, they should either disable all sessions until they are ready to resume with the observing, or enter blackout dates to cover the period they do not wish to observe.

The project page includes a panel with project team members listed. Using a checkbox, team members can select or deselect those identified as observers. They can also rearrange the order observers are listed. The top observer in the list is expected to observe the next scheduled session. If there is a change in schedule, this person will be called first.

9.8 Responsibilities

Each project has a Principal Investigator (PI) and, optionally, a list of additional investigators. An investigator is eligible to be an observer for a given project if that person is qualified for remote observing or is on site in Green Bank.

It is essential that one of the observers for a scheduled project contact GBT operations at least 30 minutes prior to the start of the observation. Observers can contact the GBT operator by telephone (304-456-2341), by the CLEO chat program (for qualified remote observers), or by showing up in the GBT control room. If the GBT operator has not been contacted within 30 minutes of a session's start time, the operator will phone observers in the order they are listed on their project web page.

The PI is responsible for:

- Managing the project
- Identifying all associated observers
- Working with project team members and the GBT project Friend to ensure that observing scripts are properly and promptly prepared.
- Enabling each session by clicking the "enable" button on the project's web page. Sessions should be enabled only if they will be ready for observing in the next 24 hours.
- Ensuring that all associated observers have provided contact information, including a current telephone number and an email address for each observer.
- Ensuring that a project's scheduling information is current. This includes checking the hours remaining on the project and ensuring that the session parameters are up-to-date and accurate.
- Ensuring that each scheduled telescope period has an observer who is available at least 30 minutes before the session is scheduled to begin.

Observers are responsible for:

- Ensuring that the DSS project web page has their current contact information. For remote observers, this includes entering telephone numbers where they can be reached at the time of observation.
- Contacting GBT operations 30 minutes prior to the start time of an observation.
- Attending to observations during a scheduled telescope period.
- Notifying GBT operations if they find conditions unsuitable for their session.

9.9 Remote Observing

To use the GBT remotely, observers must first be trained and certified by Green Bank staff. In general, astronomers must observe at least once in Green Bank before being certified for remote observing. Please note that students should be trained on site by GBT staff, not off site by others. Training and certification received prior to the DSS test period are still valid. Experienced observers, when using instruments or observing modes unfamiliar to them, should plan to visit Green Bank if they require assistance. See Chapter 11 for more about remote observing. Additional information is available at:

http://www.gb.nrao.edu/gbt/remoteobserving.shtml

Contact your project "friend" or the DSS helpdesk (helpdesk-dss@gb.nrao.edu) if you believe the DSS does not have you listed properly as a qualified remote observer.

9.10 The Daily Schedule

Each day between about 5:00 and 10:00 AM ET the telescope schedule is fixed for the 24-hour period beginning 8:00 AM ET the next day. For example, by 10:00 AM Monday, the observing schedule is fixed for the period 8:00 AM Tuesday through 8:00 AM Wednesday. Each morning this daily schedule is published and can be viewed on the DSS web site by anyone. Those with projects on the 24-hour fixed schedule will be notified by email.

Observers must ensure that their blackout dates and ”session enabled” flags are up to date each day by about 5:00 AM ET. Changes made after this time may not be reflected in the upcoming day’s schedule.

It is possible that weather conditions may change after a schedule is published, compromising the observing efficiency for some scheduled telescope periods. The observer or GBT staff may then decide to cancel a telescope period and substitute an alternate ”backup” observation in its place. Note that the observer may decide that the weather conditions are too poor even after beginning the observation. Equipment failure can also lead to cancellations. If GBT staff must change the 24-hour schedule for these reasons, affected observers will be notified immediately by email or telephone.

9.11 Backup Projects

When a scheduled telescope period is cancelled, a backup project will fill the time. Backup projects can come in two categories: observer-run and operator-run.

Observer-run backup projects are those for which observers have volunteered to be called on short notice. The notice could be as little as 15 minutes, although the GBT staff will attempt to make the lead time as long as possible. Backup project observers should be ready to take control of the telescope at any time of the day or night, consistent with their observing program and blackout dates. These call-outs are expected to be rare. By volunteering as a backup project, observers improve their project?s chances of getting observing time. Note that identifying a project as a backup does not penalize that project during the normal scheduling procedure. The project will compete for regular scheduling on an equal footing with all other projects, but the PI is agreeing to make the project available as a backup in addition to regular scheduling. Note that observer-run backup projects will not be called on to observe during times they have blacked out on their DSS calendar.

Operator-run projects contain observing scripts that may be run by the GBT operator, without need for direction from project team members. The observational strategy must be simple. Operator-run projects are characterized by:

- Minimal calibration requirements, e.g. a single pointing/focus calibration at the beginning of the run. If the observation requires more calibration than a single pointing/focus or simple repetition of a pointing/focus script at regular intervals then it will not qualify as an operator-run candidate.
- Minimal changes in observing mode.
- Use of only one receiver.
- No scientist intervention required. An operator can be expected to determine if a point/focus measurement is reliable but cannot be asked to judge the quality of astronomical data. The operator also cannot be asked to judge which source would be best to observe at any given time. If there is any doubt whether an observation will produce reliable ”blind” results then this project is not suitable as an operator-run candidate.
- Clearly written instructions for the telescope operator describing the observing procedure, including which scripts to run. These instructions can be stored in the ”Project Notes” on the DSS web page.

These requirements bias operator-run projects to low frequency observations, but high frequency projects can be considered as well. There is no intention to implement ”service observing” by GBT scientific staff. Green Bank scientific staff will not be on hand to check operator-run projects.

Getting a project approved as an operator-run backup requires consent from the GBT Friend and the GBT DSS staff. To identify your project as a backup project of either sort, inform your GBT Friend.

9.12 Session Types

There are three types of sessions defined for astronomy projects: open, windowed, and fixed. Open sessions have no major constraints on when they can be scheduled, beyond the functional requirements that an observer is available, the source is above the horizon, and the weather is suitable. Most sessions fall into this category and provide the most flexibility in the DSS. At the other extreme are fixed sessions that have no flexibility and are prescheduled at a particular date/time; that is, their telescope periods have already been defined.

The third type is windowed sessions, which have some constraints but are not fixed on the schedule. The most common examples are monitoring sessions, where the science demands that an object must be observed at defined intervals. Windowed sessions are defined by a cadence that may be either periodic or irregular. For example, an observer may require observing a target once per month for five months, with each observation having a tolerance of plus or minus 3 days. In this example, the window size is 7 days.

Currently, windowed sessions are scheduled in the following way. The cadence information from the proposal is used to preschedule all windowed sessions whereby all of the telescope periods are temporarily fixed in what are called default periods. The user is given the window template (e.g, 8-14 January; 8-14 February; 8-14 March; 8-14 April; and 8-14 May). Within a windowed period, a windowed session will be considered like an open session. Near the end of each window range is a default period. If the session has not been selected by the time the default period arrives, the session will be scheduled in the default period. The default period may be moved manually to a later time slot within the window if the human scheduler notices a problem with the original default period. When the windowed period is scheduled, the observer will be informed 24-48 hours in advance, just like an open session. The only difference is that the observer will be provided with the window template for planning purposes. In the future, historical weather data (climate) will be used to schedule such sessions more efficiently within the window.

9.13 Projects that can Tolerate Degraded Weather

The DSS is designed to schedule projects in weather that is appropriate for the frequency being observed. Some projects can tolerate lesser weather conditions than the DSS would assign by default. For example, consider a project at K-band that observes many targets, each for a short duration, say 10 seconds. The observing time for this project is dominated by overheads in slewing from one position to the next, so marginal K-band weather might be acceptable. The observing team may prefer not to wait for very good K-band weather, which is rare and would delay their scheduling.

To enable more aggressive scheduling, the observer should send an email to the DSS helpdesk requesting that the project be considered for scheduling in lesser weather conditions. The DSS support team can enter a session-specific factor (”xi”) that effectively elevates the score for this session in marginal opacity conditions. The xi parameter is tunable so the observer can request that the project be scheduled very aggressively, or modestly so. The factor only affects scoring related to atmospheric opacity, so high frequency projects that are sensitive to high winds will still not get scheduled when the forecasted winds preclude accurate pointing.

The DSS support team will help observers decide if their project can tolerate lesser weather. Note that this capability will not be used to accelerate scheduling of projects that truly do benefit from the most appropriate weather.

Chapter 10

How weather can affect your observing.

The weather affects observations in three ways: winds affect the telescope pointing, differential heating/-cooling affect the telescope pointing and efficiency, and atmospheric opacity affect the received signal and the system temperature.

10.1 Time of Day

Differential heating and cooling of the telescope alters the surface of the telescope, resulting in degradation of telescope efficiencies, and 'bends' the telescope, resulting in pointing changes. At high frequencies, these effects are important. The current recommendations are that, for best work, observing above 40 GHz should only be done at night, from 3 hours after sunset to 2 hours after sunrise. At 40 GHz and above it is recommended to use "AutoOOF" (see 5.2.2.2 at the start of an observing session. Use "AutoOOF" for daytime observing at 27 GHz or higher

Low frequency observers may want to consider night time observing for two reasons. RFI is usually lower at night; and, in some cases, the sun has a slight negative impact on baseline shapes. By default, we assume that daytime observing will be acceptable for all observations below about 16 GHz.

Figure 10.1 depicts the range of UT, EST, and LST for our definition of 'night-time' observing.

10.2 Winds

Winds tend to buffet the telescope and, to a lesser extent, set the feed arm into motion. The current recommendations for wind limits can be found in § 6.4 (specifically in Table 6.1). The fraction of time when wind speeds are low is illustrated in Figure 10.2 which shows the cumulative percentages when wind speeds are below a certain value. (Figure 10.2 is from Ries, PTCS project Note 68.1) The DSS (Dynamic Scheduling System, see Chapter 9) uses forecasted wind speeds when it determines what projects are suitable for scheduling, one should rarely see any negative impacts from winds.

10.3 Atmospheric Opacities

The frequency range covered by the GBT extends from low frequencies where the opacity is relatively low (0.008 nepers) to high frequencies where opacity is very high (> 1 nepers). Atmospheric opacity

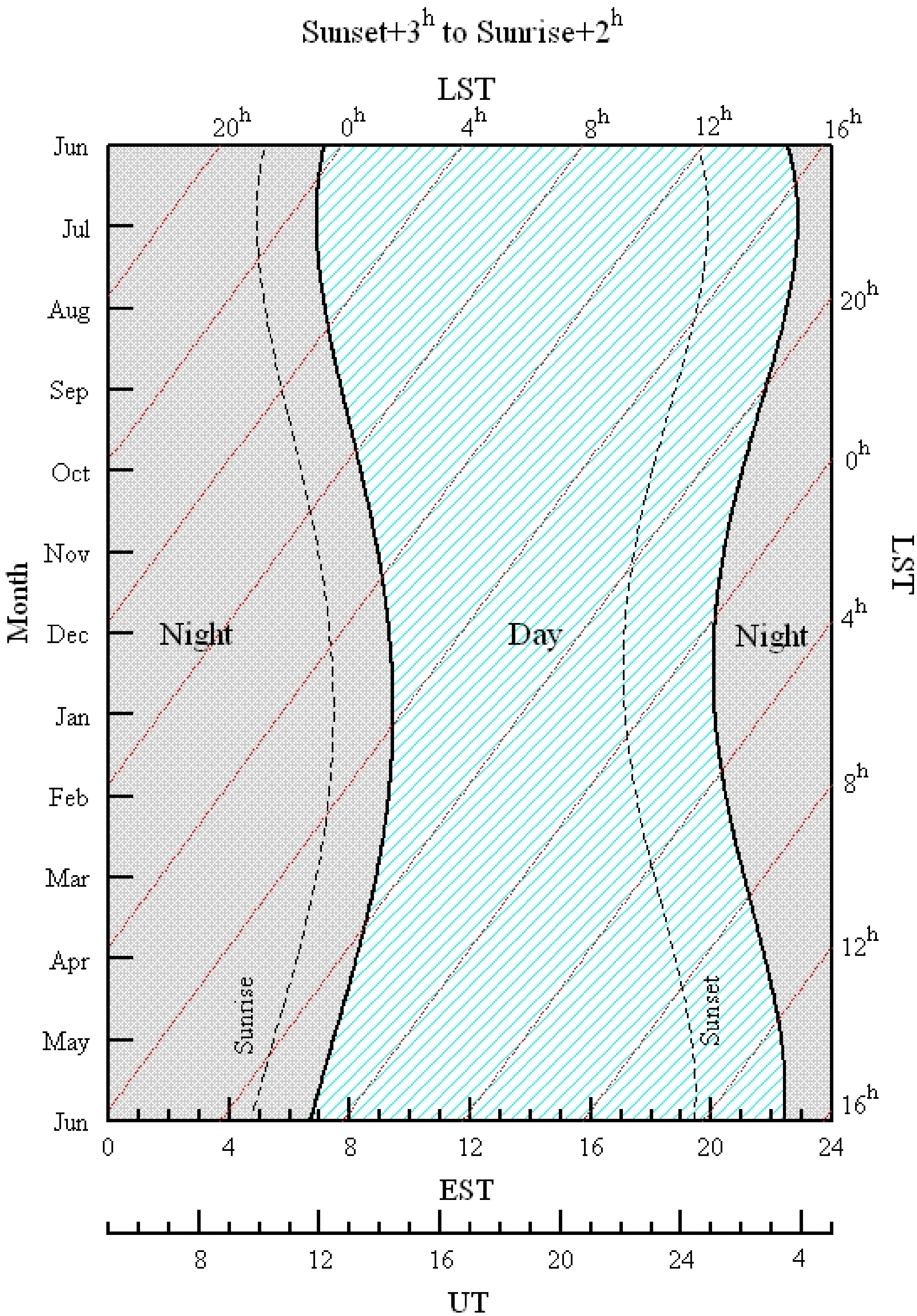


Figure 10.1: The range of UT, EST, and LST used in the GBT definition for 'night-time' observing.

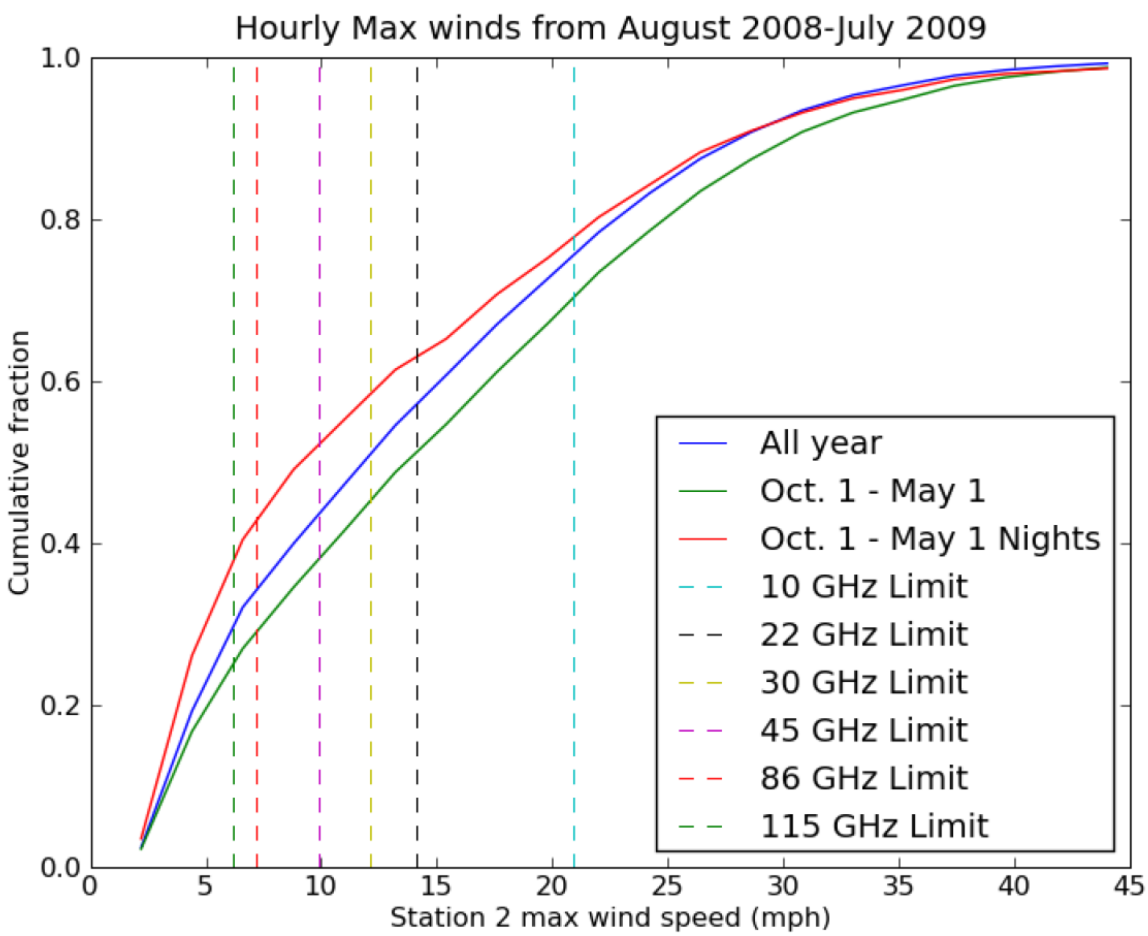


Figure 10.2: The cumulative fraction when wind speeds are below a certain value. Data from the year August 2008 to July 2009 are shown in blue; green shows winter data, and red shows winter nights.

hits observing twice – it attenuates the astronomical signal and it increases the system temperature, and thus the noise in the observation, due to atmospheric emission.

Figure 10.3 shows opacities, atmospheric contributions to the system temperature and number of air masses[1] the astronomical signal must pass through vs. elevation under three typical weather conditions as calculated using the method described on the GBT "High Frequency Weather Forecasts" web page (http://www.gb.nrao.edu/∼rmaddale/Weather/index.html). Typical total system temperatures are shown in Figure 10.4.

The opacities shown in Figure 10.3 are for planning purposes only and observers should not use them at high frequencies for calibrating data. Instead, one should use the actual opacities and the air mass from the bottom of Figure 10.3 to approximate the amount of attenuation a signal will experience at the expected elevation of the observation. The signal is attenuated by:

$$\exp^{\tau \mathrm{A}} \tag{10.1}$$

where τ is the opacity and A is the total number of air masses.

Since opacity is very weather dependent, please consult with a local support staff on how best to determine opacities for your observing run.

During the cold months, high frequency observers can expect to be observing with opacities that are at or below the average (50 percentile) winter conditions for Green Bank. Thus, high frequency observers can anticipate that the typical weather conditions under which they will observe will be best represented by the top 25 percentile conditions. In contrast, low-frequency, winter observers should expect they will observe under conditions that are worse than the 50 percentile and more like those of the 75 percentile conditions.

During the warm season (June through September), high-frequency observing is much less productive and we almost exclusively schedule low frequency observing. During these months, low frequency observers can plan on observing under the average, 50 percentile conditions.

10.4 GBT Weather Restrictions

During weather conditions that pose a risk for the safety of the GBT, the GBT operators will cease all observations and take the appropriate action to ensure the safety of the GBT. The operator is fully responsible for the safety of the GBT and their judgement is final. The operators decisions should not be questioned by the observer.

10.4.1 Winds

The following guidelines exist for periods of high winds. If the average wind speed exceeds 35 MPH over a one minute period, the operator will stop antenna motion. If wind gusts exceed 40 mph, or if winds are expected to exceed 40 mph for a period of time, the operator will move the antenna into the survival position. Only after the wind speeds have been below these criteria for 15 minutes will observations be allowed to resume.

Safety measures for high winds will take precedence over those for snow and ice.

[1] The airmass curve in Figure 10.3 is a better approximation than the csc(elevation) approximation which is only correct above about 20 degrees elevation.

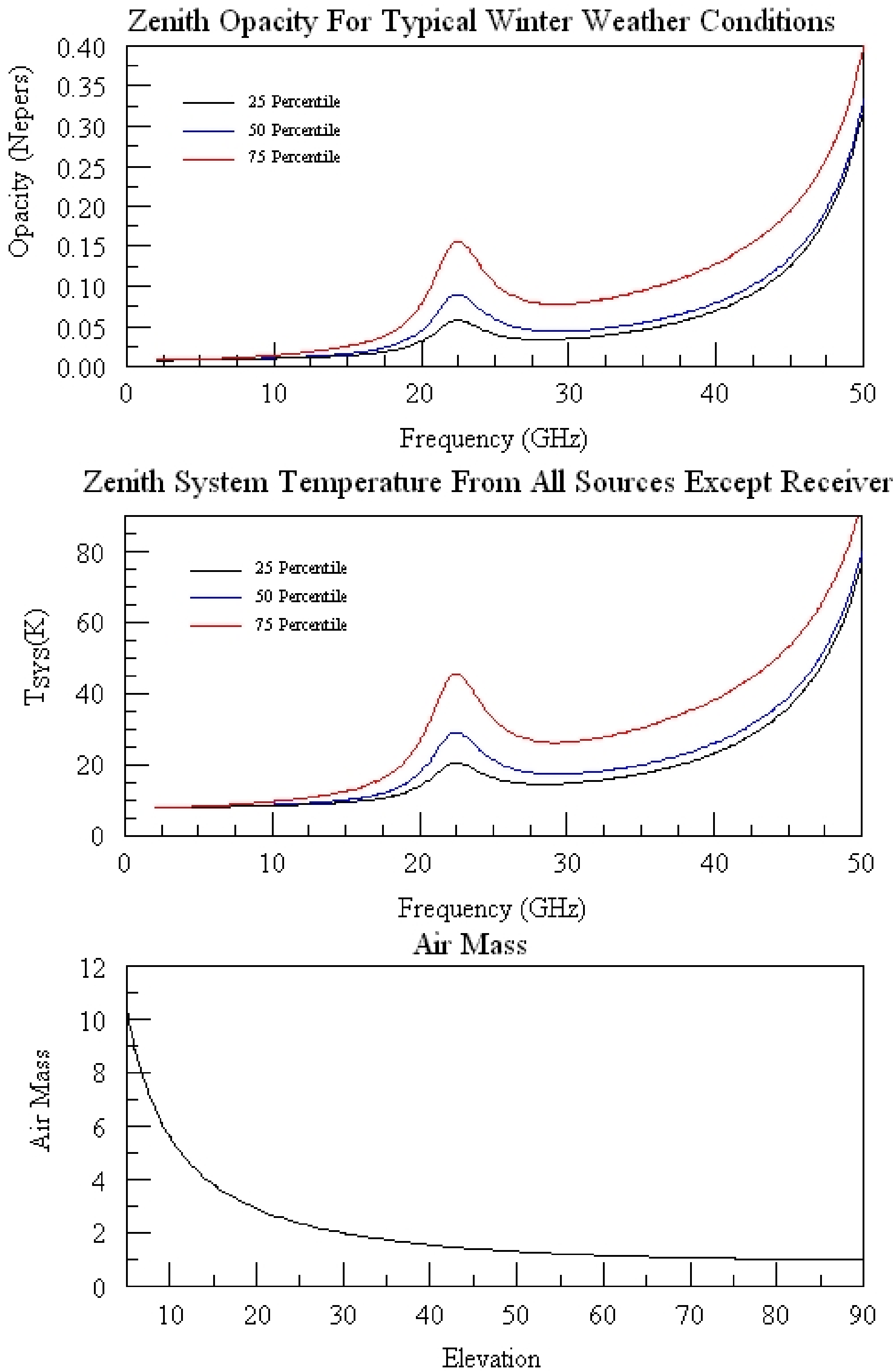


Figure 10.3: The top panel shows opacities under three typical weather conditions. The black, blue, and red curves represent the opacity under the best 25, 50, and 75 percentile weather conditions. (The 'average' opacity over the winter months is best described by the 50 percentile graph.) The middle panel is an estimate of the contribution to the system temperature at the zenith from the atmosphere, spillover, and cosmic microwave background. The bottom panel shows the number of air masses the astronomical signal must pass through as a function of elevation.

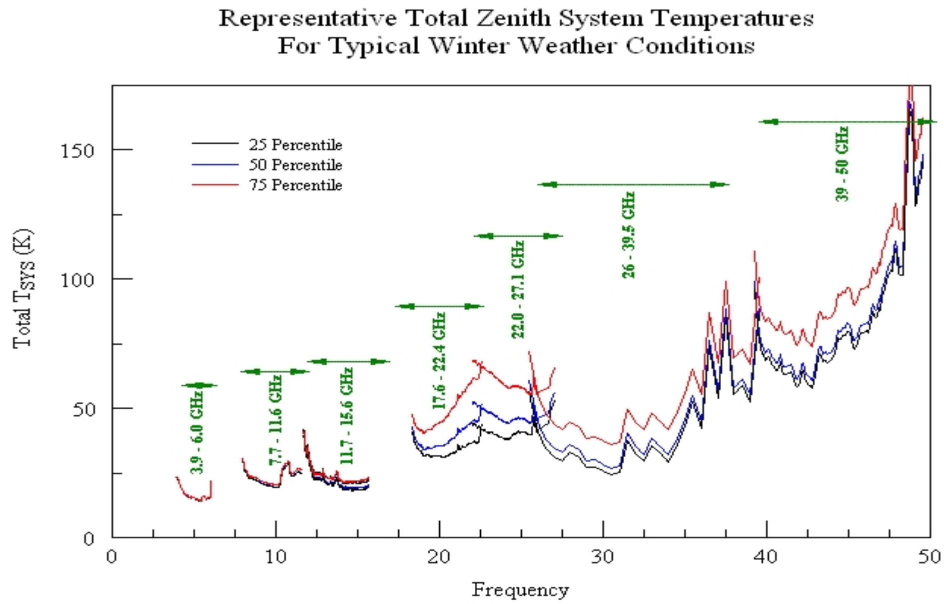


Figure 10.4: The zenith system temperatures for typical weather conditions.

10.4.2 Snow

If snow is sticking to any of the GBT structure, the operator will move the GBT to the "snow-dump" position. The decision to halt and resume observations is solely the responsibility of the GBT operator.

If dry snow appears to be accumulating, the operator may periodically interrupt operations to dump snow, and then resume observations.

10.4.3 Ice

If ice is accumulating on any part of the GBT structure, the operator will move the GBT to the survival position. The decision to halt and resume observations is solely the responsibility of the GBT operator.

10.4.4 Temperature

When the air temperature drops to 16° Fahrenheit, the Azimuth slew rate of the GBT will be reduced to half of its normal rate. (This is due to the changing properties of the oil used in the Azimuth drive bearings.) Half rate speed ($18°/min$ instead of $36°/min$) will be utilized until the temperature returns above 17° Fahrenheit. When the temperature drops below -10° Fahrenheit observations will be ceased until the temperature is above 0° Fahrenheit and the operator has determined that the Azimuth drive motors are ready for use.

10.4.5 Feed Blowers

The feed blowers blow warm air over the radomes of the feeds to prevent condensation and frost. Although beneficial for most receivers, they produce vibrations that contaminate the MUSTANG data. Thus, users of MUSTANG can request that the operator turn off the feed blower at the start of their observing session. One hour before the end of a MUSTANG observing session, the operator will decide whether or not the blower needs to be turned back on in order to ensure the feeds for all receivers are in good shape for the next observer. The operators use the criteria that the blowers will be turned back on for the last hour if either: 1) the dew point is within 5° Fahrenheit of the air temperature, or (2) the air temperature went from above to below freezing anytime during the MUSTANG run.

Chapter 11

Remote Observing With The GBT

11.1 Remote Observing Guidelines for Approved Projects

Permission to observe remotely must be explicitly granted by the Head of Science Operations (at the moment that's Toney Minter) at least two weeks prior to the observing run. Permission will be granted based on the appropriateness of the project and the demonstrated experience of the observer. Guidelines for approved remote observing projects are as follows.

Also observers should consult:
https://safe.nrao.edu/wiki/bin/view/GB/Observing/GbtObservingPolicies
and http://www.gb.nrao.edu/gbt/remoteobserving.shtml for information on observing policies and remote observing.

- Consult with the staff support astronomer at least two weeks prior to observing time.
- Provide the staff support astronomer with your telephone contact numbers (work, home, and cell) and agree in advance your location during the observations.
- Prepare ASTRID observing scripts in advance.
- Contact the telescope operator 30 minutes before the start of your setup time. The number to call is 304–456–2346 or the Operator's direct line at 304–456–2341. x2346 connects to a speaker phone and is the preferred number to use. Provide the operator with all the appropriate contact information in case they need to contact you during the run.
- We strongly recommend VNC network displays of observing applications. Suggested VNC setup procedures are provided below. If you have not run VNC before, you are strongly urged to run a test VNC session several days before your observations.
- Start up your observing applications immediately after contacting the operator. These usually include

  cleo Open at least the 'Talk & Draw' windows, then any other cleo application you need. You can use Talk & Draw to communicate with the operator during the run and after you have finished with the phone.

  astrid The GBT observing interface.

  gbtidl The GBT data reduction package. Or another data reduction program to be used

- At the start of your observing time, again call the telescope operator at 304–456–2346, who will put you on a speaker phone. The line should stay open at least through the setup period. (The backup line in case x2346 is busy is the Operator's direct line at 304–456–2341. This line does not have a speaker phone, however.)

- When your observing time starts and the operator gives you permission, you should use Astrid to load your configuration and start observing.

11.2 VNC Setup Instructions

Instructions for opening a VNC session for remote observing can be found at http://www.gb.nrao.edu/gbt/remoteobserving.shtml.

Chapter 12

Planning Your Observations And Travel

12.1 Preparing for Your Observations

After your proposal has been accepted you will be notified of how much observing time you will receive on the GBT. You will also be notified of who your scientific contact person (friend) will be. You should contact your scientific support person well in advance of your observations to help you develop observing strategies and your observing scripts.

We require that new observers (or experienced observers doing new projects outside their previous realm of experience) come to Green Bank for their initial observations. Advisers are also required to accompany their students for their first trip to Green Bank. All policies can be found at https://safe.nrao.edu/wiki/bin/view/GB/Observing/GbtObservingPolicies. You can use the online reservation system at https://bos.nrao.edu/reservations or contact Jessica Taylor at (304)-456-2227, or email jtaylor@nrao.edu to reserve rooms in the Green Bank Residence Hall.

Contact your GBT friend well in advance of the observations to determine the optimum dates for your visit and ensure that the telescope and hardware will be available for the project.

You should plan on arriving in Green Bank at least one full business day before your observations are to begin. This will allow you to meet with your scientific support person as well as the staff support person who will be on-call during your observations.

If your observations are dynamically scheduled and dependent on weather conditions, you should plan to spend at least a week, and preferably two weeks, to increase the likelihood that appropriate conditions for the observations will occur during your visit.

12.2 Travel Support

Some travel support for observing and data reduction is available for U.S. investigators on successful proposals. More information on the travel support that NRAO provides can be found at http://www.nrao.edu/admin/do/nonemployee_observing_travel.shtml.

12.3 Trains, Planes and Automobiles

In principle, observers may use a number of area airports for their travel to Green Bank. These include Washington Dulles, Pittsburgh, Charlottesville, Roanoke (Va.), Charleston (WVa), Clarksburg (WVa) or Lewisberg (WVa). In addition, limited AMTRACK train service is available to Charlottesville and White Sulphur Springs, WVa. Rental cars are available at most of the airports. The Observatory can also send a driver for pickup at any of these airports or stations. It is most convenient if visitors can make travel plans and connections to Charlottesville whenever possible. Transportation will then be available between the NRAO Charlottesville Headquarters office and Green Bank. A GSA vehicle will usually be available at the NRAO Edgemont Road Office Building at Charlottesville for you to drive to Green Bank. Plans to use the GSA vehicle should be arranged with Jessica Taylor in Green Bank (304–456–2227, email jtaylor@nrao.edu). If you do not have your own transportation or cannot make connections through Charlottesville, alternate arrangements may be made with Jessica Taylor.

12.4 Housing

The NRAO operates a residence hall where astronomers may stay while observing or reducing data after completion of their observations. Single rooms (2 beds) are available for $48.00 (+tax) a day single occupancy or $30.00 (+tax) per day per person double occupancy. Students attending a degree conferring college or university and coming to Green Bank to use the telescope will pay single room rate $40.00 (+tax) or $25.50 (+tax) per day per person double occupancy. In addition, there are four one-bedroom apartments with equipped kitchens at $72.00 (+tax) per day. Cribs, high chairs, and fold-up beds are also available. Costs of lodging in NRAO facilities can be waived on request in advance and on approval of the Site Director.

12.5 Getting To Green Bank

12.5.1 Where is Green Bank?

A map showing the location of Green Bank relative to major, nearby towns and cities is shown in Figure 12.1. Simplified directions are also shown in Figure 12.1.

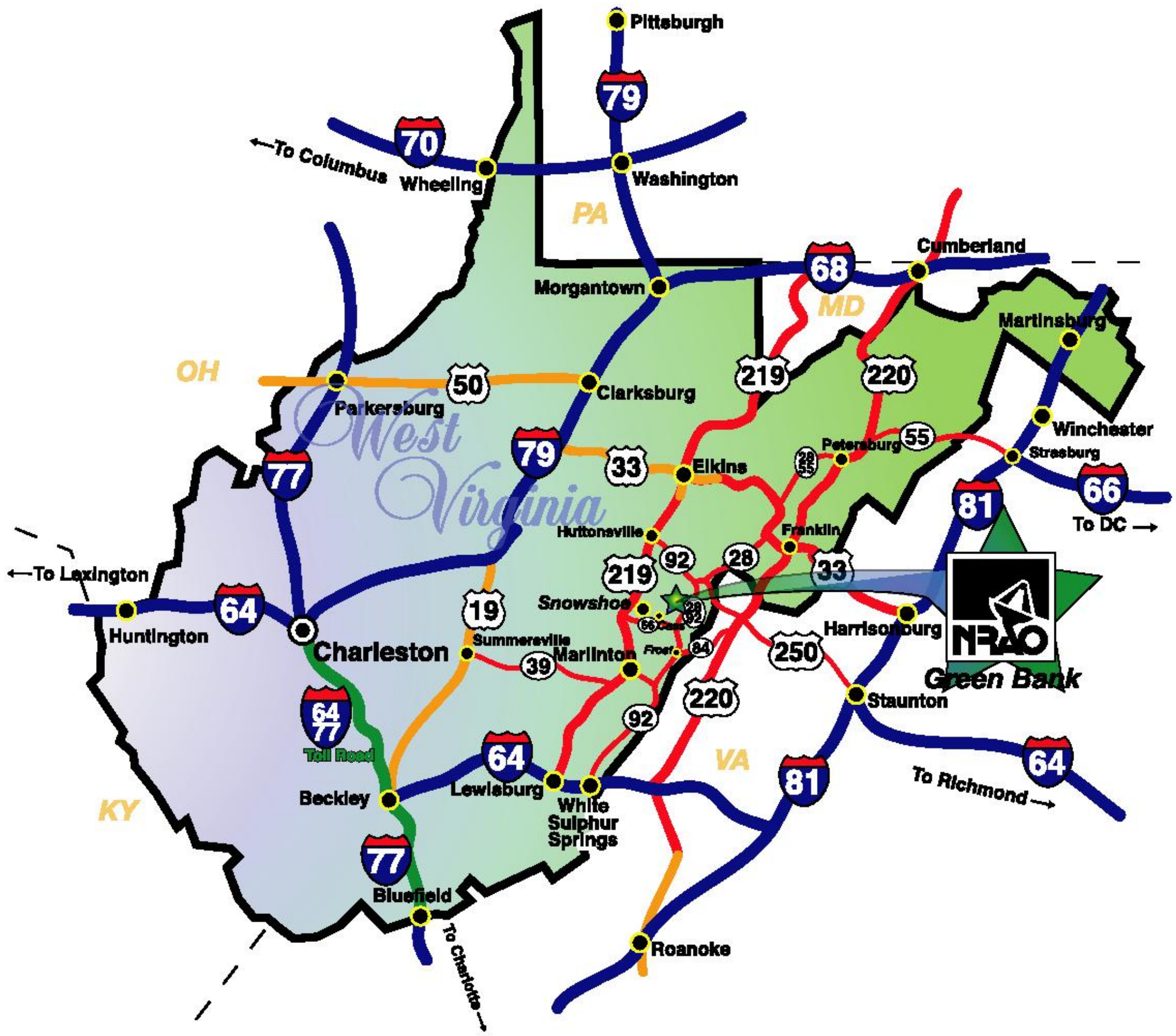
Green Bank is located in Pocahontas County, WV, very close to the Virginia border and at about the mid-point of the full extent of the Virginia–West Virginia border.

12.5.2 Directions to Green Bank

The following are directions to Green Bank from airports in Pittsburgh, Pa, Washington, DC, Charlottesville, Va and Roanoke, Va. The duration of the drive from either Pittsburgh or Washington is four to five hours. The duration of the drive from Charlottesville and Roanoke is about 2–1/2 hours.

12.5.2.1 Beware of GPS!!

GPS systems, Mapquest, and other such automated route finding systems are notoriously unreliable within 50 miles of the Observatory. Some roads that are recommended by these systems are passable only with 4-wheel drive vehicles. Do not turn onto unpaved roads !



Directions to the National Radio Astronomy Observatory Green Bank, West Virginia

From Charleston, WV:
Via I-64 East, exit at White Sulphur Springs (Exit 175) and take Rt. 92 North to Green Bank.

From Roanoke, VA:
Via I-64 West, exit at White Sulphur Springs (Exit 181) and take Rt. 92 North to Green Bank.

From Pittsburgh:
Via I-79 South, exit at Weston/Buckhannon (Exit 99), and travel US 33 East to Elkins, then take Rt. 92 South to Green Bank.

From Washington DC:
Via I-66 West to I-81 South. ***Option 1:*** Take I-81 South for ~3 miles exit at Strasburg (exit 296) to Route 55. Stay on Route 55 through Wardensville, WV, Moorefield, and Petersburg, then Rt. 28 South to Green Bank.

From Washington DC:
Via I-66 West to I-81 South. ***Option 2:*** Take I-81 South to Harrisonburg (exit 247) take the truck bypass around Harrisonburg to US 33 West. Take US 33 West through Franklin, then take Rt. 28 South at Judy Gap to Green Bank.

From Charlottesville/ Richmond:
Via I-64 West to Staunton, then I-81 North for ~3 miles to Exit 225 (Woodrow Wilson Parkway). Take the Parkway around Staunton to US 250 West. Stay on US 250 West to Monterey, VA. At Monterey take Rt. 220 South ~3.5 miles to Rt. 84 West. Take 84 West to Frost, WV. At Frost, take Rt. 92 North to Green Bank.

Figure 12.1: Direction to Green Bank.

12.5.2.2 Pittsburgh to Green Bank

From the Greater Pittsburgh International Airport, go east on route 60 to US 22/30. Follow 22/30 east to Interstate 79. Take I-79 south through Clarksburg, WV to the US 33 exit (exit 99) near Buckhannon, WV. Go east on US 33 through Buckhannon to Elkins, WV. Turn south on US 250/219 to go to Huttonsville. In Huttonsville, take US 250 (route 92) southeast to Bartow. Follow route 92/28 south to Green Bank.

12.5.2.3 Washington Dulles or National to Green Bank

From the Washington Dulles International Airport, go south on route 28 to Interstate 66.

From the Washington National Airport, take US 1/Va 110/US 50 to Interstate 66 (ask at airport for exact details).

Take I-66 west to I-81. From here, there are two alternative routes to Green Bank:

a) Follow I-81 south to the exit for route 55 near Strasburg, VA. Go west on 55 to Moorefield, WV. Turn south on US 220/55, and drive to Petersburg, WV. Go south on route 28/55 to Seneca Rocks. Continue south on 28 through Judy Gap to Bartow. Follow route 92/28 south to Green Bank.

b) Follow I-81 south to Harrisonburg, VA. Go west on US 33 through Franklin, WV to Judy Gap. Turn south on route 28, and drive to Bartow, WV. Follow route 92/28 south to Green Bank.

Route a) is the preferred route. Most of route 55 in WV has been upgraded to a new 4-lane highway and this is where you cross most of the mountains in getting to Green Bank.

12.5.2.4 Charlottesville to Green Bank

From the Charlottesville-Albemarle Airport, go east (straight) on Airport Road to US 29. Take 29 south to US 250, and follow 250 west to Interstate 64. Go west on I-64 to Interstate 81 near Staunton, VA. After traveling north on I-81 for about two miles, take the Woodrow Wilson Parkway exit. Go west on the parkway to US 250, and follow 250 west to Monterey, VA. In Monterey, turn south on US 220, and shortly thereafter, veer west on route 84 to go to Frost, WV. Follow route 92/28 north to Green Bank.

12.5.2.5 Roanoke to Green Bank

From the Roanoke Airport go left (south) on Valley View Drive/Airport road and then almost immediately turn right onto Hershberger Road. Go about 1/2 mile and then take I-581 north to I-81 north. Go two exits on I-81 north to US 220, the Daleville/Troutville exit. Take US 200 north toward (and through) Fincastle until you reach I-64. Take I-64 west to White Sulpher Springs, WV. Take the first White Sulpher Springs exit and turn right. After about 1/2 mile turn right onto route 92 north and this will take you to Green Bank.

12.5.3 Once You Are in Green Bank

The entrance to the observatory is about one-half mile north of "downtown" Green Bank on the west side of route 92/28 (see Figure 12.2). Look for the Jansky and Reber antennas. They are located on either side of the entrance.

To pick up your keys you will need to come to the Jansky Lab. The Jansky lab is the second building on the left as you enter the site. Go in the "atrium" main entrance. If after working hours, use the

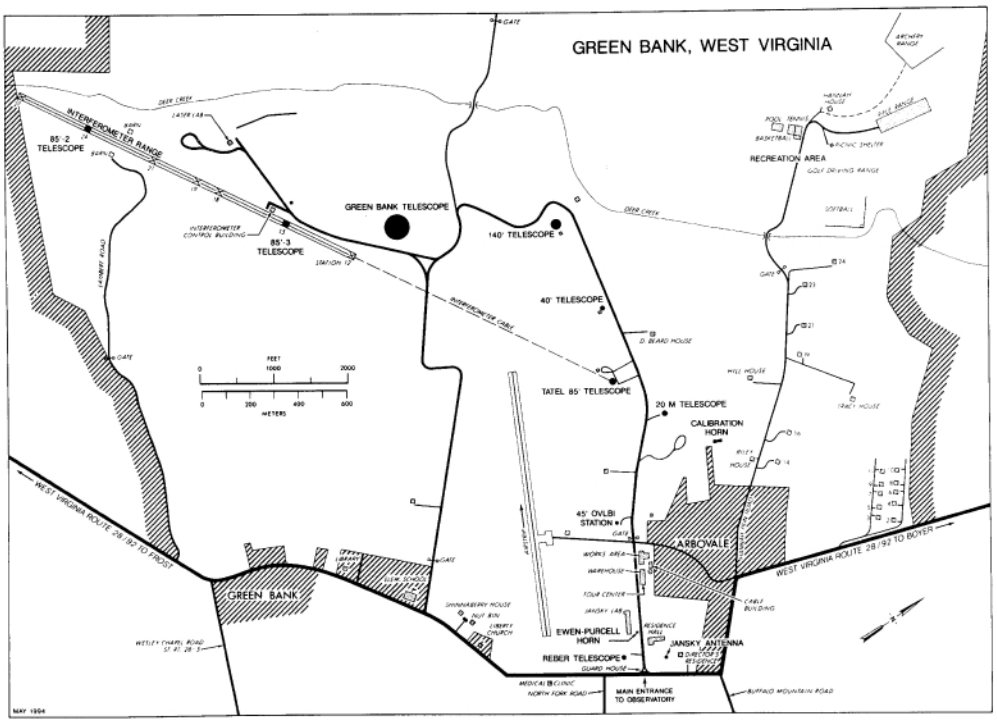


Figure 12.2: NRAO, Green Bank site map.

telephone on the left to call the GBT operator at 2341. The operator will then “buzz you in.” If during normal working hours, just walk in. The room packets and keys will be found near the begining of the hallway on your right as you enter the Jansky Lab.

The residence hall is the first building on the right as one enters the observatory. Adequate parking is provided on the west side of the residence hall. Enter the residence hall through the double glass doors on the west side of the building. The observer’s lounge is located on the second floor of the residence hall, directly above the entrance.

Chapter 13

After Your Observations

13.1 Taking your data home

There are several methods that you could use for taking you data home with you. We have the capabilities of writing ExaByte, Dat, Mammouth, and DLT tapes. We can also write your data onto CDs or DVDs. Please contact your scientific support person to help you decide which method will be best for you.

NOTE: When using the scp utility to copy your data to another machine please limit your bandwidth to 2000 Kbit/s. For example:

```
[you@yourmachine ~]$ scp -l 2000 you@GBTmachine:data.fits datacopy.fits
```

13.2 Installing GBTIDL

Most spectral line observers will use GBTIDL to reduce their data. If you have an idl license at your home institution then you can obtain a copy of GBTIDL from http://gbtidl.nrao.edu. Instructions for installation can also be found at the above web-page.

13.3 Keep Your Contact Person Informed

Don't hesitate to ask your scientific contact person if you are having trouble reducing your data, or if you have questions about your data. It does not matter how long its has been since you observed, your contact person will be more than happy to help you.

13.4 Press Releases and News-worthy Items

News-worthy items should be discussed with the NRAO press officer. The press officer can help write an NRAO press release or a press release from your home institution. For more information see page 30 of the January 2007 version of the NRAO Newsletter (http://www.nrao.edu/news/newsletters/nraonews110.pdf).

13.5 Publishing Your Results

Finally you should publish your results. The NRAO will help with page charges for the publication of the results from your observations. Please see http://www.nrao.edu/library/page_charges.shtml for more details. **Please inform your scientific contact person of any publication resulting from your observations.**

Chapter 14

Pulsar Observing with GUPPI

The Green Bank Ultimate Pulsar Processing Instrument (GUPPI) has one hardware mode and many software modes. GUPPI can be used with any receiver with the exception of MUSTANG. Only one polarization would be available for the Ka-band receiver. GUPPI uses 8-bit sampling to dramatically improve upon the dynamic range and RFI resistance of the Spectral Processor. Currently GUPPI can use bandwidths of 100, 200 and 800 MHz total intensity (i.e., 2 polarizations summed) or full stokes parameters. The number of spectral channels per polarization may be 64, 128, 256, 512, 1024, 2048, or 4096. The minimum integration time is 40.96μ s using 2048 channels and an 800 MHz bandwidth.

Advanced modes, such as real-time coherent de-dispersion or baseband recording, are available to expert users. To use them, consult Scott Ransom (sransom@nrao.edu) or Paul Demorest (pdemores@nrao.edu).

For reference, refer to the web pages:
General Information: https://safe.nrao.edu/wiki/bin/view/CICADA/NGNPP and
https://safe.nrao.edu/wiki/bin/view/CICADA/GUPPIAstridGuide
for the most up-to-date version of the information in this chapter.

14.1 Summary

Since Astrid can now control GUPPI, the key thing to remember is that you set everything up via Configure() commands in Astrid. Data is then taken by running an Astrid script. For the overwhelming majority of pulsar observations, this simply means running a Track() command on your source of choice.

14.2 An Example Configuration

The following shows a scheduling block to configure GUPPI to take S-band search-mode data. All of the available parameters are shown, even though not all of them are used for this configuration. For instance, since this is a search-mode observation, none of the guppi.fold* parameters are used. There are three types of GUPPI observations:

- **search** - Which is "normal" search-mode data, meaning writing spectra rapidly to disk. Data are written in PSRFITS search-mode format and can be analyzed using SIGPROC and PRESTO.

- **fold** - Which is fold-mode data, meaning folding data at a known pulsar ephemeris modulo the pulse period for each spectral channel. Data are written in PSRFITS fold-mode format and can be analyzed using PSRCHIVE.
- **cal** - Which is a special-case of fold-mode data, where the 25 Hz pulsed cal signal is folded and saved. It can be used for flux and polarization calibration and is analyzed with PSRCHIVE. Cal scans are very easy now (and the configure blocks themselves turn the cals on and off so you won't forget) and we highly recommend that people start using them, if only for sanity checks of the system. (you can view cal scans using the PSRCHIVE commands: pav -X filename, pav -SFT filename, or psrplot -pC filename. Each shows a slightly different view of the cal file.)

```
#
# An example well-documented S-band "search"-mode script
#
config_g="""
# usually 'Rcvr_342', 'Rcvr_800', 'Rcvr1_2', 'Rcvr2_3', 'Rcvr4_6'
receiver = 'Rcvr2_3'
restfreq = 2000.0, 2000.0   # in MHz.  Must have 2 identical freqs
obstype = 'Pulsar'
# talk to Scott if you want 'GUPPI/GASP' or others as well...
backend = 'GUPPI'
pol = 'Linear'       # C-band and below are native 'Linear'
ifbw = 0             # 0 for >100MHz BW modes, 80 for 100MHz.
bandwidth = 800      # in MHz. 100, 200, or 800 currently
tint  = 64e-6        # sample time in seconds (very flexible)
swmode = 'tp_nocal' # 'tp' for cals, 'tp_nocal' for no cals
noisecal = 'off'     # if no cals, set to 'off', else 'lo'
# The following are boilerplate until 'guppi' section
# You should probably not change them...
swtype = 'none'
swper = 0.04
swfreq = 0.0, 0.0
nwin = 1
deltafreq = 0,0
vlow = 0
vhigh = 0
g.vframe = 'topo'
vdef = 'Radio'
# -- GUPPI specific params -- #
# obsmode can be 'search', 'fold', or 'cal'
guppi.obsmode = 'search'
# numchan can be a power-of-two between 64 to 4096
guppi.numchan = 2048
# polnmode is 'full_stokes' or 'total_intensity'
guppi.polnmode = 'total_intensity'
# scale should be set in first config block and
# tweaked while taking data and viewing with guppi_monitor
guppi.scale = 9.0
guppi.outbits = 8          # Currently only 8 is available
# Folding specific params -- not needed for cal or search
guppi.fold_dumptime = 10  # in sec
guppi.fold_bins = 256     # number of bins in profile
# Make sure that the parfile exists!
guppi.fold_parfile = "/users/sransom/parfiles/1713.par"
# Top level disk where data will be written
```

```
guppi.datadisk = 'data2'  # 'data1' or 'data2'
"""
Configure(config_g)
```

Most of the parameters are self explanatory, however, a few need some further explanation.

- **tint** - The integration time for each spectrum. The formula is

$$tint = acc\_len * guppi.numchan/bandwidth$$

where the values of acc_len can vary from 2 in special cases (more typically 4 or 6) up to 1024 for each BW setting of GUPPI (100 MHz, 200 MHz, or 800 MHz, currently). The fastest limits are based on writing $< 200MB/s$ to disk, while the slowest limits can be a few tenths to several tens of milliseconds depending on the bandwidth and the number of channels, guppi.numchan.

- **ifbw** - This parameter must be set to 80 (MHz) when using the 100MHz bandwidth modes as the GBT currently does not have 100MHz bandpass filters. Also, it is highly recommended that it is set to 0 when using other modes as that will prevent previously set values of ifbw from giving you strange bandpasses.

- **guppi.fold_dumptime** - For fold or cal observations, this is how much we will integrate the pulsar (or cal) before dumping a set of profiles to disk. It must be shorter than the scan duration that you set via the Track() command.

- **guppi.datadisk** - This is the top-level directory (i.e. RAID array, 'data1' or 'data2') where your data will be stored. It will go in a subdirectory called /guppi.datadisk/observername/projectID/date/. The data will be owned by monctrl and so you will not be able to remove it – that means you may be bugged mercilessly until you process your data!

14.3 Status Monitoring

When you observe using GUPPI with Astrid, you must first make sure that you have several xterm's open on beef ("ssh beef") where you setup the GUPPI environment using
source /opt/64bit/guppi/guppi_daq/guppi.bash or
source /opt/64bit/guppi/guppi_daq/guppi.csh depending on your shell.

In one of them, monitor the GUPPI shared memory buffers using **guppi_status** (see Figure 14.1). The bottom of that screen will tell you if you are taking data (lots of numbers changing) and the top of the screen shows all of the key GUPPI parameters.

14.4 Setting Levels

Before you attempt to balance the system, you must first configure the system. Do that by running a GUPPI config block with receiver, center frequency, and bandwidth settings as appropriate for your session (we recommend using a 'cal' mode block) which will cause data to begin flowing internally through the GUPPI hardware. Then use the astrid Slew() command to slew to your source of interest, and finally, balance the system with the astrid Balance() command. All of these things can be done in a single astrid scheduling block.

At this point, the input levels are set, and we need to set the internal scaling of GUPPI via the

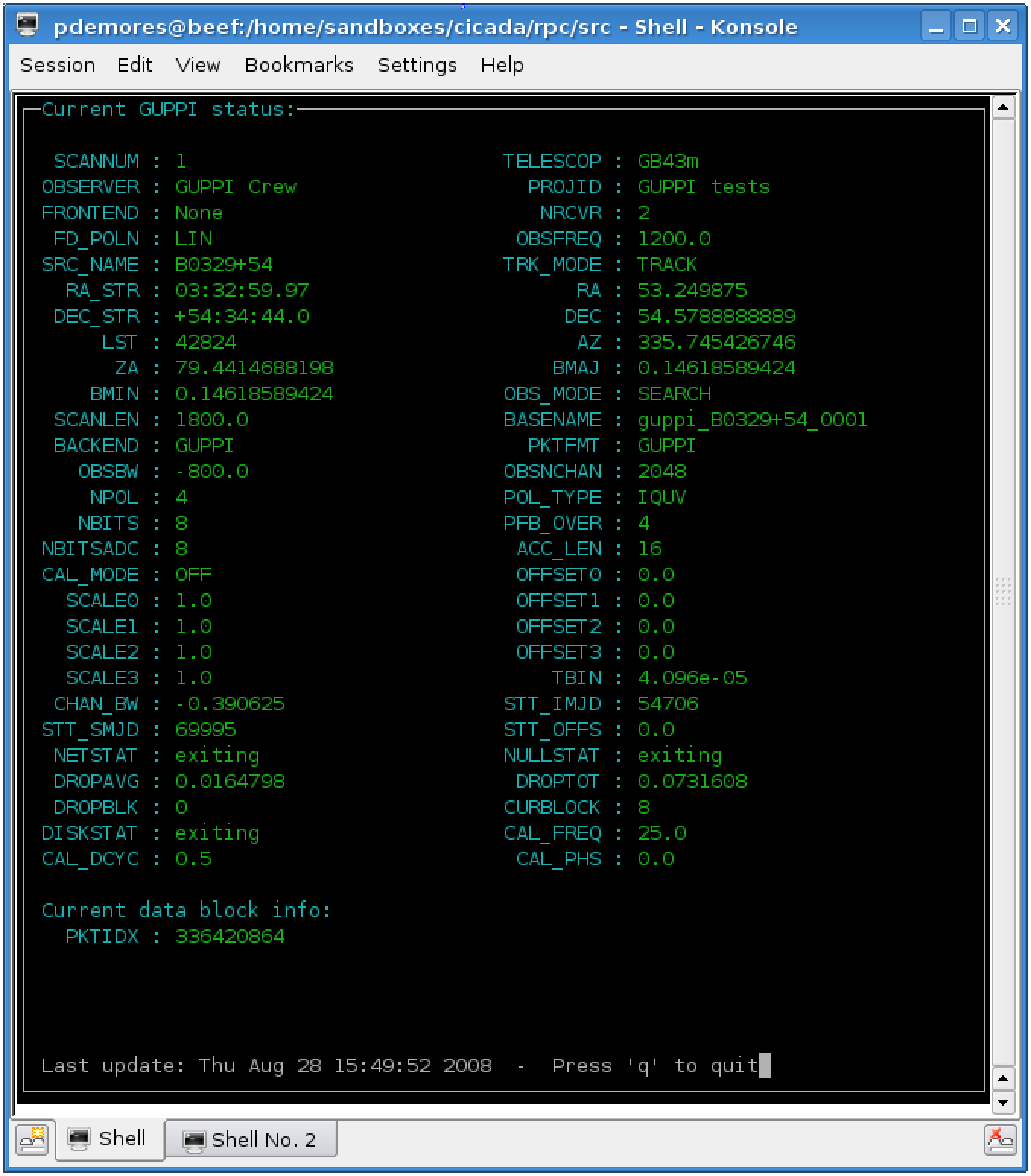


Figure 14.1: The GUPPI Status Display screen.

guppi.scale parameter. The example blocks (see section 14.7) should have reasonable starting values for that parameter. To do this, run a Track() scan, and once data begins flowing (which you can tell via guppi_status) start up an guppi_monitor instance so that you can see the bandpass. Decide on how much you need to increase or decrease the scaling linearly, and change the guppi.scale parameter appropriately. When the scan is over (or aborted), re-configure and re-run the Track() scan. guppi_monitor should now show "good" levels for the bandpass. Remember that we would like the average passband to be between 30-50 or so. Since these scans had incorrect scaling, they should not be used in your data processing, so make sure that if you are saving real cal scans, take a new one once guppi.scale is properly set. The bandpass for saved cal (and fold-mode) files can be plotted using psrplot -pb filename.

ALSO: Be careful when copying scheduling blocks! They might have bad values of guppi.scale in them! Once you set guppi.scale for your observing session, make sure that the other configurations either do not have it set (which means that it will continue to use the currently set value) or else have it set to the new correct value!

14.5 Taking Data

Once you have the input levels of GUPPI set, you are ready to take real data. That is accomplished by configuring your scan and then running a Track(). The scan durations are set in seconds, and they determine how much data you will take. Note that for short scans, you should set the duration about 5 seconds longer than the amount of data that you really want. For example, if you want 6x10-sec dumps for a cal scan, set the Track() scanDuration parameter to 65. An example of a scheduling block to track a well known MSP is:

```
#
# Slew, balance, then take data...
#
bright_MSPs = Catalog(pulsars_bright_MSPs_GBT)

Slew("B1937+21")
Balance()

# Track is how we take data now.
# Scan duration is in sec.  Recommend you
# add 5-sec to account for some delays in the system

Track("B1937+21", endOffset=None, scanDuration=65)
```

Similarly, if you want to take driftscan data (or need to test GUPPI on a maintenance day when the telescope cannot point), we can tell the GBT to track the current settings of the azimuth and elevation encoders (i.e. telling the GBT not to move):

```
#
# Take Drift-scan data
#

Balance()
loc = GetCurrentLocation("Encoder")
Track(loc, endOffset=None, scanDuration=20000)
```

You can interrupt a scan by using the Astrid **Abort** button if you need to stop a scan early.

14.6 Data Monitoring

You can watch the standard output of the GUPPI data acquisition server by tailing the log file which is written in /tmp on beef. It is highly recommend that you do this as it will show you if you drop a lot of packets, if your data rate is too high, and/or too many others are working on beef. Which reminds me: Always nice your jobs on beef!

```
> tail -f /tmp/guppi_daq_server.log
```

14.7 Other Examples

There are several other example configurations which you can copy, load into Astrid, or simply browse in /users/sransom/astrid. They are:

- /users/sransom/astrid/GUPPI_astrid_example.py The well-documented S-band search-mode example from above
- /users/sransom/astrid/GUPPI_astrid_820MHz_cal.py A 200-MHz BW 'cal' scan using the PF1 receiver at 820 MHz
- /users/sransom/astrid/GUPPI_astrid_820MHz_fold.py A 200-MHz BW 'fold' scan of a bright MSP using the PF1 receiver at 820 MHz
- /users/sransom/astrid/GUPPI_astrid_Xband_fastdump.py A special 256-channel fast-dump mode at X-band for Crab Giant Pulses
- /users/sransom/astrid/GUPPI_astrid_350MHz_fastdump.py A way to dump at 81.92us for 100MHz-BW mode data for searching
- /users/sransom/astrid/GUPPI_astrid_slew+takedata.py The Slew() and Track() example from above
- /users/sransom/astrid/GUPPI_astrid_driftscan.py The Track() for driftscans example from above

14.8 Warnings

- Do not run any commands from the GUPPI prompt!
- Do not run guppi_set_params from the command line at all! This is all handled by configuring in Astrid now.

Chapter 15

The CalTech Continuum Backend (CCB)

The Caltech Continuum Backend (CCB) is a dedicated continuum backend for the GBT Ka-band receiver, built in collaboration with A.C.S. Readhead's radio astronomy instrumentation group at Caltech and commissioned on the GBT in 2006. The driving consideration behind its design is to provide fast electronic beam switching in order to suppress the electronic gain fluctuations which usually limit the sensitivity of continuum measurements with single dish radio receivers. To further improve stability, it is a *direct detection* system: there are no mixers before the conversion from RF to detected power. The Ka-band receiver provides *eight* simultaneous, directly detected channels of RF power levels to the CCB: one for each feed, times four frequency channels (26-29.5 GHz; 29.5-33 GHz; 33-36.5 GHz; and 36.5-40 GHz). Astronomical information and labels for these 8 channels (or "ports" in GBT parlance) is summarized in Table 15.

The following sections outline the process of observing with, and analyzing the data from, the CCB. Much of the information in this chapter is also maintained at

`/users/bmason/ccbPub/README.txt`

which is convenient, for instance, for cutting and pasting data analysis commands. Template scheduling blocks are also in this directory.

Port	Beam	Polarization	Frequency
9	1	Y	38.25
10	1	Y	34.75
11	1	Y	31.25
12	1	Y	27.75
13	2	X	38.25
14	2	X	34.75
15	2	X	31.25
16	2	X	27.75

Table 15.1: CCB Port labels and the astronomical quantities they measure.

15.1 Observing with the CCB

15.1.1 Configuration

Configuration of the CCB is straightforward, and for most purposes the only two configurations needed are provided in the two configuration files

`/users/bmason/ccbPub/ccb.conf`

and

`/users/bmason/ccbPub/ccbBothCalsLong.conf`.

These differ only in the duration of the integrations: the former configures for 5 millisecond integrations, which is useful for estimating the scatter in the samples to obtain meaningful χ^2 values in the analysis of science data; the later configures for 25 millisecond integrations, which is useful in peak/focus observations to speed up processing of the data (see Chapter 4, section 4.1.3). `ccb.conf` is reproduced and explained below.

The following keywords tell ASTRID to expect beamswitched continuum observations with Ka and the CCB:

```
receiver='Rcvr26_40'
beam='B12'
obstype='Continuum'
backend='CCB'
nwin=4
restfreq=27000,32000,35000,38000
deltafreq=0,0,0,0
bandwidth=600,600,600,600
swmode=''sp''
swtype=''bsw''
pol=''Circular''
vdef=''Radio''
frame=''topo''
```

they do not have any practical effect on the actual instrument configuration but are necessary to set up internal variables and insure the recorded FITS files are accurate.

The following keywords configure the CCB itself:

```
ccb.cal_off_integs=20
ccb.XL_on_integs=2
ccb.both_on_integs=2
ccb.YR_on_integs=2
ccb.bswfreq=4
tint=0.005
```

The meaning of these keywords is as follows:

- The first four specify the cal firing pattern.
- `ccb.bswfreq` specifies the beam switching frequency in kHz. 4 kHz is standard; the "> 10% blanking" warning which results is also standard and may be safely ignored.
- `tint` is the integration time in seconds.

15.1.2 Pointing & Focus

The online processing of pointing and focus data is handled by GFM (which runs within the Astrid Data Display window) similarly as for other GBT receivers and the DCR. A few comments:

- because the Ka-band receiver currently only has one polarization per beam, GFM will by default issue some complaints which can be ignored. These can be eliminated by choosing "Y/Right" polarization in the Astrid Data Display window (see Chapter 4, section 4.1.3.3) under Tools − > Options − > Data Processing.
- in the same menu (Tools − > Options − > Data Processing), choosing "31.25 GHz" as the frequency to process, instead of the default 38.25 GHz, can improve robustness of the result.
- The results shown in the Astrid Display are in *raw counts*, not Kelvin or Janskys.
- Choosing "Relaxed" heuristics is also often helpful.

There is a template pointing and focus SB for the CCB in `/users/bmason/ccbPub` called `ccbPeak.turtle`. This scheduling block does a focus scan, four peak scans, and a symmetric nod (for accurate photometry to monitor the telescope gain).

15.1.3 Observing Modes & Scheduling Blocks (SBs)

Science projects with the CCB typically fall into two categories: mapping, and point source photometry. The majority of CCB science is the latter, since this is what, by design, it does best. Template scheduling blocks for both are in `/users/bmason/ccbPub`, called:

- `ccbObsCycle.turtle`: perform photometry on a list of sources.
- `ccbRaLongMap.turtle`: perform a standard `RALongMap` on a source (see 5.2.2.4.
- `ccbMap.turtle, ccbMosaicMap.turtle`: make maps using longer, single-scan, custom raster maps. Your staff friend will help tailor these to your project's needs, should you choose this approach.

Point source photometry is accomplished with an On-the-fly variant of the symmetric NOD procedure described in 5.2.2.3. This procedure, which we refer to as the OTF-NOD, alternately places the beam in each of the two beams of the Ka band receiver in a B1/B2/B2/B1 pattern. This sequence cancels means and gradients in the atmospheric or receiver emission with time. Plotting the beamswitched data from this sequence produces a sawtooth pattern shown in Figure 15.1; this is discussed more in § 15.1.5.Each NOD is 70 seconds long (10 seconds in each phase, with a 10 second slew between beams and an initial 10 second acquire time).

Note: OTF-NOD is not one of the standard scan types; it is implemented in the scripts mentioned here (e.g., "ccbObsCycle.turtle").

15.1.4 Calibration

If at all possible, be sure to do a peak and focus, and perform photometry (an OTF-NOD, as implemented in `ccbObsCycle.turtle` or `ccbPeak.turtle`) on one of the following three primary (flux) calibrators: 3c48, 3c147, or 3c286. This will allow your data to be accurately calibrated (our calibration scale is ultimately referenced to the WMAP 30 GHz measurements of the planets). If this is not possible the calibration can be transferred from another telescope period (observing session) within a few days of the session in question.

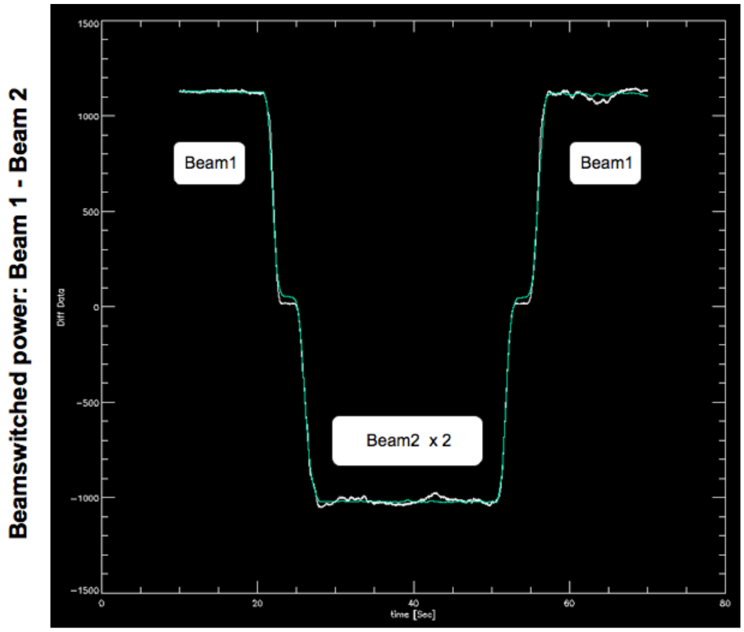


Figure 15.1: Data from a CCB, beamswitched OTF-NOD, showing data and model versus time through one B1/B2/B2/B1 scan. The white line is the CCB beamswitched data and the green line is the fit for source amplitude using the known source and telescope (as a function of time) positions.

15.1.5 Online Data Analysis

It is important to assess data quality during your observing session. There are a set of custom IDL routines for analyzing CCB data; if you use the observing procedures and config files described here, your data should be readily calibratable and analyzable by them. To use the IDL code, start IDL by typing (from the GB UNIX command line)

```
/users/bmason/ccbPub/ccbidl
```

Here is an example data reduction session that provides a quick look at your data:

```
; set up global variables
;  don't write files or plots to disk...
proj='AGBT06A_049_09'
setccbpipeopts,gbtproj=proj,ccbwritefiles=0,$
  gbtdatapath='/home/archive/science-data/tape-0016/'
; to use postprocessing scripts, set ccbwritefiles=1

; a good color table for the plots:
loadct,12

; create an array indexing scan numbers
;  to file name
indexscans,si

; summarize the project
summarizeproject

; read a nod observation from scan 12
```

```
readccbotfnod,si[12],q
; fit the data, binning integrations to 0.5sec bins
fitccbotfnod,q,qfit,bin=0.5
; the resulting plot shows the differenced
;  data (white) and the fit to the data (green)
;  for each of 16 CCB ports. (the first 8 are blank)

; look at the next nod that just came in
;  this time calibrate to antenna temperature
;  before plotting
; First you need to derive a calibration, which
;  requires a scan with both cals firing independently.
; /dogain tells the code to solve for the calibration;
;  the results are stored in calibdat, which we can
;  pass into subsequent invocations of the calibration.
indexscans,si
readccbotfnod,si[13],q
calibtokelvin,q,/dogain,calibdat=calibdat
fitccbotfnod,q

; the scan index si must be updated to read in scans
;  collected after it was first created
indexscans,si
readccbotfnod,si[14],q
; and calibrate to kelvin using the information
;  we just derived
calibtokelvin,q,calibdat=calibdat
; fit/plot
fitccbotfnod,q

; et cetera...
```

Example OTF-NOD data for bright sources (under good and poor conditions) and a weak source (under good conditions) are shown in Figures 15.2 through 15.5

Mapping data can also be imaged using the IDL tools:

```
; make a map from scans 7-10 using port 11 data
;  (note the port must be specified; valid ports are
;   9-16)
img=makedcrccbmap([7,8,9,10],/isccb,port=11)
; replot the map
plotmap,img,/int
; make a png copy of it
grabpng,'mymap.png'
; save the map in standard FITS format--
saveimg,img,'mymap.fits'
```

This will be a *beamswitched* map. The beamswitching can be removed by an EKH[1] deconvolution algorithm also implemented in the code. Your GBT friend will help you with this, if needed.

[1]Emerson, Klein, Haslam 1979 A&A **76**,92.

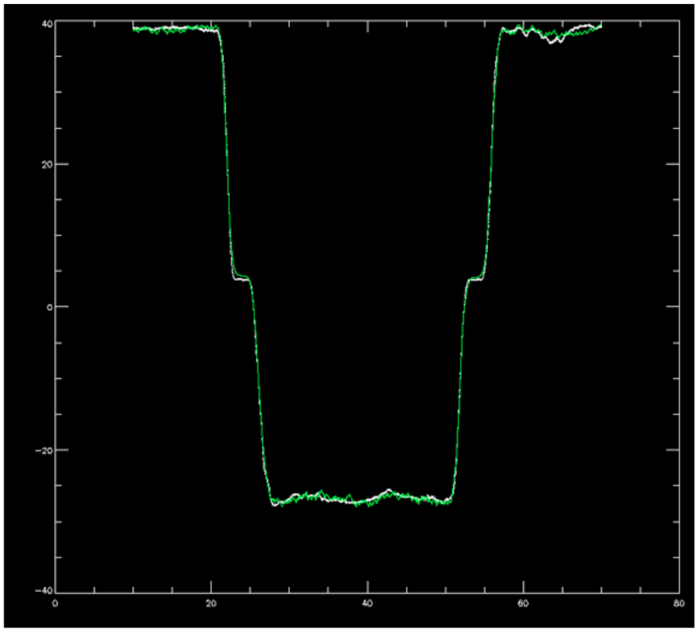


Figure 15.2: CCB data from an OTF-NOD observation of a bright source, showing data and model versus time through one B1/B2/B2/B1 scan. The white line is the CCB beamswitched data and the green line is the fit for source amplitude using the known source and telescope (as a function of time) positions. The close agreement between the data and the fit indicate that neither fluctuations in atmospheric emission nor pointing fluctuations (typically due to the wind on these timescales) are problems in this data.

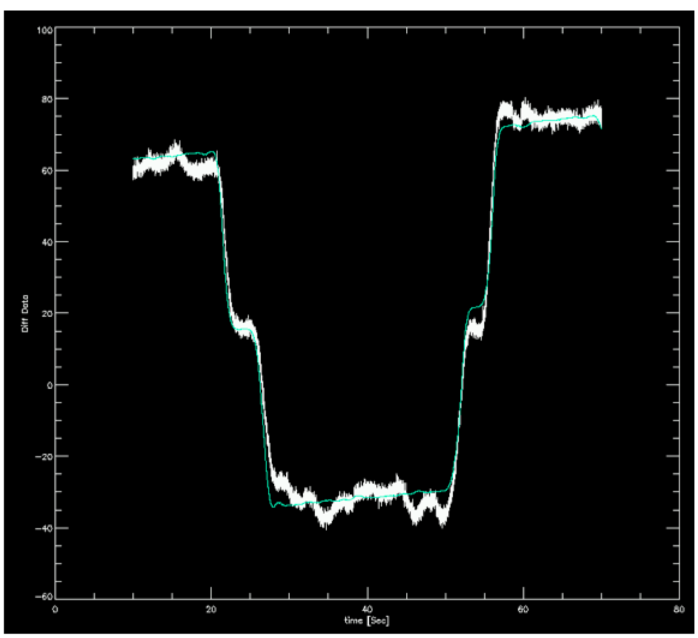


Figure 15.3: CCB OTF-NOD data on a bright source under marginal conditions.The differences between the data and the model are clearly larger in this case.

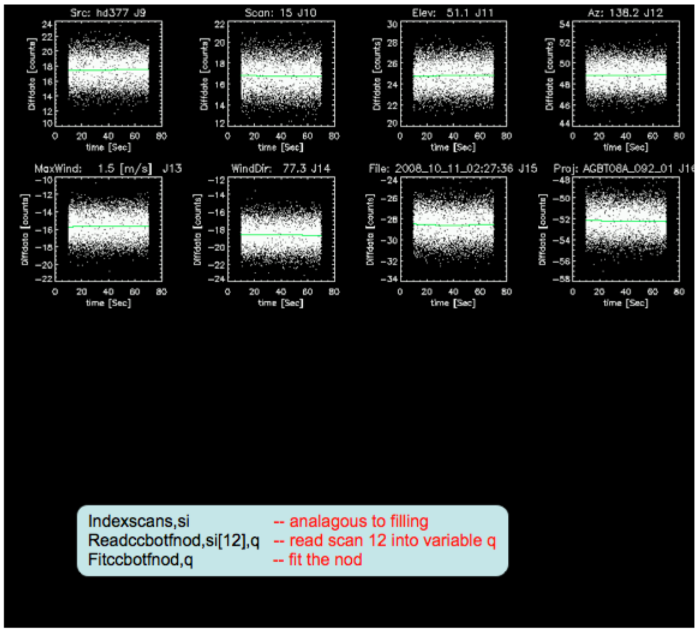


Figure 15.4: CCB OTF-NOD measurement of a weak (mJy-level) source under good conditions. The IDL commands used to obtain this plot are shown inset.

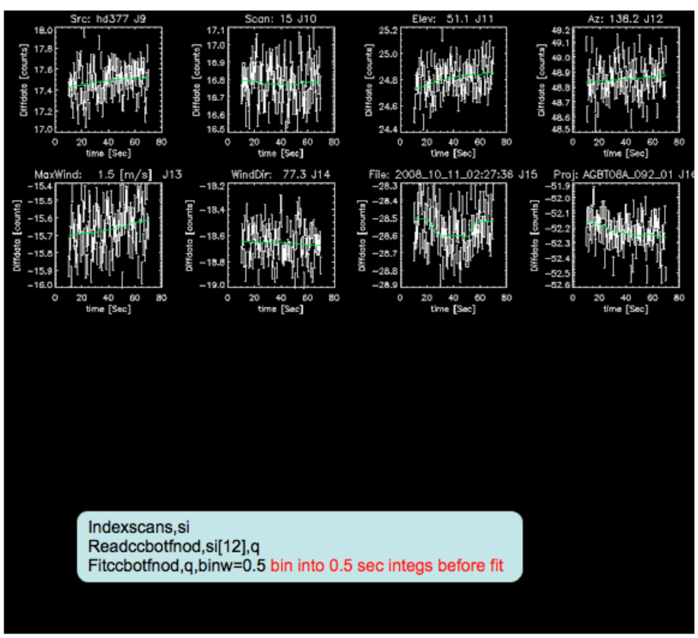


Figure 15.5: The same weak-source data, this time with the individual integrations binned into 0.5 second bins (using `fitccbotfnod`'s `binwidth` optional argument in seconds) so the thermal-noise scatter doesn't dominate the automatically chosen y-axis scale. This better shows any gradients or low-level fluctutions in the beamswitched data (due, for instance, to imperfect photometric conditions). In this data they are not significant.

15.2 Performance

Recent tests under excellent conditions show a sensitivity of 150 μJy (RMS) for the most sensitive single channel (34 GHz), or 100 μJy (RMS) for all channels combined together. These are the RMS of fully-calibrated, 70-second OTF-NODs on a very weak source. Typical "reasonable-weather" conditions are a factor of two worse. Improvements to the receiver made since these data were acquired may result in better sensitivity for the 2010/2011 season.

15.3 Differences Between the CCB/Ka System and other GBT Systems

There are a few differences between the CCB/Ka system and other GBT receiver/backend systems which users familiar with the GBT will want to bear in mind.

- Because it is a direct detection system, the GBT IF system does not enter into observing.
- The Ka/CCB gains are engineered to be stable (10% - 20% over months), so no variable attenuators are in the signal chain. Consequently there is no `Balance()` step.
- To optimize the RF balance (for spectral baseline and continuum stability), the OMT's have been removed from the Ka band receiver. It is therefore sensitive to *one linear polarization per feed*. The two feeds are sensitive to orthogonal linear polarizations (X and Y).
- Feed orientation is 45° from the Elevation/cross-Elevation axes. All other receivers have feed separations that are parallel to the Elevation or cross-Elevation axes (except for the K–band Focal Plane Array).
- There are two cal diodes (one for each feed), and they are separately controlled (*i.e.*, it is possible to turn one on and not the other). Cals are ON or OFF for an entire integration; they are not pulsed ON and OFF within a single integration.

Chapter 16

MUSTANG

MUSTANG (MUltiplexed SQUID TES Array at Ninety GHz), the GBT's first 3mm instrument, comprises a nearly fully-sampled array of 64 Transition-edge Superconducting (TES) bolometers which provide a $9''$ beam on the sky and instantaneously measure a $\sim 40'' \times 40''$ field-of-view. It was built at the University of Pennsylvania by PI Mark Devlin's group, in collaboration with NIST, NASA, NRAO, and Cardiff University. It is now a facility instrument on the GBT. This chapter of the Observing Guide describes how to observe with MUSTANG on the GBT.

16.1 Conditions Affecting MUSTANG Observations

This section outlines the factors which can affect the efficiency and success of MUSTANG observations.

16.1.1 Weather & Solar Illumination

Observations at 90 GHz are affected by clouds and water vapor, which attenuate the astronomical signal and contribute spurious emission. As a rule of thumb, data obtained with zenith 90 GHz sky brightnesses (at zenith) under 35 K provide good data, and data obtained with 90 GHz sky brightnesses under 50 K provide usable data for some of the easier types of projects (compact or bright sources, for instance). Due to the fact that MUSTANG instantaneously samples many points on the sky, and the GBT beams traverse nearly identical paths through the atmosphere, the spurious emission contributed by the atmosphere can be effectively removed by variants of a common-mode subtraction implemented in the data reduction routines. The penalty which results is that astronomical structures on scales much larger than one instantaneous FOV are removed from the map. If the weather is clear and stable, however, the common mode subtractions can be less aggressive and larger structures can be reliably imaged. *The main effects of degraded weather will therefore be loss of large-scale structure in the maps and further attenuation of the astronomical signal.* In the case of poor weather (rain, heavy and variable cloud cover) a large and variable attenuation can prevent reliable flux calibration, so observing is not recommended.

A device measuring the net near-IR irradiance of the sky (called a pyrgeometer) has been installed near the GBT and provides an approximate indication of cloud cover. Its data are shown on the `GBTSTATUS` screen near the wind information. Values more negative than $-70\,\mathrm{watts/m^2}$ indicate clear skies while values more positive than $-15\,\mathrm{watts/m^2}$ indicate thorough cloud cover.

Wind and solar illumination affect the telescope structure and therefore influence 90 GHz observations with the GBT as well. Refer to Chapter 10 for discussion of how weather affects the GBT. The effect of wind is somewhat less on MUSTANG observations than for traditional targetted, single-beam, photometric observations: since the sky is densely sampled and all observations are conducted "on-the-fly", one needs only to know where the beams were pointed at a given time rather than to keep the telescope pointed accurately at a given spot. A "quadrant detector" on the GBT helps to increase the accuracy of this reconstruction on the GBT and the data from it is automatically used by analysis algorithms. *A good rule is to only use data from scans with mean winds under 10 mph and peak winds under 12 mph.* There are three GB weather stations, and to be conservative, use the maximum of the readings of the operable weather stations at a given time. The data reduction tools described in § 16.4 provide this information on a scan-by-scan basis during or after the observations. Conditions can also be monitored in the GBTSTATUS tab of astrid and in various CLEO screens (principally the weather and status screens).

The Dynamic Scheduling System (DSS) will schedule MUSTANG proposals when the forecasted winds are $\leq 10mph$, opacities are reasonable ($\leq\sim 0.25$) and cloud cover is sufficiently low.

16.1.2 Source Elevation

Several considerations constrain the range of elevations at which useful MUSTANG observations can be conducted: the atmospheric opacity, typically $0.07 < \tau_{90\,GHz} < 0.15$ per airmass in useful observing conditions, which attenuates the astronomical signal at low elevations; the MUSTANG cryogenics, which do not operate effectively below 19 degrees elevation; microphonics from other receivers, which couple more strongly at lower elevations; and the antenna primary surface, which shows a constant 90 GHz gain between 20 and 80 degrees elevation but can drop off sharply outside of this range. Putting these considerations together and summarizing:

- elevations below 20 degrees: not recommended due to reduced telescope gain and severely degraded MUSTANG cryogenic performance.
- 20 - 30 degrees elevation: usable performance for observations of bright sources. Receiver noise can be several times higher due to microphonics and the astronomical signal will be attenuated by a factor of 2 to 3 more than at zenith.
- 30 - 80 degrees elevation: good performance. Within this range, the optimal performance is seen between 50 and 65 degrees elevation.
- above 80 degrees elevation: not recommended. Photometric performance in this range is not consistent, probably due to degradation of the primary surface model. Above 85 degrees the antenna often fails to execute MUSTANG scan trajectories due to the rapid slew rates in azimuth that are required for an Alt-Az telescope such as the GBT to track close to the zenith.

16.1.3 Receiver Cryogenic State

The receiver should be cold and stable before observations begin. Whether or not this is the case can be determined by inspecting the MUSTANG CLEO screen (see Section 4.2). The detector array ("array G0") should be under 400 milliKelvin and shouldn't be changing by more than a milliKelvin or two per reading (there is one reading every few minutes); the series array should be under 5 Kelvin. The bolometer array is kept cold by a closed-cycle helium fridge, consisting of two separate, closed "pots": one with liquid He3 and one with liquid He4. The liquid in these pots boils away, thereby cooling the detectors, and is captured internally; when it has all boiled away it must be re-condensed or "cycled". The MUSTANG CLEO screens present conservative estimates of how much time remains for the He3 and He4 fridges. Ideally there will be enough of both to cover your entire observing run. Only He3 is required to operate MUSTANG, although if He4 runs out in the middle of a run the detectors will warm up slightly and need to be re-biased (§ 16.2).

Observatory staff are responsible for delivering appropriate cryogenic conditions but due to external factors it will occasionally not be possible at the start of a run. If this has occured the operator or support scientist will inform you of the situation and recommend a course of action. Time lost to cryogenic failures, like other lost time from hardware failures, will not be charged to the balance of your project time.

16.2 Preparing for, and Cleaning up after, Observations

Your GBT friend or support staff will have MUSTANG tuned up and ready to go at the start of your run. These settings should have been saved to

`/home/gbt/etc/config/mustang_config/current.conf`

From a terminal(s) on a 32-bit machine on the gbt network:

- As for other GBT observations, start `ASTRID` and `CLEO`. Within `CLEO`, select `Launch` → `Receivers` → `MUSTANG`.The housekeeping tab is most useful to have up during observations.
- Start the MUSTANG data monitor. To do this first

  `source /home/gbt/gbt.bash`

  (or source gbt.csh if using C-shell). Then type

  `mustangdm`
- Run the `mustanginit` scheduling block (SB). This scheduling block sets up ASTRID and performs basic initialization of MUSTANG.
- Run the `findbestbias` and `tweaktargets` SBs, in that order. These SB's find and set usable detector biases and optimize the SQUID tuning; they will require about 5 minutes to execute.
- Collect a cal file to check that everything is in working order using the `calandblank` SB. This will collect two scans, one with the cal diode flashing on and off to check for optical responsivity, and one file with the cal off the check detector noise.
- Fire up the MUSTANGIDL GUI and check the cal data as described in § 16.4
- Proceed with observing, as described in § 16.3.

As the last scheduling block (SB) in your program please run `mustangshutdown` to leave the receiver in a quiescent state.

16.3 Observing with MUSTANG

16.3.1 Mapping Strategies

All MUSTANG data are collected with variants of an "on-the-fly" (OTF) mapping strategy in which the antenna is slewed to cover a given region of interest while data are recorded. Two main scan

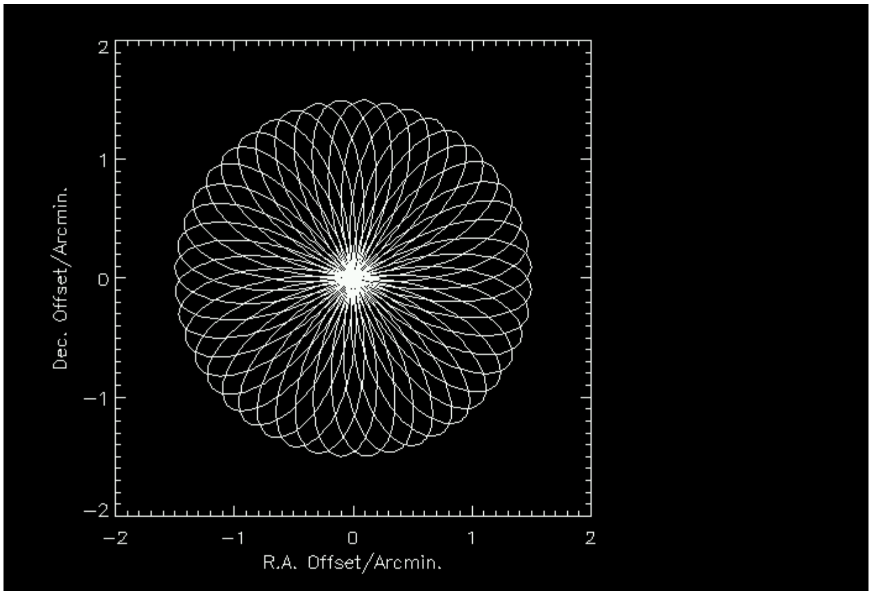


Figure 16.1: Full (22-cycle) daisy scan trajectory with a radius $r = 1'.5$.

strategies have been developed: a "box" or "billiard-ball" scan which covers a rectangular region with approximately uniform coverage; and a "daisy" or "spirograph" scan pattern which covers a circular area with a more center-weighted distribution of integration time on the sky. Under normal circumstances we do not recommend the standard ASTRID observing procedures RALONGMAP or DECLATMAP which perform discrete, linear raster-scans over a given region because these tend to excite vibrations of the GBT feedarm which in turn give rise to unacceptable pointing errors at 90 GHz. Due to the desirability of covering a given point of interest on the sky with many detectors, and due to the effects of fluctuations in detector gain and sky noise, we do not recommend staring at a single point either, even for point-source photometry projects. Under these circumstances the center-weighted coverage of the daisy scan is appropriate. For the identical total integration times and areas covered the daisy pattern gives a factor of ~ 3.6 more integration time on the central field-of-view, or a reduction of ~ 1.9 in RMS noise compared to uniformly distributed integration.

The daisy scan pattern is supported by ASTRID directly and can be invoked as follows:

```
Daisy('1331+3030',daisyRadius,daisyRadialPeriod,0,0,scanDuration, \
        beamName='C')
```

where

- daisyRadius is the radius of the circular area to be covered, in arcminutes
- daisyRadialPeriod is the period of radial oscillations, in seconds (not to be less than 15 sec $\times\sqrt{daisyRadius/1.5\,arcmin}$ for radii $> 1.5'$ and in no case under 15 seconds).
- scanDuration is the scan duration in seconds. Approximately 22 radial periods are required to completely cover a circular area with the default parameters. (Also see notes in section 16.6.7.1).
- beamName must be 'C' for MUSTANG since individual beam offsets are not defined in the antenna database.

An example trajectory is shown in Figure 16.1. For photometry of point sources you want the array at some point to be completely off source ($r \sim 0'.8$). Typical radii are $1'$ to $5'$.

To cover rectangular areas more uniformly there is a custom "Box" scan procedure, invoked as follows:

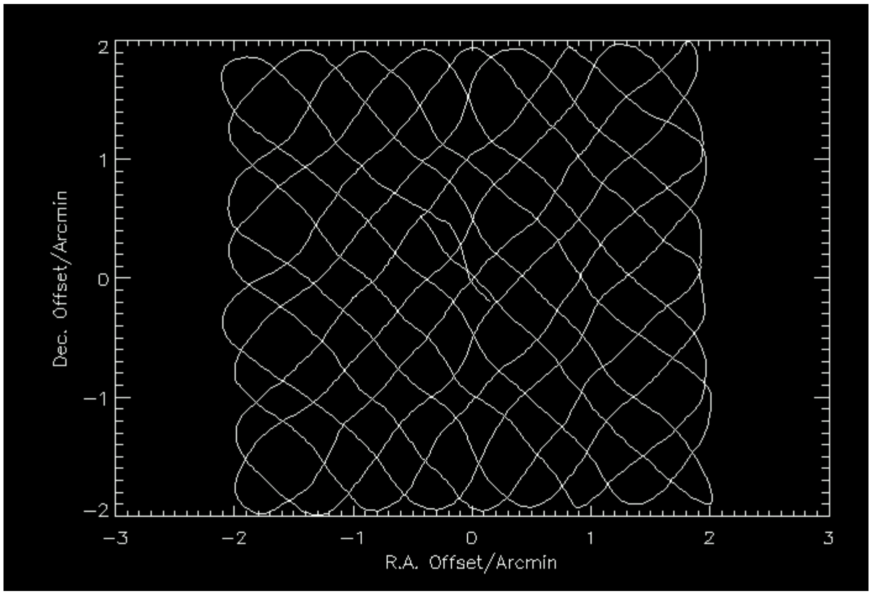


Figure 16.2: 4′ box scan trajectory.

```
DefineScan("boxtraj", "/users/bmason/gbt-dev/scanning/ptcsTraj/boxtraj.py")
boxtraj(mySrc, x0=x0, y0=y0, taux=taux, tauy=tauy, scanDuration=scandur, \
          dx=dx, dy=dy)
```

where

- x0 and y0 specify the box width and height in arcmin
- taux and tauy specify the periods of motion in either direction, in seconds
- scanDuration specifies the scan duration in seconds
- dx and dy specify dither offsets for the trajectory in arcminutes. In practice a 2/3 beam (6″) triangular dither serves well.

`boxtraj` executes a truncated sawtooth waveform in each direction (RA and Dec, typically) with specificable periods taux and tauy. Combinations of (taux,tauy,x0,y0,scanDuration) which give approximately uniform coverage have been determined by experimenting with simulations. Some common ones which also comply with GBT motion limitations are:

- $2' \times 2'$ to $3' \times 3'$, square patterns, are well covered by taux=10 sec, tauy=8 sec, and a 160 second total scan duration.
- $5' \times 5'$ to $7' \times 7'$ square patterns are well covered by taux=9, tauy=8, and a total scan duration of 290 seconds.

An example box scan trajectory is shown in Figure 16.2. If the area to be covered is substantially larger than any of these regions and not suitable to be covered by tiling, your staff friend can help find suitable parameters.

16.3.2 Sensitivity

When mapping a $3' \times 3'$ region uniformly (i.e., with a box-scan approach) MUSTANG typically achieves a map noise of 0.4 mJy/beam (RMS) in one hour of integration time. This RMS descreases as $\sqrt{t}$ for many hours under good conditions, and (for a fixed integration time) increases as the area covered for larger areas. Gains cannot be made by covering smaller areas uniformly due to practical limitations associated with the telescope motion and detector noise characteristics. However, for photometry of point or compact sources, the daisy scan will deliver a factor of 1.9 improvement in RMS for a fixed integration time (0.2 mJy/beam RMS in one hour integration, for the central $d = 1'$ area covered).

16.3.3 Establishing & Monitoring Good 3mm Performance of the Antenna

It is imperative to establish good 90 GHz performance of the GBT at the start of your observing run and monitor it carefully throughout. This entails determining any corrections to the GBT subreflector focus position that are needed, and determining any corrections for thermal deformation of the GBT surface that are needed. To do this we use the technique of Out-of-Focus (or phase retrieval) Holography, also known as OOF. The phase retrieval technique uses data from a series (typically three) of beammaps collected at different focus settings to reconstruct low-order phase errors at the dish surface, which can then be corrected with the GBT's active surface. The same data are used to derive a consistent set of corrections to telescope pointing and focus, which are also applied online via the telescope control system. The `ASTRID AutoOOF()` procedure will acquire and analyze these data and give you the option to send them to the telescope. Acquiring the OOF data should take about seven minutes; analyzing them will take another five to seven. When complete, send the surface correction to the telescope from the AUTOOOF tab and collect a single in focus scan to verify that the beam shape is still reasonable (not larger than before!) and that the source amplitude is equal to or greater than what was seen before the surface correction was sent. If the data pass these sanity checks, you are ready to proceed to your science observing. Otherwise, repeat the focus and/or primary surface measurement. The `AUTOOOF()` procedure also determines and applies subreflector focus and pointing corrections.

Note: once the processing is complete, it is the *first* "Send Solutions" button (marked "new, recommended method") that you want to push to send the surace, pointing, and focus corrections to the telescope. If you push the second ("old, original") button the focus offset will not be sent, which may result in irretrievable degradation of the quality of your data.

You can also manually check the focus with a more fully sampled series of maps (typically five) collected at a range of focus positions. The `parFocusDaisies` example scheduling block implements such a measurement, which can be analyzed as described in § 16.4. We recommend you only take this approach if you have reason to doubt the corrections derived by `AutoOOF` – for instance, the peak source intensity obtained after applying the focus and pointing solutions from `AutoOOF` is lower than the peak source intensity before. **Note:** `parFocusDaisies` centers the focus scans on the focus correction value (LFCY) set by the variable **nomfocus**, and leaves it at this value at the end! Do not forget to set nomfocus to the current local focus Y-correction (LFCY) before running it. After running `parFocusDaisies` and analyzing the data, you can send the pointing (in arc*minutes*) and focus corrections (in millimeters) either by telling them to the operator or by editing and running the example `applyptg` SB.

Every half hour you should check the focus and beam shape by obtaining a single quick map (SB `quickdaisy`) of a calibrator source, monitoring the beam size, pointing, and relative source amplitude over time. At this time, another `calandblank` should also be done. If the beam size increases by more than 10%, or the source amplitude decreases systematically by 15% or more, then it is likely that the thermal corrections to the GBT primary and/or the subreflector focus corrections need updating, and it is time to do another AUTOOOF. (note: beam size and source amplitude degradation may also be caused by poor weather). While the AUTOOOF procedure will update the telescope pointing corrections

Name	R.A. (J2000)	Dec. (J2000)	Notes
Uranus	0h	0d	In ASTRID
Neptune	22h	-12d	In ASTRID
Mars	13h-24h	-10d	In ASTRID; extended
Saturn	12h	0d	In ASTRID; extended
Ceres	17h - 22h	-20d	Use custom ephemeris
W3(OH)	02 : 27 : 03.8	+61 : 52 : 24.8	extended
Mwc349	20 : 32 : 45.6	+40 : 39 : 37	-
CRL2688	21 : 02 : 18.8	+36 : 41 : 37.8	slightly extended

Table 16.1: A list of secondary flux calibrators suitable for use with MUSTANG. Coordinate ranges are for the 2010/2011 observing season.

as well, it is not necessary to perform an AUTOOOF just to correct pointing which has drifted assuming the beam shape and source amplitude are stable. The pointing offsets can be corrected after the fact in the data analysis using your periodic calibrator monitoring observations.

16.3.4 Calibration

It is essential to observe a flux calibrator during each observing session, and to do this only after the initial set of active surface (OOF) corrections have been applied. Changes in gain due to subsequent OOF corrections if any can be tracked with the secondary calibrator. The recommended MUSTANG flux density scale is ultimately based on the WMAP 7-year (Wieland et al. 2010) measurement of Mars, however, Mars is often not visible and is resolved by the GBT at 90 GHz, complicating its use. A list of secondary flux calibrators sufficiently compact and stable (or have modelable light curves) is shown in Table 16.1. Other sources may also be suitable given appropriate bootstrapping observations. Custom ephemeris can be obtained from the JPL online `HORIZONS` system and translated into the `ASTRID` format. Work is under way to increase the number of secondary (bootstrapped from planets) flux calibrators.

A catalog of 27 compact, 90 GHz bright sources is at `/users/bmason/mustangPub/mustangpnt.cat`. These sources are suitable for peak/focus/OOF check observations and for monitoring the antenna performance, but are *not* suitable for flux calibration.

16.3.5 Observing Summary: Example Observing Sequence

An example observing sequence would be as follows:

- mustanginit
- findbestbias & tweaktargets
- calandblank
- on a 1 Jy or brighter point source–
  - quickdaisy: check the gain and beam size on a calibrator
  - autooof: establish GBT surface corrections
  - quickdaisy: check gain and beam size after applying surface corrections
- quick daisy on primary flux calibrator. if this is a large slew (over 60 degrees in az, over 30 degrees in el) you might want to run focus daisies centered on the current LFCY just to be safe.

- quick daisy on secondary flux calibrator within 15 degrees of science target. if this is a large slew you might want to run focus daisies centered on the current LFCY just to be safe.
- half hour of observing (boxmap or parfulldaisy)
- quickdaisy on nearby secondary (pointing/focus) calibrator. If the beam gain has gone down by 15% or more, or the beam become 10% fatter in one or both directions, repeat an AutoOOF on the pointing calibrator, and verify results with a quickdaisy.
- half hour of observing (boxmap or parfulldaisy)
- quickdaisy to check gain and focus
- *et cetera.*
- mustangshutdown

16.4 Quick Look Data Reduction

It is essential to monitor the quality of your data as you are collecting it. A suite of tools has been written in IDL to make this possible. This section outlines its use. **We highly recommend that you reduce your imaging data as you go along so that you can catch problems should they develop**.

For monitoring ongoing observations, observers should use the GUI interface to the MUSTANG IDL code. This can be started from the UNIX command line on any Green Bank UNIX machine as follows:

```
[username@prospero]~bmason/mustangPub/mustangidlgui
```

The first step is to select the project you are working with via the "Browse Projects" button; selecting "Online" will select the most recently updated project in /home/gbtdata (probably your project, if you are observing on the GBT). You can also, for inspection of already-acquired data, type in the full path to the data directory in the box (for instance, /home/gbtdata/AGBT11C_033/ or /home/archive/science-data/tape-030/AGBT07B_012/ – note the trailing slash). Once this is done the area at the bottom of the GUI will display a summary of your telescope period so far. This summary can be updated with the "Update Scan Summary". At this point IDL will read all files in the project directory so far and generate a summary (in the GUI window, and also in the terminal window the IDL GUI was launched from):

```
IDL> summarizeparproj
#                                        Scan#/
# Scan          File name       ScanType  out of        SB        SrcName
   1  2009_02_21_07:35:25.fits    Track    1 / 1   calandblank      cal
   2  2009_02_21_07:35:49.fits    Track    1 / 1   calandblank     Blank    ...
   3  2009_02_21_07:52:30.fits  RALongMap  1 / 1      focus       1337-1257
   4  2009_02_21_07:53:28.fits  RALongMap  1 / 1      focus       1337-1257
```

The GUI will also now have a list of valid "CAL" scans in the drop down CAL SCAN box (see below).

There are some latencies involved in getting all the FITS files written to disk so the current scan may not be accurately reflected in the summary. For this reason you should also not try to make a map with the current scan – wait about 1 minute after the completion of a scan to try and map it. We tried to make the IDL code as robust as possible but this will sometimes crash the code. In this event restart the IDL gui, re-enter the proejct (or click online) and re-select the cal scan you dsire.

The relative pixel gains are calibrated by flashing an internal cal lamp on and off, done by the `calandblank` SB. The data from a CAL scan are analyzed by selecting the desired scan in the drop down box in the GUI (see Figure 16.3). Partial example output is as follows:

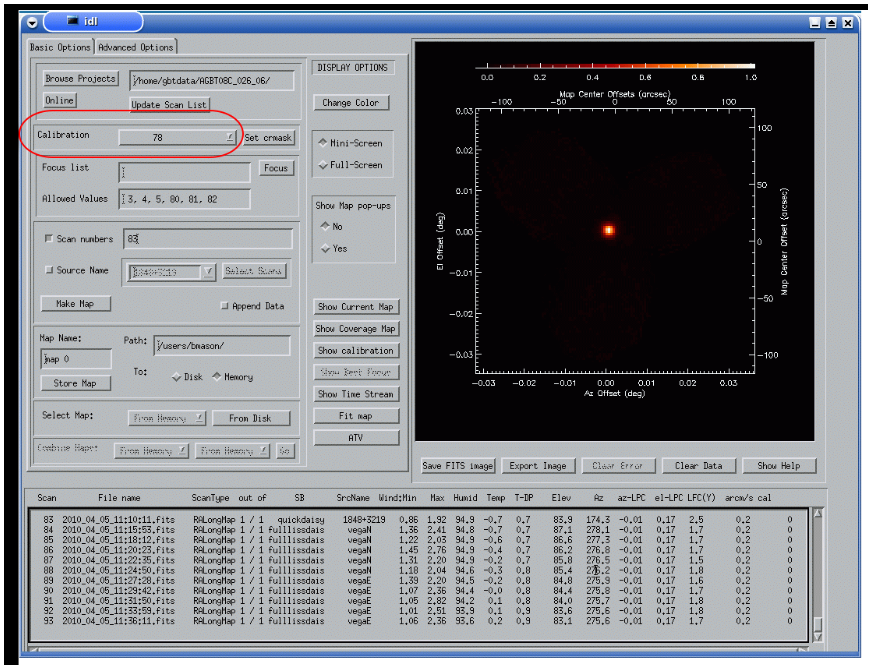


Figure 16.3: Selecting the CAL scan in the MUSTANG IDL GUI.

```
docalib ,calscan=1,gain=mygain ,crmask=mycrmask
.
.
.
 *** GAINS:
 -1.18084e-12    0.0688694      155.829      200.706
      0.00000     -93.2362      83.6297      148.163
 -1.18084e-12     0.320343      101.007      298.489
      0.00000     -77.4885      141.224      184.050    .....
      0.00000     -81.9272      79.5458      107.280
      0.00000     -82.3489      92.1591   -0.0603582
      0.00000     -57.1215      73.7274      148.184
      0.00000     -59.1193      73.8523      313.731
      0.00000  0.000713527   -0.0108514   -0.0494373
 *** CRMASK and total:       49.0000
      0.00000      0.00000      1.00000      1.00000
      0.00000      1.00000      1.00000      1.00000
      0.00000      0.00000      1.00000      1.00000
      0.00000      1.00000      1.00000      1.00000
      0.00000      1.00000      1.00000      1.00000      .....
      0.00000      1.00000      1.00000      0.00000
      0.00000      1.00000      1.00000      1.00000
      0.00000      1.00000      1.00000      1.00000
      0.00000      0.00000      0.00000      0.00000
```

The top shows the gain (in nominal counts per Jansky) and the bottom shows the "column-row" mask denoting optically responsive detectors with "1" and optically non-responsive detectors (or those flagged by automated criteria) by "0". The pixel gains serve the purpose of flat-fielding the detectors but the flux density scale is fiducial only and celestial calibration is still essential. The routine summarizes its findings in the above form, organized by logical row and column (*i.e.*, electrical labels for each detector

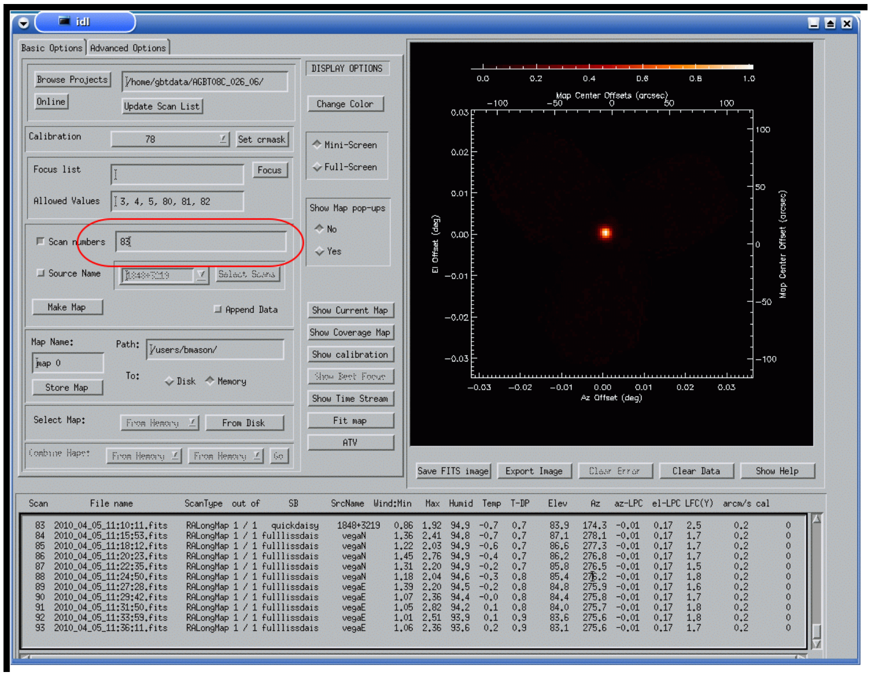


Figure 16.4: Specifying scans to image in the MUSTANG IDL GUI.

rather than array-face geometry). In this case all of column zero is non-responsive and only four columns are shown due to space limitations. Row 8 (zero indexed) is always non-responsive since this is by design each column's dark SQUID, not connected to a bolometer.

Given a nominal detector calibration, mapping data can be made into images by entering the desired scans in the box labelled "Scan Numbers" and clicking "Make Map" (see Figure 16.4). This uses the default imaging parameters, which should be suitable for most situations but which can be changed in the "Advanced" tab. The coverage map can also be displayed by clicking the "Show Coverage Map" button; this results in an image of weight per pixel (units of inverse Janskys squared) and is approximately proprtional to integration time, and inversely proportional to map noise variance. The map can be saved to a FITS file by clicking on "Save FITS Image". These files can be manipulated by standard astronomy imaging packages (*e.g.*, `ds9`, `fv`).

For monitoring the amplitude and beam-width of the twice-hourly pointing calibrator observations, the GUI provides the facility to fit each map to a Gaussian. To reduce the calibrator observations you can use the default (Right Ascencion/Declination) coordinate system but it is often more useful to make the map in Elevation/Cross-Elevation coordinates (i.e., offset relative to the source nominal "True" position) in order to monitor the telescope pointing offset. This can be done by selecting "EL/XEL" coordinates in the "Advanced" tab (see Figure 16.5). A given map can be fit to a Gaussian with the "Fit Map" button; the amplitude and beam width (FWHM) are shown in the terminal window and the GUI text output window. Since an elliptical Gaussian is used, there are two parameters for the width.

The imaging routines produce diagnostic feedback primarily intended for support staff. Many IDL routines also produce a floating point exception which can be safely ignored.

In the unlikely event that the beam degrades, `AutoOOF` does not fix it, and you have acquired `parFocusDaisies` data to check it, you may enter the scan numbers of the focus maps in the "Focus list" box. Each scan will be imaged and fit to a Gaussian, and the beam width and amplitude as a function of focus offset (LFC) will be plotted. Choose the LFC that minimizes the beamwidth and maximizes the source response and enter this into the Antenna Manager CLEO screen, or have the operator do so. The amplitude does not always peak at the same focus position that minimizes the beam shape, but it should be within a few millimeters. Choose a focus position that for your purposes

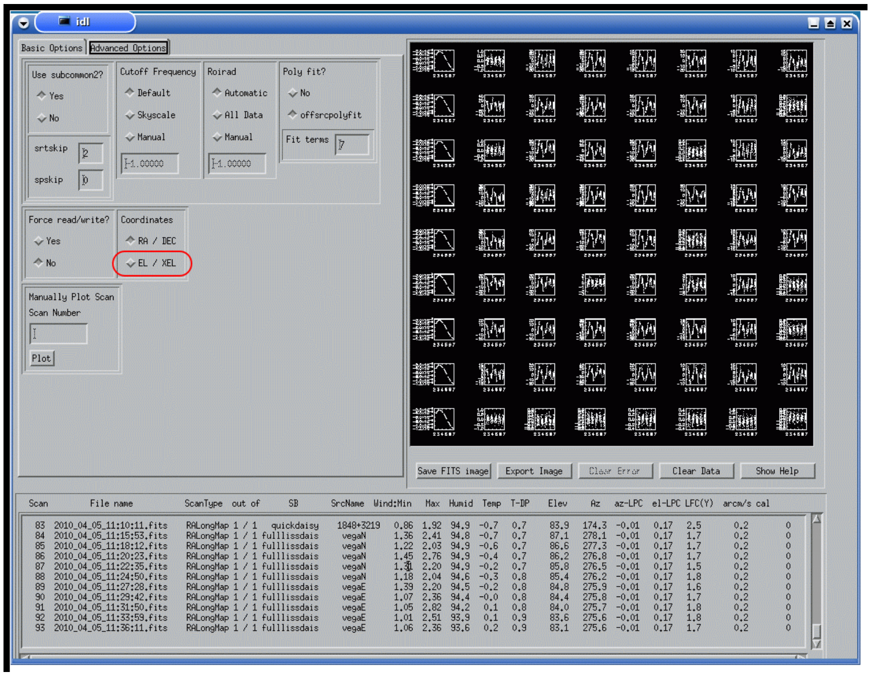


Figure 16.5: Specifying the coordinate system for the maps in MUSTANG IDL GUI.

is a good compromise between these two. In the lower (beam width or FWHM) plot, the two widths are represented by a dot and a diamond. Therefore a circular beam, which is a good indication of being in focus, will correspond to the dot and the diamond coinciding. This is illustrated in Figure 16.6.

When processing focus data, note that it can often take up to a minute for the most recently completed scan to become readable on the filesystem, so exercise some patience when processing your data on-line.

You can see the current LFCY in the ASTRID “GbgStatus” tab in the entry labeled “LFC (XYZ mm):” — it is the second number of the three.

16.5 Troubleshooting

GBT 90 GHz observing requires diligence on the part of the observer. Following are some problems that can come up, their symptoms, and first actions to take in response:

- *GBT out of focus*: the beam gets larger and the peak gain lower than it had been before. Check and reset the telescope focus.
- *GBT primary deforms* due to changing structure temperatures: after optimizing the focus the in-focus beam is still larger or lower-gain than before. Rerun `AutoOOF`.
- *Detectors reach the edge of their dynamic range*: DAC values in mustang monitor reach 0 or 16384. run a `calandblank`, which relocks the detectors mid-scale, or relock manually in the MUSTANG `CLEO` screen (column control tab).
- A *greatly diminished cal or source response, or greatly diminished number of live detectors*, can occur if the He4 runs out and the detectors warm up a little. With Helium-3 and Helium-4 both the Array G0 (detector array) temperature should read about 300 mK when detector biases have been applied for a few minutes; with only Helium-3 it will read about 340 mK with biases.

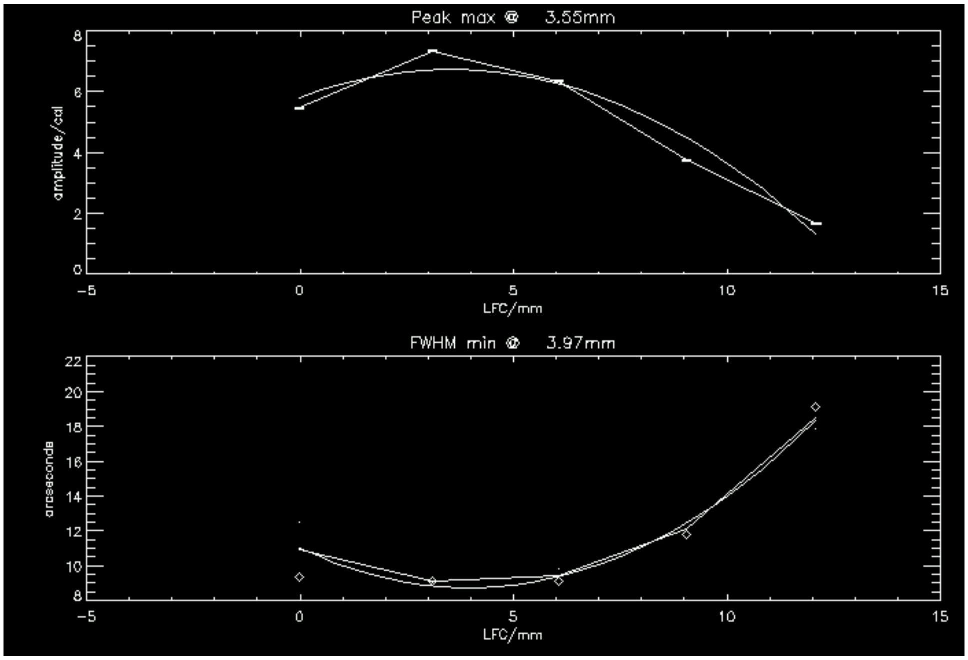


Figure 16.6: Summary plot produced by `best_focus` showing an optimum value for `LFC_Y` of about 3.7 mm.

Recondensing the He4 requires 1.5 hours and puts the instrument out of use for that time; it is possible to observe without, provided new biases are obtained (support staff can help with this). If the He3 has run out it must be recondensed.

- The MUSTANG IDL pipeline has an automatic binary cache mechanism which greatly speeds up processing of a given scan after the first time it has been read. Unfortunately if the first reading of a scan was messed up for some transient reason the binary cached version can get messed up. If the data are strange or repeatedly unprocessable, try including the `forceread` or `focewrite` options in analysis routines such as `multimakemap` or `best_focus`.
- *Data Latencies Crash IDL*: as previously noted, it sometimes takes 10s of seconds for the data files to be complete and visible on disk to IDL. Trying to read the latest scan too soon can result in IDL crashing. In this case, restart the IDL gui, reselect your project (or click "online" if you are observing, which will automatically select the most recent telescope period with data); reselect the cal scan; and proceed.

16.6 Example ASTRID Scripts

The following sub-sections present some template ASTRID scheduling blocks (SB's). Templates are also kept in ASCII format in `/users/bmason/mustangPub/sb/`. **Note: the SBs in this document are for the sake of example only; use the template SBs in the above UNIX file path to get the latest version**.

16.6.1 mustanginit

```
Configure("/users/bmason/mustangPub/sb/mustangfull.conf")
```

16.6.2 autooof

```
mySrc=''1159+2914'
Catalog()
Slew(mySrc)
AutoOOF(source=mySrc)
```

16.6.3 calandblank

This SB runs a 15 second scan with the cal diode firing on and off, allowing the bolometer responsivities to be measured, and a 30 second scan tracking blank sky, allowing a check on the detector noise.

```
# duration of cal and blank scans
#  in seconds
calduration=15
blankduration=30

# uncomment these 3 lines
#  to do the calibration at a given
#  az/el
# myAz=260
# myEl=78
# myLoc=Location("Encoder",myAz,myEl)
# or use this one to use the current az/el
myLoc=GetCurrentLocation("Encoder")

# do not modify from here down
#  extra information (source names, caltags etc)
#  is added for the sake of data analysis software,
#  which depends on these tags.
############

Slew(myLoc)
Configure("/users/bmason/mustangPub/sb/mustang.conf")
execfile("/users/bmason/mustangPub/ygor/relockAstrid.py")
Configure("/users/bmason/mustangPub/sb/calon.conf")
Annotation("CALTAG","DIODE")
SetValues("ScanCoordinator",{"source":"cal"})
```

```
Track(myLoc,None,calduration)
Configure("""
mustang.init='cal'
swmode='tp_nocal'
tint=0.001
""")
Annotation("CALTAG","BLANK")
SetValues("ScanCoordinator",{"source":"Blank"})
Track(myLoc,None,blankduration)
Annotation("CALTAG")
SetValues("ScanCoordinator",{"source":"nothing"})
Configure("/users/bmason/mustangPub/sb/caloff.conf")
```

16.6.4 parFocusDaisies

This SB runs a sequence of five maps at a range of subreflector focus settings about the nominal focus. *The nominal focus is in the script as the variable* `nomFocus`*; you should set it to the current best-determined focus setting, and remember that the script will leave it at that value upon completion.* Each map takes 45 seconds; the total SB should run in about 5 minutes.

```

# some good sources are...
# 1642+3948; 2253+1608; 0927+3902, 0319+4130
#  0359+5057. 1955+5131, 1256-0547
# 0854+2006
mySrc="1415+1320"

####################################
# CATALOGS
# none needed for planets

Catalog()

Configure("/users/bmason/mustangPub/sb/mustang.conf")

####################################
# start, stop, and increment for focus,
#  in mm
#
# current, nominal LFC
nomFocus=0
# LFC deltas to try around this
dfocus=(-10,-3,0,3,10)
# focus is left at nomFocus

####################################
#  Daisy Params
#
daisyScanDur=45.0
daisyRad=1.5
daisyRadPd=15.0

############################################################
```

```
##########################################################
# Do not modify below here
#


from time import sleep

Slew(mySrc)
Configure("/users/bmason/mustangPub/sb/mustang.conf")

# relock MUX
execfile("/users/bmason/mustangPub/ygor/relockAstrid.py")

for df in dfocus:
    # set focus
    ff=df+nomFocus
    SetValues("Antenna",{'local_focus_correction,Y':ff})
    SetValues("Antenna",{"state":"prepare"})
    sleep(3)
    Daisy(mySrc,daisyRad,daisyRadPd,0,0,daisyScanDur,\
          beamName='C',cos_v=True,coordMode="Encoder",calc_dt=0.2)

#
# leave in nominal state
SetValues("Antenna",{'local_focus_correction,Y':nomFocus})
SetValues("Antenna",{"state":"prepare"})
```

16.6.5 applyptg

This SB applies pointing and focus offsets to the telescope. The desired focus offset in millimeters is set in the python variable `focusoff`. Typically it is not necessary to do this by hand any more, as the `autooof` procedure takes care of the pointing and focus offsets in addition to the surface.

```
myLoc=GetCurrentLocation("Encoder")
# offsetts to apply in arcmin - LEAVE AT ZERO
azoff=0
eloff= 0
# Y focus offset in mm
focusoff=-3

###############################

azoff=azoff/60.0*3.14159265/180.0
eloff=eloff/60.0*3.14159265/180.0
SetValues("Antenna",{'localPointingOffsets,azOffset2':azoff})
SetValues("Antenna",{'localPointingOffsets,elOffset':eloff})
SetValues("Antenna",{'local_focus_correction,Y':focusoff})
```

16.6.6 quickdaisy

`quickdaisy` collects a single quick map using the daisy scan pattern. This is useful, for instance, to check your calibration every half hour. It's not a bad idea to do two of these back to back, since they each take less than a minute, which will also give you a check of the photometric accuracy of your measurements every half hour.

```
# some good sources are...
# 1642+3948; 2253+1608; 0927+3902, 0319+4130, 0854+2006
mySrc= "Ceres"

Catalog("/users/bmason/gbt-obs/par/ceres.ephem")

####################################
# CATALOGS
# none needed for planets
#
# standard pointing source catalog :
#       /home/astro-util/pointing/pcals4.0/pointing.cat
Catalog()
# or user defined catalog
# Catalog("/users/rfeynman/gbt-obs/nobel.cat")
Configure("/users/bmason/mustangPub/sb/mustang.conf")

####################################
#  Daisy Params
# nominal: 1.5', 30sec, 0, 120sec
# better coverage: 0.8', 30 sec, 100 sec
daisyScanDur=75
daisyRad=1.5
daisyRadPd=15.0

#
# Coord Sys in which to execute trajectory
#
# eg-- J2000, Encoder
coordSys="J2000"


############################################################
############################################################
# Do not modify below here
#

Slew(mySrc)
# relock detectors
execfile("/users/bmason/mustangPub/ygor/relockAstrid.py")

Daisy(mySrc,daisyRad,daisyRadPd,0,0,daisyScanDur,\
       beamName='C',cos_v=True,coordMode=coordSys,calc_dt=0.2)
```

16.6.7 parfulldaisy

`parfulldaisy` does a sequence of five daisy scans phased appropriately, relative to each other, to provide good coverage of a circular region (in this case, with a radius of $r = 2'.8$). The full sequence of 5 will run in about 10 minutes with the chosen radial period of 25 seconds.

```
#
# parfulldaisy
#  script to execute 22 radial cycles, chopped into 5 scans,
#  of a daisy scan which gives full coverage
#

mySrc="1256-0547"

####################################
# CATALOGS
# none needed for planets
#
# standard pointing source catalog :
#      /home/astro-util/pointing/pcals4.0/pointing.cat
Catalog()
Configure("/users/bmason/mustangPub/sb/mustang.conf")

####################################
#  Daisy Params
#
#
#daisyScanDur=100
daisyRad=2.8
daisyRadPd=25

#
# Coord Sys in which to execute trajectory
#
# eg-- J2000, Encoder
coordSys="J2000"

############################################################
############################################################
# don't change below here
#

# embedded in Daisy() astrid routine--
periodRat=3.14159

Slew(mySrc)

################################################
# Derived Parameters
# chop the full daisy (22 cycles) into 5
#  individual scans
daisyScanDur=daisyRadPd*22.0/5.0
nradosc=daisyScanDur/daisyRadPd
rotphasestep=2*3.14159265*nradosc/periodRat
```

```
myphases=[0,rotphasestep,rotphasestep*2,rotphasestep*3,rotphasestep*4]

for myphase in myphases:
        Daisy(mySrc,daisyRad,daisyRadPd,0,myphase,daisyScanDur,\
              beamName='C',cos_v=True,coordMode=coordSys,\
              calc_dt=0.2)
```

16.6.7.1 Notes on Daisy Scans

The way the daisy scan is set up it takes 22 radial periods to (more or less) "complete" one full daisy. The radial periods are typically in the range of 15-30 seconds, depending on the radius being used, so 22*20 sec = 440 sec – a fairly long time. Such a long trajectory being sent to the antenna manager, due to intrinsic inefficiencies in GRAIL's array handling mechanism, really slows things down: the overhead at scan start can easily exceed 1 minute. Therefore one should typically keep individual scans (which have nontrivial trajectories) to 5 minutes or less. parfulldaisy is a SB that does 22 radial periods, broken up into 5 individual invocations of the Daisy() procedure. There is an optional "phase" argument to the daisy that lets you do this, so that scan 2 starts where scan 1 left off. See above example.

16.6.8 boxmap

`boxmap` covers a rectangular region with a sawtooth scan pattern in each direction. To limit the accelerations at the turnaround points only the first few terms in the fourier series are retained. This SB executes the map three times, each requiring five minutes, with a triangle-patterened 6″ dither between the three helping to smooth out small variations in the coverage.

The coordinate system choice "Encoder" here denotes the coordinate system used for the trajectory offset, not for the central point inthe map, which is defined by the source catalog. Therefore this SB as written will map a rectangular region around a given point on the sky, with the rectangle rotating with parallactic angle.

```
mySrc= "rxj1347"

###################################
# CATALOGS
# none needed for planets
#
# standard pointing source catalog :
#    /home/astro-util/pointing/pcals4.0/pointing.cat
Catalog()

###################################
# Box Trajectory Parameters
#  box width and height, in arcmin
x0=5.0
y0=5.0
# number of times to repeat the map
nrepeat=3
# dithering offsets
#  THE LENGTH OF THIS ARRAY MUST
#  BE THE SAME AS NREPEAT!! (3)
#  to not dither set them all to zero.
```

```
dx=[0.0,-0.07,0.07]
dy=[0.1,-0.07,-0.07]
# period for h and v oscillation, sec
taux=9
tauy=8
# duration, sec
scandur=290
#
# this SB will therefore run for 290 x nrepeat seconds
#  plus overhead
#
# phase of scan in seconds (default 0)
tstart=0
# NOTES
#  3'x3' or 2'x2' covered well by taux=10sec,tauy=8sec, 160 sec scan
#  5'x5' or 7'x7' covered well by taux=9,tauy=8, 290sec
#
##################################

# Coord Sys in which to execute trajectory
#
# eg-- J2000, Encoder
coordSys="Encoder"

# do not change from here down
#####################################################
#####################################################

Slew(mySrc)
Configure("/users/bmason/mustangPub/sb/mustang.conf")

DefineScan("boxtraj", "/users/bmason/gbt-dev/scanning/ptcsTraj/boxtraj.py")
for i in range(nrepeat):
    boxtraj(mySrc,dx=dx[i],dy=dy[i],phix=tstart,phiy=tstart,\
       x0=x0,y0=y0,taux=taux,tauy=tauy,scanDuration=scandur,\
       calc_dt=0.25,coordMode=coordSys)
```

16.6.9 mustangshutdown

```
SetValues("Rcvr_PAR",{'cryoCycleType':'Custom'})
SetValues("Rcvr_PAR",{'cryoAutoCycle':'He3'})
SetValues("Rcvr_PAR",{'cryoDAQPowerSafety':'On'})
SetValues("Rcvr_PAR",{'cryoDAQPower':'Off'})
SetValues("Rcvr_PAR",{'cryoTowerPower':'Off'})
SetValues("Rcvr_PAR",{'fireCal':'Off'})
SetValues("Rcvr_PAR",{'hlDetBias,column':'all'})
SetValues("Rcvr_PAR",{'hlDetBias,value':'0'})
SetValues("Rcvr_PAR",{'scanType':'Default'})
SetValues("Rcvr_PAR",{'state':'Prepare'})
```

Chapter 17

Zpectrometer

Place-holder for a revised chapter on the Zpectrometer.

Appendix A

The GBTSTATUS IF Path Nomenclature

The nomenclature used for the IF path information in the Astrid Status Tabs:

IF # The # displayed is the number corresponding to the IF Rack switch in use. The value displayed is the RF power in Volts detected by the IF Rack.

CM # The # displayed is the number corresponding to the Converter Module in use. The value displayed is the RF power in Volts coming out of the Converter Module after the LO2 and Third LO (LO3) mixers and before the Converter Module filters.

CF # The # displayed is the number corresponding to the Analog Filter in use. The value displayed is the RF power in Volts coming out of the Analog Filter Rack after all filters have been applied (used with 100MHz Converters.

SG # The # displayed is the number corresponding to the Analog Filter in use. The value displayed is the RF power in Volts coming out of the Analog Filter Rack after all filters have been applied (used with 1.6 GHz Samplers.

ACS-Port # The # displayed is the number corresponding to the port of the Spectrometer in use. The value displayed is the duty cycle in db [1]. This value is relative to the optimum power level; for best performance, it should be between -3 and +3 db.

SPP-Port # The # displayed is the number corresponding to the port of the Spectral Processor in use. The value displayed is the power level in db.

Radar-Port # The # displayed is the number corresponding to the port of the Radar in use.

DCR-Port # The # displayed is the bank and number corresponding to the port of the DCR in use. The value displayed is the total power in raw counts.

backendIF The value displayed is the frequency of the Doppler track rest frequency as seen by the backend, in GHz.

TSYS # The # displayed is the number corresponding DCR port in use. The value displayed is the system temperature as reported by the DCR (should be considered a loose approximation).

[1] x represented in db is given by $10 \log x$

Appendix B

Introduction to Spectral Windows

Several simultaneous frequency bands may be specified by the configuration keyword nwin (number of spectral windows, see § 5.2.1.4) and a list of rest frequencies and offsets (keywords restfreq, deltafreq). Each spectral window includes both polarizations. i.e., if you specify one window, you get two IFs routed to the back end device, one for each polarization; if you specify two windows, you get 4 IFs, and so forth.

The configuration software tries to put the midpoint of the total frequency range spanned by all windows at the center of the nominal IF1 band so as to use the narrowest I.F. bandpass filters that will pass the desired range of frequencies. In some uncommon cases this is not possible, so the IF bandwidth must be increased to pass the desired range of frequencies. For prime focus receivers, the total I.F. bandwidth is 240 MHz; for the Gregorian receivers, up to 4 GHz is possible, depending on the receiver.

The user specifies the rest frequencies (restfreq keyword) and a range of radial velocities (vlow and vhigh keywords). The various IF filters are set to include the required range of frequencies in the local frame required by the radial velocity range. The configuration software predicts the local frequency for each spectral window based on the rest frequencies and the radial velocity. During observing the tracking Local Oscillator (LO) will correctly track the velocity of spectral window number 1. Because there is only one tracking L.O., the other spectral windows are set up with frequency offsets in the local frame with respect to window number 1. When observing at a variety of high velocities, one should run a configuration for each change of velocity (i.e., do not rely on just changing the velocity in the LO1 manager), and one should set vlow=vhigh.

Note that the deltafreq keyword gives frequency offsets that are applied in the local (or topocentric) frame. For example, if V_{frame} is velocity of the reference frame, V is source velocity in that frame, ν_{rest} is the rest frequency of the line and we use the Radio definition of velocity then the topocentric frequency will be

$$\nu_{topo} = \nu_{rest} \left(1 - \frac{(V + V_{frame})}{c}\right) + deltafreq \tag{B.1}$$

Finally note that the expert user may specify any of the IF conversion frequencies and total IF bandwidth, overriding the calculations done by the configuration software ("ifbw", "if0freq", "lo1bfreq", "lo2freq", and "if3freq" keywords). This option may be needed in some peculiar cases. Of course one needs a good knowledge of the IF to make use of this option.

Appendix C

Usage of vlow and vhigh

The configuration keywords vlow and vhigh give the range of velocities of all sources to be observed. This information is used to set various filters in the system so as to admit the required range of frequencies. Setting the velocity for each specific source is done later in the observing block. For galactic sources where the range of velocities is rather small it is usually best to set both vlow and vhigh to zero.

When strong RFI is present is it best not to use vlow and vhigh. The use of vlow and vhigh can cause the GBT IF system to have a larger IF bandwidth than is necessary for a single source. This can let parts of the IF system be unnecessarily affected by RFI. The observers might need to reconfigure after each source if the change in velocity is larger than the bandwidth of a filter.

An example of how vlow and vhigh can be used is as follows. Suppose that you are looking for water masers in extragalactic AGN. Furthermore, lets say that you are looking at 100 candidates with velocities from 1000 km s^{-1} to 40000 km s^{-1}. The you would set vlow=1000.0 and vhigh=40000.0 and you would not have to change the IF configuration when you changed sources.

Note that if vdef="Red" (i.e., redshift), then you must give the redshift parameter "z" as the values for "vlow" and "vhigh" instead of velocity.

Your scientific contact person can help you decide if you should use vlow and vhigh.

Appendix D

Location and Offset Objects

A Location is used to represent a particular location on the sky. Locations can be specified in the following coordinate modes: "J2000", "B1950", "RaDecOfDate", "HaDec", "ApparentRaDec", "Galactic", "AzEl", and "Encoder." A Location is specified by two values, the meanings of which are dependent on the coordinate mode chosen. E.g. For J2000, the two values are time and degrees.

Here is an example of how to specify a Location:

```
location = Location("J2000", "16:30:00", "47:23:00")
```

An Offset object represents a displacement from a Location.

Here is an example of how to specify an Offset:

```
offset   = Offset("J2000", 0.1, 0.2, cosv=True)
```

which represents an offset of 0.1 degrees in RA and 0.2 degrees in DEC. "cosv=True" (the default) means the RA offset is divided by cosine Dec before applying the offset.

Two Offsets may be added together, but they must have the same coordinate type. For example:

```
off1   = Offset("J2000", 0.1, 0.2, cosv=True)
off2   = Offset("J2000", 1.0, 1.0)
totaloffset = off1 + off2

BUT:
off3   = Offset("B1950", 1.0, 1.0)
totaloffset = off1 + off3
-- result in an error!
```

In Astrid scripts, one may add an Offset to a Location, if they have the same coordinate types.

Here is an example of how to add an Offset to a Location:

```
myoffset = Offset("B1950", "00:00:30", "00:00:45")
mysrclocation = Location("B1950", "16:30:00", "47:23:00")
mynewposition = mysrclocation+myoffset
```

But note that addition is not commutative for Astrid!! The following produces a validation error in Astrid.

```
mynewposition = myoffset + mysrclocation
-- ERROR!
```

Appendix E

A Note on Angle formats and units in Astrid and Catalogs

There are two formats for angles in Observing Scripts and Catalogs:

sexegesimal: e.g., hh:mm:ss[.ss], dd:mm:ss[.ss]

decimal numbers, e.g, ddd.ddd

When the quantity is RA or HA, an angle given in sexegesimal is hours, minutes, seconds of time.

For all other angle quantities (e.g., dec, az, el, glon, glat) an angle given in sexegesimal is degrees, minutes, seconds of arc.

In "Location" and "Offset" objects, a quantity given as a decimal number is always understood as being in units of degrees of arc, regardless of the type of unit. However in Catalogs, a decimal number for RA or HA is assumed to be hours; for other angles (DEC, Az, El, Glon, Glat) it is degrees of arc.

For example, if one is specifying an Offset object as in the following Astrid directive:

```
OnOff( "3C286", Offset("J2000", "00:04:00", 0.5), 60, "1")
```

The offset will be 4 minutes of time in RA and 0.5 degrees of arc in DEC. (The coordinate mode "J2000" means the coordinate pair is (RA,DEC), hence sexegesimal RA is assumed to be in hours.)

Alternately if one says:

```
OnOff( "3C286", Offset("J2000", 1.0, 0.5), 60, "1")
```

The offset will be one degree of arc in RA and 0.5 degrees of arc in DEC.

Appendix F

Advanced Utility Functions

There are a few advanced utility functions that once can use in an Observing Script.

F.1 General Functions

F.1.1 GetValue()

The GetValue() function can be used to retrieve any parameter value within the Monitor and Control system. The syntax is

```
value = GetValue("ScanCoordinator", "source")
```

which returns a string (which, in the above example, is stored in value). If you need for the value to be another data type, such as an integer or a float, please consult your favorite Python manual to find out how to use conversion operators.

Your scientific contact person can help you if you wish to use GetValue().

F.1.2 SetValues()

The SetValues() function can be used to directly set any of the parameters within the Monitor and Control system. As a result, it is used to support complex configurations and expert observations. Please note that SetValues() does not issue a “prepare” on the M&C Manager containing the parameter. If you wish to do a “prepare”, you can also use SetValues() to do that as well. A complicated example, which assumes that you have defined values for lfYs and other variables, is the following:

```
lfcValues = {
    'local_focus_correction,Y': lfYs
  , 'local_focus_correction,Z': lfZs
  , 'local_focus_correction,Xt': lfXsTilt
  , 'local_focus_correction,Zt': lfZsTilt
  , 'localPointingOffsets,azOffset2': lpcAz2
}

SetValues("Antenna", lfcValues)
SetValues("Antenna", {"state": "prepare"})
```

Your scientific contact person can help you if you wish to use SetValues().

F.1.3 DefineScan()

If you have written your own scan type using the Python language, the DefineScan directive is used to load your new scan type into the current observing script. Once loaded, it can be referred to by name, just like any other scan type. The syntax is DefineScan("ScanName", "/path/to/my/NewScan.py") and the new scan must be written in Python.

F.1.4 GetCurrentLocation()

Given a coordinate mode, GetCurrentLocation() returns a Location object (see Appendix D) containing the coordinates of the currently selected receiver beam's position on the sky (as selected in the most recent scan type), e.g., location = GetCurrentLocation("J2000"). The position is read from the antenna at the time the directive is executed in the Observing Script. The returned location may be used in conditional statements or as an argument for scan types.

F.1.5 SetSourceVelocity()

The SetSourceVelocity() function sets the LO1 source velocity directly, in units of km/s. The syntax is SetSourceVelocity(10.5). Note that if you include the velocities in your Catalog then you do not need to use this function. (This function is mainly used in pulsar observations.)

F.2 Specialty Scan Types Submitted By Observers

F.2.1 LSFS

This scan type performs a "Least Squares Frequency Switch" where a single scan is broken into 8 equal parts such that each subscan has a difference frequency (as described above). LSFS() only works with the Spectrometer as the backend.

If you wish to use Least Squares Frequency Switching you should read GALFA Technical Memo 2005-1 by Carl Heiles.

Syntax: LSFS(location, deltaf, scanDuration, beamName)

The parameters to LSFS are

location A Catalog source name or Location object. It specifies the source which is to be tracked.

deltaf A float. It specifies the change in frequency in MHz which sets the multiplicative factor for the frequency offsets. That is, the frequency offsets are equal to [0.0, 8.5, 3.5, 1.5, -4.5, -7.5, -8.5, -22.5]*deltaf

scanDuration A float. It specifies the length of the subscans in seconds. It must be evenly divisible by 8 seconds. Each subscan (each frequency) will integrate for 1/8 of the Scan Duration.

beamName A string. It specifies the receiver beam to use for the scan. beamName can be "C", "1", "2", "3", "4" or any valid combination for the receiver you are using such as "MR12" and "MR34". The default value for beamName is "1".

The following example generates an LSFS observation of 1258+6126:

```
LSFS("1258+6126",0.0244,80)
```

F.2.2 Spider

Spider executes the specified number of slices of duration scanDuration through the specified location. Each slice is of length 2*startOffset. The argument startOffset also specifies the angle of the initial slice. The user may specify unidirectional or bidirectional subscans of length calDuration and when to run calibration subscans relative to each slice, i.e., at “begin”, “end”, or “both”.

Syntax: Spider(location, startOffset, scanDuration, slices, beamName, unidirectional, cals, calDuration)

The parameters for Spider are

location A Catalog source name or Location object. It specifies the source which is to be tracked.

startOffset An Offset object. It specifies the 1/2 length of the subscans and the angle from location of the initial subscan. That is, if you were to use startOffset = Offset(“AzEl”, “00:40:00”,“00:00:00”,cosv=True) then the first leg of the scan would start at +40 arc–minutes in azimuth (from the location) and would complete at -40 arc–minutes in AZ. If instead you used startOffset = Offset(“AzEl”, “00:40:00”,“00:40:00”,cosv=True) the first leg would start at AZ=+40 arc–minutes, EL=+40 arc–minutes, and would go to the opposite (AZ=-40 arc–minutes, EL=-40 arc–minutes).

scanDuration A float. It specifies the length of the subscans in seconds.

slices An integer. It specifies the number of subscans through location. The default is 4 (making a spider shape – i.e eight legs).

beamName A string. It specifies the receiver beam to use for the scan. beamName can be “C”, “1”, “2”, “3”, “4” or any valid combination for the receiver you are using such as “MR12” and “MR34”. The default value for beamName is “1”.

unidirectional A Boolean. It specifies whether each slice is scanned once in one direction or twice in both directions. The default is True (one direction).

cals A string. It specifies the order of calibration subscans, i.e., at the beginning of the slice subscan (“begin”), at the end of the slice subscan (“end”), or both (“both”). The default is “both”.

calDuration A float. It specifies the length of the calibration subscans in seconds. The default is 10.0.

The following example generates subscans through 1258+6126 starting the first leg 40 arc–minutes from the source’s “right.”

```
Spider("1258+6126", Offset("AzEl", "00:40:00", "00:00:00"), 30.0)
```

F.2.3 Z17

Z17 executes two circles of point subscans around location at 45 degree intervals. The first circle with a radius of startOffset and the second circle at a radius of sqrt(2)*startOffset. The initial subscan is at the angle specified by the startOffset. After circling twice, the procedure executes a subscan on location. The entire set of 17 subscans each of length scanDuration, is sandwiched between two cal subscans of lengths calDuration which consist of equal parts calibration noise signal on and off.

Syntax: Z17(location, startOffset, scanDuration, beamName, calDuration)

The parameters for Z17 are

location A Catalog source name or Location object. It specifies the source which is to be tracked.

startOffset An Offset object. It specifies the angle from location of the initial subscan as well as the radius of the inner circle.

scanDuration A float. It specifies the length of the subscans in seconds.

beamName A string. It specifies the receiver beam to use for the scan. beamName can be "C", "1", "2", "3", "4" or any valid combination for the receiver you are using such as "MR12" and "MR34". The default value for beamName is "1".

calDuration A float. It specifies the length of the calibration subscans in seconds. The default is 10.0.

The following example generates subscan points around 1258+6126 starting the first circle at the source's "right."

```
Z17("1258+6126", Offset("AzEl", "00:09:00", "00:00:00", cosv=True), 60.0)
```

Appendix G

Advanced Use of the Balance() Command

You can specify which devices are to be balanced. This overrides the default behavior of Balance() and should only be used when absolutely necessary. The syntax for specifying the balancing of individual devices is Balance("DeviceName"), where DeviceName can be any of the following: IFRack, Spectrometer, SpectralProcessor, RcvrPF_1, and RcvrPF_2.

An optional parameter to the Balance() can be a python dictionary that contains one or more of the balancing options listed below. Items which are not in the dictionary are assigned their default values. Non-applicable options are ignored. An example of using balancing options is

```
Balance("SpectralProcessor", {"target_level": -6})
```

which balances the Spectral Processor inputs to a level of −6 dB rather than the default.

The first argument is "DeviceName" and is the name of the device which you would like to have balanced. Possible values for "DeviceName" are "IFRack", "Spectrometer", "SpectralProcessor", "RcvrPF_1", and "RcvrPF_2". The second argument allows you to control the balancing. The supported keywords for this and their default values are:

"target_level" Default value is -6. This keyword is applicable only when balancing the Spectral Processor.

"port" = Default is to balance all active ports. This keyword is applicable to the Spectrometer. The keyword value is an integer list (e.g. [1,5,7,22,34]) which values between 1 and 40.

"sample_time" Default is sample every 2 seconds. This keyword is applicable only when balancing the Prime Focus Receivers. The value is an integer between 1 and 41 seconds.

"cal" Default value is "off". This keyword is applicable when balancing the Prime Focus Receivers. Possible keyword values are "on" or "off" – other values will be treated as if "on" was specified.

Examples of this advance use of Balance() are

```
Balance("RcvrPF_1", {"sample_time":5})
Balance("SpectralProcessor", {"target_level": -6})
```

List of Acronyms

GBT Auto-Correlation Spectrometer (ACS)
The main spectral line backend for the GBT. Also known as the spectrometer. 50

Analog to Digital (A/D)
A term used for the conversion of an analog signal into a quantized digital signal. 51

Active Surface (AS)
The surface panels on the GBT whose corner heights can be adjusted to form the best possible parabolic surface. 3, 26, 94

Astronomer's Integrated Desktop (Astrid)
The software tool used for executing observations with the GBT. 1, 9–23, 27, 28, 30, 32–34, 37, 41, 42, 71, 81, 82

Caltech Continuum Backend (CCB)
A wideband continuum backend designed for use with the GBT Ka-band receiver. 6

Digital Continuum Receiver (DCR)
A continuum backend designed for use with any of the GBT receivers. 6, 43, 47, 55, 93, 100, 169

Federal Aviation Administration (FAA)
The U.S. Government agency that oversees and regulates the airline industry in the U.S. 51

Finite Element Model (FEM)
This is a model for how the GBT support structure changes shape due to gravitational forces at different elevation angles. 26

Field Effect Transistor (FET)
A type of amplifier used in the receivers. 5

Fast Fourier Transform (FFT)
An approximation for a Fourier Transform which is computationally fast. 7

Focus Rotation Mount (FRM)
A mount that holds the Prime Focus Receivers which allows the receivers to be moved and rotated relative to the focal point. The FRM has three degrees of freedom, Z-axis radial focus, Y-axis translation (in the direction of the dish plane of symmetry), and rotation. 5

Full Width at Half the Maximum (FWHM)
Used as a measure for the width of a Gaussian. 29, 31, 56, 70

Green Bank Telescope (GBT)

An off-axis, 100 meter, single dish telescope. 1, 3–12, 15–18, 22–30, 33, 41, 42, 47, 51–53, 55, 59, 75, 80, 81, 84, 93, 94, 96–99, 103, 115, 118, 125

GBT IDL Data Reduction Package (GBTIDL)

Data reduction package written in IDL for analyzing GBT data. 1, 80, 131

GBT Fits Monitor (GFM)

The software program that provides real-time looks at GBT data. 27–31, 37

GBT Observing (GO)

23, 80

Intermediate Frequency (IF)

A frequency to which the Radio Frequency is shifted as an intermediate step before detection in the backend. Obtained from mixing the RF signal with an LO signal. 6, 11, 27, 34, 50, 52, 91, 93, 99

Intermediate Frequency paths (IF path)

The actaul signal path between the reciever and the backend through the IF system. 26, 49, 55, 93

Intermediate Frequency system (IF)

A general name for all the electronics between the receiver and the backend. These electronics typically operate using an Intermediate Frequency (IF). 1, 10, 41, 42, 52, 75, 80, 81, 93, 94, 99, 171

Local Oscillator (LO)

A generator of a stable, constant frequency, radio signal used as a reference for determining which radio frequency to observe. 52, 171

First LO (LO1)

The first LO in the GBT IF system. This LO is used to convert the RF signal detected by the receiver into the IF sent through the electronics to the backend. This is also the LO used for Doppler tracking. 52, 75, 99, 171, 180

Third LO (LO3)

The third LO in the GBT IF system which operates at a fixed frequency of 10.5 MHz. 169

Second LO (LO2)

The second LO in the GBT IF system. This is actually a set of eight different LOs that can be used to observe up to eight different spectral windows at the same time. 52, 101, 169

Local Pointing Correction (LPC)

Corrections for the general telescope pointing model that are measured by the observer. 25

Monitor and Control (M&C)

The group of software programs which control the hardware devices which comprise the GBT. 11, 15, 16, 25, 28, 29, 55, 179

Modified Julian Date (MJD)

24

North American Datum of 1983 (NAD83)

An earth-centered model for the Earth's surface based on the Geodetic Reference System of 1980. The size and shape of the earth was determined through measurements made by satellites and other sophisticated electronic equipment; the measurements accurately represent the earth to within two meters. 4

National Radio Astronomy Observatory (NRAO)

The organization that operates the GBT, VLA, VLBA and the North American part of ALMA. 4, 8, 10, 125, 131

National Radio Quite Zone (NRQZ)

An area around the GBT setup by the U.S. government to provide protection from RFI. 4

Ortho-Mode Transducer (OMT)

This is part of the receiver that takes the input from the wave-guide and separtes the two polarizations to go to separate detectors. 5

PulsaR Exploration and Search TOolkit (PRESTO)

A software package used to analyze pulsar observations. Mainly used for spigot card data. 10

Radio Frequency (RF)

The frequency of the incoming radiation detected by the GBGT. 6, 51, 52, 169

Radio Frequency Interference (RFI)

Light polution at radio wavelengths. 1, 4, 9, 49, 94, 103, 115, 173

Telescope Allocation Committee (TAC)

The group that decides how much observing time your proposal will get. 8, 9

Two-Line Element (TLE)

71, 79

Very Long Baseline (VLB)

A general acronym for VLBI or VLBA. 7, 53

Very Long Baseline Array (VLBA)

An interferometer which unconnected elemets run by the NRAO. 6, 7, 97, 98

Very Long Baseline Interferometer (VLBI)

The use of unconnected telescopes to form an effective telescope with the size of the separation between the elements of the inteferometer. 47, 51, 52, 97

Virtual Network Computer (VNC)

A GUI based system that is platform independent that allows you to view the screen of one computer on a second computer. This is very useful for remote observing. 9

Glossary

A

The number of air masses along the line of sight. One air mass is defined as the total atmospheric column when looking at the zenith. 118

Analog Filter Rack

A rack in the GBT IF system that provides contains filters to provide the ACS with signals of the proper bandwidth. The signals from this rack can also be sent to the DCR. 47, 169

baseline

Baseline is a generic term usually taken to mean the fitted, extrapolated across spectral lines, continuum emission in an observed spectrum. 28, 56, 94, 115

beam–width

The FWHM of the Gaussian response to the sky, the beam, of the GBT. 30

c_w

The speed of the wind. 94

C–band

A region of the electromagnetic spectrum covering 4–8 GHz. 5, 50, 51, 56, 94, 95

Converter Rack

A rack in the GBT IF system that recieves the signal from the optical fibers (sent from the IF Rack), mixes the IF signal with LO2 and LO3 references, and then distributes the IF signal to the various backends. 52, 100, 101

Dynamic Corrections

A system that uses temperature sensors located on the backup structure of the GBT to correct for deformations in the surface, and deformations that change the pointing and foucs of the GBT. 26

IDL

The Interactive Data Language program of ITT Visual Information Solutions. 1, 80, 186

IF Rack

A rack in the GBT IF system where the IF signal is distributed onto optical fibers and sent from the GBT receiver room to the GBT equipment room where the backends are located. 26, 47, 50, 52, 93, 99, 100, 169

K–band

A region of the electromagnetic spectrum covering 18–26 GHz. 5, 46, 50, 52, 56, 94, 95

K–band Focal Plane Array

New seven-beam focal plane array receiver covering 18-26.5 GHz. 5, 94, 146

Ka–band

A region of the electromagnetic spectrum covering 26–40 GHz. 5, 6, 50–52, 56, 94, 95, 99

Ku–band

A region of the electromagnetic spectrum from 12–18 GHz. 5, 50, 52, 56, 94, 95

L–band

A region of the electromagnetic spectrum covering 1–2 GHz. 5, 43, 44, 50–52, 56, 94

MUSTANG

The 80-100GHz bolometer receiver: MUltiplexed SQUID TES Array at Ninety GHz. 94, 95

P–band

A region of the electromagnetic spectrum covering 300–1000 MHz. Also known as the Ultra High Frequency (UHF) band in the U.S. (Sometimes P–band is considered to be a narrow region around 408 MHz, while A–band is the region around 600 MHz.) 50

PF1

The first of two prime focus receivers for the GBT. This receiver has four different bands: 290–395, 385–520, 510–690 and 680–920 MHz. 5, 56, 94

PF2

The second of two prime focus receivers for the GBT. This receiver covers 901–1230 MHz. 5, 56, 94

PROCNAME

A GO FITS file keyword that contains the name of the Scan Type used in Astrid to obtain the data. 23

PROCSEQN

A GO FITS file keyword that contains the current number of scans done of the total scans given by PROCSIZE in a given Scan Type. 23

PROCSIZE

A GO FITS file keyword that contains the number of scans that are to be run as part of the Scan Type given by PROCNAME. 23

Q–band

A region of the electromagnetic spectrum from 40–50 GHz. 5, 50, 56, 94, 95

S–band

A region of the electromagnetic spectrum covering 2–4 GHz. 5, 50, 51, 56, 94

σ_{w}

The two–dimensional standard deviation of the GBT pointing error resulting from the wind. 94

Spectral Processor

A spectral line and pulsar backend. Usually used for spectral line observations at low frequencies or when strong RFI may be present. Also used to observe the brightest pulsars. 6, 7, 44, 47, 50–52, 80, 169, 183

τ

The opacity of the atmosphere. 118

T_{rec}

The blackbody equivalent temperature flux that the GBT receiver contributes to the detected signal. 5

T_{src}

The blackbody equivalent temperature flux from the astronomical source. 93

T_{sys}

The blackbody equivalent temperature flux that the GBT sees if there is no astronomical signal detected. 5, 53, 93

$v_{optical}$

The velocity of a source using the optical approximation of the velocity–frequency relationship. 50

v_{radio}

The velocity of a source using the radio approximation of the velocity–frequency relationship. 50

$v_{relativistic}$

The velocity of a source using the relativistic definition of the velocity–frequency relationship. 50

X–band

A region of the electromagnetic spectrum covering 8–12 GHz. 5, 45, 50, 52, 56, 94, 95