

February 3, 1992

J.R. Fisher and F.J. Lockman

Observer Monitor and Control Requirements

CONTENTS

Preface

- A. Observer control
 - 1. Interactive Mode
 - 2. Programmable Observing
 - 3. Default setups
 - 4. The GBT's treatment of command files
 - 5. User Interface Look and Feel
 - 6. Observing Simulator
 - 7. Getting Information into the M&C system
- B. Monitoring
 - 1. Data Monitoring
 - 2. Status Panels
 - 3. Interference display
 - 4. Out of limits conditions
 - 5. Telescope logs
- C. Data
 - 1. Required header information
 - 2. Access to the data
 - 3. Data export
 - 4. Acquisition rates and data sizes
 - 5. Data responsibility
 - 6. Simultaneous use of more than one backend
 - 7. An Example of Data Preprocessing
- D. Utilities
 - 1. Source catalogs
 - 2. Ephemerides
 - 3. Auto-schedulers
 - 4. Calibration and pointing
 - 5. Online help and documentation
- E. Backend / frontend timing requirements

Appendix A: Spectral processor keyword specifications
Appendix B: Functions derived from the 140-ft
Appendix C: Data monitoring wish list

Preamble

This document contains only broad requirements for the GBT Monitor and Control system. It is not a detailed specification. The requirements are intended to be as timeless and as independent of technology and specific implementations as possible.

The basic tenet is that there should be many levels of interacting with the telescope, from very hands-on to extremely programmed and remote, and that a single experiment may go through the entire range in the course of hours or days. The experienced user may wish to do things as quickly and consisely as possible and may chafe at anything that seems cumbersome. The novice user may be most concerned with clarity and reliability. The level of monitoring required is also not fixed. Observers will want to view many cuts through the data parameter space.

There are two programming concepts with which observers may be assumed to be familiar, the programmable calculator and a computer programming language. The calculator is an ideal interactive instrument, but most if not all attempts to produce a programmable calculator by memorizing keystrokes have proven cumbersome and hard to edit. When there are more than a dozen or so operations to be performed, most scientists prefer to write a computer program, even when that program is to be used only once. We can assume the same to be true for telescope control. This requires that a telescope control language be adopted and that conceptual and syntactical bridges be built between the interactive and programmed modes of observing.

The bridge between requirements and a specific implementation will require that more detail be specified than is contained in this document. Ideally, a requirement would be written in such a way that the detail does not matter to the user, and choices can be made on technical grounds. In practice, it is not that clear-cut, but our goal is always to write the requirement as concisely as possible and to give as much latitude to the implementer as possible. Where examples are given, they are mainly to clarify our intentions, not to be a formal specification. Many somewhat arbitrary choices (usually common preferences) will need to be made for the way things are done, but the separation between fundamental requirements and these choices must be maintained.

The development of the user interface must be iterative and will need frequent feedback from astronomers. Throughout this memo we have assumed that there are no significant limitations on the bandwidth or graphical tools available to the observer, but we do require that they be based on a widely available and affordable system.

A. Observer Control:

1. Interactive Mode:

There needs to be an interactive control mode so that control of the telescope and its observing functions can be changed very quickly. Our conceptual model for this is the familiar hardware control panel, where someone dials in the desired values and pushes a button to make the whole thing go. We will call this the "control panel" mode, designed to be used by the telescope operator and telescope operations personnel, and possibly by the observer.

We visualize a virtual control panel that displays the status of the telescope (position, rates, reference position, time), the setup of various front and back ends, information on the procedure that is being run, and so on, on some sort of screen. For each of the items on the control panel there would be a display of the current value (e.g. the exact position of the telescope at that instant) and below this, a line that can be edited by pointing at it with a mouse (or keyboard cursor) and typing in new values. An "update" button, that can be clicked with a mouse, commands the system to set the functions to their new value. At each "update" command the status of the control panel will be stored and tagged with the time to constitute a log for the observation. There should be small help files for each entry.

For some entries, most notably positions, it might be useful to have four numbers on the screen: the actual, the currently commanded values, the new values, and the time-to-position, since the telescope cannot respond quickly to position commands.

The control panel should have a place for command line input so that commands, or groups of commands collected in a macro, can be executed from the panel.

A program will check each new value before it becomes the commanded value to ensure that it is legal in relation to other parameters. If not, an error message will be generated.

It would probably be good to separate the telescope control, front-end, back-end and so on into separate parts of the control panel which are invoked when they are appropriate (e.g. the control panel only displays the setting of the back end that is in current use). The panels should follow logical hardware divisions, but a few panels may need to contain fields from several hardware divisions for displays where screen space is limited.

Just as all the settings on the control panel can be stored in a labeled setup, so can a labeled setup be recalled to set all variables on the control panel. In the same sense, the setup generated by the control panel should be available to the observer to be incorporated into the observer's own programmed command file, either by reference or line-for-line.

If it is desirable to isolate the observer from direct control of the telescope (and this will certainly be the case at first) the observer's control panel mode would simply generate a setup that is sent to the operator's control panel (its values would become the new values on the screen) and only the operator could push the "update" button to command the new values. In general, the observer will be able to update variables without needing the operator's intervention, except for those variables that effect safety or telescope security.

The control panel mode of program control provides a way to design an experiment entirely as a "real-time" operation, by specifying, altering and saving various states of the control panel, then submitting the files generated by each state as a batch job to the telescope control system.

A. Observer Control:

2. Programmable Observing.

After the initial interactive setup phase of an observing session, the observer usually wants to program an observing sequence for the telescope to execute for the rest of the session. The transition from interactive to programmed observing must be as smooth as possible to minimize the number of new things that the observer needs to learn and to allow a hybrid of programmed and interactive observing during the transition or at any stage of the observing session. Many observing sessions will run entirely in programmed mode.

Interactive observing will range from GUI buttons and switches to command line input. The bridges between interactive and programmed observing should include the following:

1. Command line interactive input must be identical to program language statements.
2. Predefined keywords must have a one-to-one correspondence to entry fields in the interactive screen.
3. Interactive complete or partial setup states may be stored under a chosen name to be run as a program or command line statement.
4. An interactive session log is always generated in a form identical to the equivalent observing program, part or all of which may be saved in a file and edited with any text editor.
5. An observing program may be interrupted at the end of any statement, modified with the addition or deletion of any statement, and resumed at any statement. This is much like the interpretive BASIC programming on the 9800 series HP desktop calculators that allowed program modification in the middle of execution. While the observing program is suspended, interactive control of the telescope is permitted with program resumption either from the suspended parameter states or the new states as modified interactively.
6. Program segments (macros or procedures) may be defined and executed by name from either interactive or programmed mode.
7. Identical help documents such as keyword dictionaries and built-in function descriptions should be available as both interactive context-sensitive help or something accessible from a text editing program.

More than one interactive or programmed user interface may be defined for the Monitor and Control system, either to duplicate an existing system at another telescope or to provide more convenience for particular types of observing. One generic programming language must be defined from the start, and others may be added later. The generic language must allow complete control of all telescope and receiver system parameters. The syntax of this language remains to be defined after surveying existing telescope control languages, but the language must contain the following concepts:

Keywords - All control and monitorable parameters in the system will be assigned unique keywords. Our experience with these keywords has uncovered two sources of ambiguity which must be resolved. First, some parameters may have two meanings, commanded and actual. For example, in practice the commanded J2000 dec will differ from the actual J2000 dec of the telescope. These need to be qualified. Second, some potentially obvious keywords have different meanings depending on the context. For example, "integration_time" or "sample_time" has different meanings for a spectrometer, a continuum receiver, or pulsar observing. The whole context problem may be hidden from the typical observer by including a "context" statement in the setup. Unresolved contexts must be flagged by an error message.

The observer should be able to define an "alias" for any keyword. Minimum match rules would be very desirable.

Keywords will be treated like variables in a computer program and can be assigned values in statements that do calculations using the standard "= + - * / " symbols.

Constants, Units and Variables - Some quantities such as time and angle are normally written in several non-decimal formats, and the command processor must be capable of recognizing these formats on the basis of the keyword being assigned. Non-keyword variables are not a prominent part of telescope control, but they are needed. These must conform to the units and conversion properties of the keywords without requiring a lot of formal variable declaration rules.

Statement Delimiters - Program statements are terminated with any or all of the characters: line-feed, carriage-return, or semicolon.

Predefined Functions - Telescope system control consists of setting a number of parameters and invoking an action. This may be as simple as specifying J2000 'ra' and 'dec' and saying "move", or it could be something as complex as a point mapping operation that requires an initial position, ra and dec point spacing, integration time per point, etc. invoked with the "point_map" function. A predefined set of these functions will be supplied for the GBT, and this set will be expanded as experience requires. Users will not normally define functions (macros are available for this purpose) because many of these operations require accurate time sequencing that is not made directly available to the user except in the form of functions. Appendix B contains sample functions derived from the 140-ft control language. The 140-ft set of functions is much more extensive than this, and the adopted set will include useful functions currently used at other instruments.

With functions, the issue of parameter context raises its ugly head again. Global keyword parameters like 'ra' and 'dec' make the common parameters of one function depend on the settings of a previous function, but in many circumstances this is exactly what we want. We propose a that the solution be, in spirit, like the following:

epoch=J2000		epoch=J2000
coord=ra,dec		coord=ra,dec
ra=23:18:19.0742		move(23:18:19.074,-45:09:18.4)
dec=-45:09:18.4		
move		

These two sets of commands are intended to take the telescope to the given ra and dec, but in one case using variables, and in the other (on the right) by using the function. For these purposes it would be useful if functions could be designed so that if no argument is given, it uses the values of the variables as they have been previously defined. All keywords retain their last assigned values until assigned again, and function arguments override global keywords within the function.

Macros - User-defined macros are really convenient time savers and permit the observer to build a library of useful routines. However, they do not require the power and isolation of a function nor do they want to get involved in the formality necessary for a function. Macros are simply groups of program statements that may be named and invoked with that same name. All global keywords are available to a macro as if its statements were put directly into the main program.

Tables - The program for a source survey can get long and tedious if there is not some way to set up a shorthand for a limited number of parameters that are changed in an orderly fashion. Something like table definition and invocation commands are required. For example, the user might be allowed to define a named table of source names, positions, velocities, bandwidths, and integration times for a bunch of galaxies with

TABLE Galax(source, ra, dec, helio_vel, bandwidth, integration_time)

and execute the table as if it were a series of assignments and "move" commands such as

```
TABLE Galax
      NGC1613  02:13:25   +12:13:07.2  256.3   10   600
      NGC2345  04:25:15.6 -03:23:51   2380.0  20   600
      .        .        .        .        .        .
      .        .        .        .        .        .
      .        .        .        .        .        .
ENDTABLE
```

Table field delimiters may be a comma, a tab, or two or more spaces (one space may be used to separate hours-minutes-seconds fields).

The "UPDATE" Command - In the interactive mode one needs to do something like click a go button to say "move all of my new setup parameters into the system." This lets the observer coordinate telescope and back-end operations or to set them up independently. The same control is handy in programmed

mode. Some functions like "move" imply the update of the commanded position, but a general "update" command is still necessary for parameters for which there is no associated function.

Program Control Flow - Some form of program control is required for telescope control, but the language need not be as sophisticated as a full programming language. A modest number of control flow words like "if," "while," "do," "for," etc. are necessary, and some redundancy of control statements is probably desirable to make the language familiar to a wider range of users.

Case Sensitivity - The command language will be completely insensitive to the case (upper/lower) of keyword or name characters.

Keyword Selection

Many of the keywords required by the GBT are already in use at Tucson or Green Bank. These should be the starting list. Experience with the spectral processor showed that each new hardware subsystem will require additional keywords to go along with new functions that become available. To avoid making the adoption of new keywords a cumbersome process, the developers of each new back-end or other subsystem will be responsible for generating a list of required control parameters and assigning keywords starting with the existing list. New parameters that do not appear to be in the current list may be given tentative new names that convey the meaning of the new parameters. The complete list of keyword name will then be reviewed by a designated astronomer for compatibility with astronomical conventions and other existing keywords. The final list will then be added to the current list and documentation, and one or more interactive setup screens added to the Monitor and Control system for the new subsystem.

This same process will be used to document the control parameters and keywords for existing and planned back-ends, front-ends, and telescope control. A staff member familiar with each subsystem shall be designated to compile its list of parameters and make tentative keyword assignments. The reviewing astronomer is urged to consider keywords already in use at other sites.

The spectral processor required about 56 keywords. About a third of them fit existing keywords nicely. Another third fit with some stretching of the original definitions (some probably should have been new words). The final third were entirely new. With such a high proportion of new or potentially new keywords, it did not seem worth a great deal of effort to ensure detailed compatibility with other sites that did not have a comparable instrument. Something like a new spectrometer with simply more channels and more bandwidth should closely conform to existing spectrometer keywords.

Appendix A contains an example keyword specification for the spectral processor.

Macro Library:

A small library of commonly used macros could provide all of the programmed observing power that most of our users need. These macros can be developed along with the command language, but we must be prepared to include new macros in the library as the style of using the telescope develops. The macro library must be well documented and organized with something like a menu so that the observer may quickly see what is available. One-liner and full help explanations of each macro are required.

A. Observing Control:

3. Default Setups

Every piece of equipment including the telescope will have one or more default setups, probably in the form of macros, that may be invoked with a simple command. Two or more hierarchies of defaults should be defined to provide complete system defaults, observing type defaults (line, continuum, pulsar, etc.), subsystem defaults, or special purpose defaults. These

defaults should be sufficiently documented to provide a starting place for new observers in case they want to copy one of the default files and modify it for their purposes. The grouping of subsystem default keyword assignments should match the interactive parameter screen divisions as closely as possible.

Every programmed command file is required to begin with a default setup. Undefined control parameters are forbidden. It should be easy to store the current setup and recall it by name at a later time.

A. Observing Control:

4. What the GBT does with the command files.

As a line of the command file is executed, the date and time will be appended to it, so that the observer perusing a command file can tell instantly what was done and when. Any comments by the telescope operator should also become part of the command file or a copy of it. Switching between different observing programs in any day, or over the course of months or years, could mean just starting the command file with the first not-yet-executed statement.

There should be enough storage in the monitor and control computers that an observer's command files remain accessible for a long time (years). This seems to have been achieved already at the 140ft, and it has turned out to be extremely useful.

The state of the currently active command file can be displayed through the monitor system (see section B.2).

A. Observing Control:

5. User Interface Look and Feel

The development of the look and feel of the observing software will require close cooperation between one or more NRAO astronomers and a programmer plus user community comment at several stages of the process. Rather than write a detailed user interface specification we suggest that the initial prototype be a collation of the 140-ft functions (Appendix B) with the 12-meter control user interface and functions. This will take advantage of the years of development at both instruments. This is a look-and-feel and functional prototype, not a requirement for compatibility nor for code reuse.

Both the 12-meter and GBT systems should be allowed to innovate as they see fit, and any benefits to be derived from sharing designs or future improvements are outside of the scope of the GBT requirements. If there is a common set of user interface and functional features that can be identified for support by both telescopes, and there is a net benefit perceived by the implementors of both systems, some consideration should be given to making this a subset of both telescopes.

A few items seem certain. The vast majority of the observing programs will have multiple investigators, many of whom will want to come to Green Bank to see the GBT, at least for the first few years of its operation. This implies that the observer's space should have several connections into the M&C systems, perhaps two workstations for the current users and another for users-to-be. This is in addition to a screen that contains monitor information. We should not skimp on space or facilities for the observers.

The user interface should be based on a widely available and affordable display system. When the design is complete, a simple interface will be distilled from what we have and offered for remote users who may not have access to graphics screens.

A. Observing Control:

6. Observing Simulator

To save time at the telescope, observers should be able to run their observing program on a telescope simulator to make sure that it will work as they intend. The realism of the simulator will depend on the computer on which it is run, so it should probably be built in two parts. The basic simulator will check for syntax errors and illegal observing conditions such as observing below the horizon or setting parameters to values that are not available. Error messages would be put into a copy of the observing program so that a text editor could be used to correct the errors. This part of the simulator could be run on any computer with an 80x24 character text screen. Note that this cannot guaranty an error-free file, however, for the legality of certain commands will depend on the date and time they are executed, which might not be known in advance.

The fancier simulator will require much of the GUI and monitoring capability of the Monitor and Control user interface computer. This could be a training tool for new telescope operators or a familiarization platform for new observers. It could be run at real time speed, much faster than real time, or single stepped through an observing program. Short of seeing real data, all of the monitor and control functions could be exercised interactively or under programmed control. This should be a much quicker way to learn the system than reading a manual.

The simulator should not be a separate software package. It must include all current modifications to the telescope system, so it should be a compilation of the actual on-line code plus software simulators of hardware subsystems that were built to debug the software before the GBT was finished and to diagnose system problems when the telescope is running and not available to the programmers and engineers. The exact same simulator should be useful to both the observer and to the system maintenance people.

A. Observing Control:

7. Getting Information into the M&C system.

Besides creating files in Green Bank, the observers can move files into the M&C computers through the network, or bring them on standard media.

The M&C system must be prepared to integrate the GBT into the VLBA control system.

B. MONITORING

1. Data Monitoring

The purpose of on-line data monitoring is to provide a running check on data integrity and proper operation of the system. User and operator interaction is limited to selecting the type and options of the display appropriate to the data being collected plus some scroll-back through the most recent displays. Ancillary information that does not go on to the data analysis system could be kept in a circulating history file of a few days duration for diagnostics, e.g., in a switched continuum system one would like to be able to review the recent history of system temperature, gain, and total power level. The displays that have proved useful so far are outlined below. New back-ends or new uses of the GBT may require different display types.

Total Power:

The total power (or system temperature) is the single most important diagnostic of the status of the observations. It contains information on

everything from elevation to interference to the weather. Display of total receiver power or system temperature is useful for all types of observing.

Basically what one needs here is a multipen chart recorder that doesn't waste paper. Display speed may force some compromises, but here is a wish list for a CRT-based chart record.

1. The plotting function is some quantity like intensity or system temperature versus time with time on the horizontal axis. An arbitrary number of these plots, consistent with the graphics window size, may be selected by the user.

2. New data appears at the right edge of the window scrolling old data smoothly to the left. Left-to-right data filling with wrap-around is feasible but less desirable.

3. Old data that have scrolled off the left of the window can be recalled by manually scrolling data smoothly or by specifying a time/date to center on the screen.

4. The vertical scale for each plot on the screen can be rescaled and have its zero point set independently. This should be able to be done automatically or by giving scale factors.

5. The default horizontal scale is determined by the frequency of observation (e.g., shorter at low frequencies where interference is a problem) which can be expanded or contracted under user control. A contracted scale would use data point averaging for the display but not on the data itself.

6. There should be a way to measure the value of the data using a cursor or graphical scale that can be moved around on the screen.

7. A hard copy of the current display can be generated.

Continuum, Spectral Line, Pulsar:

Since much of the back-end-specific data monitoring requirements is a subset of the requirements for most data analysis packages, we recommend that one or more of the existing data reduction packages be adopted for data monitoring.

Some features such as event triggering (e.g., plot the next spectrum when it arrives) and scrolling might need to be added.

Similarly, data reduction tasks such as pointing solutions, calibrations, etc. will need to be added with a way to send the results back to the control software at the option of the operator or observer.

Having a data monitoring environment already familiar to the astronomer is quite desirable. Wish lists for the three types of observing are given in Appendix C to indicate the scope of typical data monitoring requirements and some of the features that are unique to monitoring as opposed to data reduction.

VLBI:

Most of the data monitoring will take place in the VLB back-end, which will most likely be the VLBA recorder. It is often useful, however, to run the standard continuum detector in parallel with the VLBI observation.

B. Monitoring

2. Status panels.

There needs to be a screen display of the current status of the telescope: its position and rates, the state of any observations that are being done, etc. This could be assembled by the user from a number of smaller panels. Examples of the subpanels would be:

spectrometer status -- a display of the configuration, bandwidth, sampling

rate and so on.

observation status -- a display of the duration of the current observation, the time it has left, values for the system temperature, etc.

frequencies and velocities -- this would show the rest, sky, IF frequencies of the spectrometers, and the doppler corrections.

command file -- there should be a subpanel that displays the observing file that is being executed, highlighting the current command.

telescope status -- positions and rates, projected time to some telescopes physical limit at the current rates, any relevant warning signals.

front-end status -- probably not too much is necessary here, perhaps just information on the cal, polarization, and a few temperature monitors.

pointing -- the values of the current pointing corrections and when they were determined; data from the laser-ranging pointing system; information on pointing errors expected due to wind and weather.

active surface --

feed and subreflector -- information on focus, turret position, feed rotation.

weather -- temperature, humidity, dew point, sky transparency from an on-site water vapor radiometer.

sky map -- the current and projected positions in relation to celestial objects, including the galactic plane, sun and moon.

operator's comments -- various system reminders, cafeteria schedule, weather forecast. For remote observation this would be a means of talking with the operator.

Some of these "panels" will require substantial graphic capability, others can be simple text fields. We must be prepared to develop some special purpose displays that include information derived from many panels.

B. Monitoring

3. Interference Monitoring

The spectral line monitoring wish list includes the necessary features for interference diagnosis using the best available spectrometer. However, when the spectrometer is in routine astronomical use, or we want to quickly survey a much broader frequency range, a standard spectrum analyzer should be available. Any number of modern commercial instruments with peak hold and averaging capabilities are adequate. The spectrum analyzer should be placed near the operator and observer's work area to make it a part of the operator's monitoring routine and easy for the observer to use in setting up observing frequencies and bandwidths. The Monitor and Control system should be prepared to accommodate additional interference monitoring equipment.

B. Monitoring

4. Out of limits conditions.

The monitor system should frequently check a number of parameters to see if they remain within their specified limits, and if not, to take some action that ranges from simply informing the operator or observer, to halting the data taking, to stowing the telescope. For the observer, some of the most important parameters to monitor would include: pointing errors, that might become intolerably large due to wind or weather; the system temperature, which might become large because of bad weather, low elevation or equipment malfunction; and the local oscillator, which might come out of lock. The observer should be able to set tolerance levels for a number of conditions, and specify the action that is to occur if the variables are out of limits. Most if not all of these parameters could be predefined. Those that are monitored at the 140-ft and the 12-meter should form a starting list for the GBT.

B. Monitoring

5. Telescope Logs

The GBT will need to generate a number of "logs" that might well be viewed as independent tasks rather than as one.

The observer's command file, after execution, will be a sort of log, since each line will be tagged with the time and possibly with operator (or observer) comments.

The status of the control panel or some other record of observing telescope actions will likewise constitute a log. This log is the one most similar to the one currently kept at the 140-ft, but it will record both interactive and programmed mode actions.

The output of a data base manager when applied to the data file will produce a data log or index. This might be done automatically and placed in a file for general documentation of the telescope's use.

Various outputs of the monitoring system will go into another log, whose extent and permanence will depend on interest in the data. For example, someone might want to monitor the dewer temperature of a receiver every few minutes for a month or so. It would be good if these data could be merged with the general operational log rather than have to be recorded by on special equipment brought in just for this purpose. This log should be specified by the operations and electronics division.

C. DATA:

1. Required header information

The header for each front-end/backend combination should contain enough information to archive the data. That is, from the header one could recreate the status of all important variables of the receiver and backend. This header should also contain enough information on the telescope status (positions, rates, pointing) to recover its behavior during the observation. We suspect that the backend, rather than the frontend or IF system, would drive the design of the header so that in practice there might only be a few to cover all observing modes with the GBT. The design of these headers needs input from all groups involved with the telescope: operations, electronics, computing, astronomical. We encourage the use of large headers -- when in doubt, put it in.

A moderate amount of information redundancy in the data header is often worth the small additional storage space. A significant amount of data that could have been lost due to a software bug or a hardware glitch has been salvaged with redundant header information. Two specific examples are telescope position (commanded, encoder readings, encoder converted to apparent, and encoder converted to epoch) and LO settings (sky center frequency, IF center frequency, all commanded LO settings, and LO frequency counter readings). Commanded values should be drawn from the actual numbers sent to the hardware, even if it requires conversion back to a format appropriate to the data header. Where possible, information on the actual settings (frequency counters, encoders, etc.) are desirable in the header.

The "archival" header, i.e. the header stored online, can be highly compressed and can use codes for various states of the equipment. In other words, a printing of the literal header values might have no meaning without use of a key to decipher it. This will have no practical impact on the user, however, who will not necessarily ever see this header. The user will see information derived from the archival header in one of any number of formats, designed for compatibility with the particular data reduction system chosen. In summary, we do not care how the data are stored online as long as the data are complete and can be accessed by the observer and easily transferred to other formats.

Observations with an extremely high data rate must be limited only by the current hardware state-of-the-art. If the data and header format(s) adopted for everyday observations degrade the available data rate, we may need to consider special formats, but we prefer a format standard that can handle all cases.

C: DATA

2. Access

The incoming data need to be accessed by a competent data base manager so that the observer can ask: "What Orion A positions were observed yesterday?" This is to a large extent a problem for the DATA REDUCTION GROUP and so is outside of Monitor and Control. The observer does not want to be aware of any distinction between on-line and off-line data access. There should be no difference between the most recently recorded record or scan and one recorded two days before. It will also be useful to have some sort of data base management system that can be applied to the telescope log (see section B.5).

All system calibration data and solutions should be archived in an easily accessible data base for the life of the telescope. This will provide a history of telescope performance and a way for observers to track problems that may be discovered in the data reduction process.

C: DATA

3. Data Export

The M&C system should present astronomical data to the user in formats compatible with common data analysis programs, e.g. AIPS, ANALYZ, IRAF, UNIPOPS etc. This must be done almost instantaneously so that the observers have access to the incoming data as it is being acquired.

C. DATA:

4. Acquisition rates and data sizes

Data Formats

The raw data from the GBT will arrive in a few general forms according to which independent parameters are changing most rapidly. For example, in a spectrometer output there will be many frequency samples for each time or position sample, while a continuum receiver will produce samples at many time/position coordinates at a single frequency. These two cases with several variations on each are outlined below as a starting place for GBT Monitor and Control design with the understanding that future signal processing systems may add new raw data characteristics. The formats given are just outlines of what should be expected since each receiver system will have its own variations that must be handled by the Monitor & Control system.

Every datum has a large number of system parameters associated with it, but most of these parameters remain fixed for many adjacent data samples. Two or more levels of header/data hierarchy are suggested to reduce the parameter redundancy, and it is this hierarchy that distinguishes spectrometer from continuum data. This hierarchy may need to be disassembled if the data are divided in the process of filling the data parameter space.

Spectrometer:

Common header

IF Channel 1 header

Phase 1 header

IF Channel 1, Phase 1 Spectrum

```

Phase 2 header
    IF Channel 1, Phase 2 Spectrum
Phase 3 header
    IF Channel 1, Phase 3 Spectrum
:
:
IF Channel 2 header
    Phase 1 header
        IF Channel 2, Phase 1 Spectrum
    Phase 2 header
        IF Channel 2, Phase 2 Spectrum
    Phase 3 header
        IF Channel 2, Phase 3 Spectrum
:
:

```

The IF Channel header contains information that pertains only to that channel such as center frequency, bandwidth, number of frequency points, and beam position in case each IF is connected to a different beam. Each phase spectrum within one IF will have the same number of frequency points, but each phase refers to different front-end or source conditions such as signal, reference, cal-on, cal-off, a given phase in a pulsar period, or one of a continuous series of short time interval spectra. The phase header will identify the conditions for each phase including information about any data excised by automatic RFI blanking. Each IF will generally contain the same number of phases.

Continuum:

```

Common header
    Position header
        Data point 1 position pair
        Data point 2 position pair
        :
        :
    IF Channel 1 header
        Data value 1
        Data point 1 qualifier
        Data value 2
        Data point 2 qualifier
        :
        :
    IF Channel 2 header
        Data value 1
        Data point 1 qualifier
        Data value 2
        Data point 2 qualifier
        :
        :

```

Continuum generally requires that each datum be associated with a telescope position encoder reading since one cannot assume that the position moves in equal intervals between data points. Each IF Channel need not contain a separate position pair since the relative position offset between channels can be assumed to be constant and called out in the IF channel header. One or more data point qualifiers may be required with every data value in every IF channel in cases where the information is not common between channels. An example would be normalization information from automatic an RFI blanking operation.

Other Formats:

The two formats shown above are very general to cover most requirement for spectral line and continuum data. In some cases, particularly where data are sampled rapidly, much of the header and qualifier information can be omitted. For example, 100 us dedispersed pulsar data from the spectral processor contains a very brief ASCII header at the beginning of the data stream, and everything from then on is in groups of 8, 16-bit values per time sample in a continuous set of 1/4-megabyte blocks without intervening headers. The Monitor & Control system should plan to accommodate special format such as this or other variations on the formats above to permit full use of existing back-ends and new ones in the future.

Data Sizes and Data Rates

In the immediate future, spectral line data will contain anything from a few hundred channels in one IF to a few thousand channels in each of up to 16 IF's. The total number of channels might grow by an order of magnitude every ten years or so. Sky mapping typically produces a set of spectra with 2 or 4 phases every 20 seconds, and long integrations usually produce a spectrum set every minute or longer. Rapid sky mapping and real time interference excision are beginning to require spectra produced every second or possibly more often so the system should be prepared for these data rates at the preprocessing stage if not all the way to data archiving. Pulsar spectra are now being produced every 10 seconds by the spectral processor with 2 IF's, 256 channels/IF, and 128 phases. This could be expanded by a factor of 10 or so if the spectral processor MassComp were replaced by a faster computer. Pulsar data will always push data transfer rates and storage requirements to their limits since these will be the limiting factors to the science for the indefinite future.

Continuum data will immediately produce 14 IF's at sample periods of a few tens of milliseconds. The number of IF's may eventually grow to 100 or more. We haven't done it, yet, but it is very likely that spectrometers such as the spectral processor will begin to see service in continuum work where interference is a problem. This probably will not add to the data storage requirements, but it will require the rapid processing of 200 to 1000 spectral channels per IF to remove pulsed and narrow band interference before collapsing the data into single intensity values.

C. DATA:

5. Data responsibility

The telescope operator is responsible only for seeing that the data are collected with apparently functioning equipment. The Observer is responsible for the ultimate quality of the data, its calibration, and so on. For example, the observer has the responsibility to make sure that the velocity range and velocity resolution of a spectral observation match the line under study, and that the reference frequency (or position) is emission-free. We should have a package of "tools" available for the observer to help insure data quality. These might include pointing checks, procedures that make observations of standard sources and compare the data with what is expected. Some of these are discussed in section D. The operator and especially the "friend of the telescope" will be available for consultation with the observer, and will assist in technical matters, but it is the observer's duty to ask them for help.

C. DATA:

6. Simultaneous use of more than one backend

Over the years there have been a few requests to be able to record continuum and spectral line data simultaneously for programs such as recombination line measurements where line-to-continuum ratios are important. Had we had the computing resources to do this, it is not clear how extensively the use of more than one back-end at a time might have grown. On the assumption that such a capability will enhance the efficiency of the GBT, the back-end control and data processing and storage parts of the Monitor & Control system should be design in such a way that two or more data types may be recorded simultaneously either with complete independence between the back-ends or with appropriate synchronization between them in a master-slave relationship.

For example, one might do a pulsar search while doing either continuum or spectral line mapping, or the spectrometer might be the primary back-end that controls front-end load and cal switching, and the continuum receiver could follow these switching cycles to record calibrated continuum data with an independent integration period. Separate data

records with their own headers would be produced for line and continuum data and independently stored on disk.

C. DATA:

7. An Example of Data Preprocessing

Pulsar observers have often provided their own form of real-time data preprocessing to reduce the data storage load, but this has not been common in continuum or spectral line work except for summing all records within a scan. Here is an example of spectral line preprocessing currently under development that must be provided for in the Monitor and Control system.

Both wideband and narrowband interference is variable on time scales less than a second. Even below 1 GHz, most of the spectrum at Green Bank is relatively interference free, but the uncontaminated pieces of spectrum are broken into relatively small chunks in both frequency and time. Instead of blindly integrating for several minutes at a time, we intend to record spectra every second or so, scan the spectra for wideband interference bursts and throw out the contaminated data, identify all of the narrowband signals and clean their effects from adjacent channels, and sum the cleaned spectra into a total spectrum for the longer integration period. With only 2048 spectral channels this requires processing about a 1/2-megabyte of data with several algorithms in one minute. More complex algorithms may be expected in the future.

D. Utilities:

1. Source catalogs

The GBT will need extensive online source catalogs as well as a modest data base management system to allow them to be used efficiently. Some of this might be accomplished by network connections to SIMBAD and other reference services that are coming on line, but some level of support should exist independent of these other agencies. Along with source catalogs should be tables of the frequencies of spectral lines and information on transmitters and the interference environment at Green Bank. Users should be able to place catalog data in an observing file.

2. Ephemerides

The GBT should give observers access to several programs to check their observations, give an independent calculation of precession, radial velocities, pulsar timing, and so on. These programs should include DOPSET, the Naval Observatory's Floppy Almanac, HAZEL, and many others that are floating around.

3. Auto-schedulers

A survey program might benefit from an automatic scheduling program that would operate on a Table of data and order it for most efficient observing.

4. Calibration and Pointing

The GBT should have a high level of automated calibration and pointing checks. There should be an online list of calibration and pointing sources, for each frequency band, that can be measured by canned routines and the positional and intensity information presented in a form that can easily be fed back into the control system. It will be necessary to have a high level of data display in these routines so that the operator and observer can decide if the measurement was good. It should be made easy to manually override all of the defaults (e.g. the size of the area which is searched for the source) in case pointing is or is suspected to be seriously in error.

5. On-line help and documentation

Ideally we would like every defined variable to have its own online help file which gives default values, a range of permitted values, and perhaps reference to relevant publications for more detailed information. Helps should contain many, many examples.

E. Backend/front-end timing requirements

Except for the fact that they must be coordinated to respond to the observer's requests, the receiver back-end, front-end, and local oscillator are nearly independent. However, they do require a bit of synchronization for firing calibration signals or switching reference loads or frequencies on time scales much too short to be transmitted through the software systems. Hence, a few direct signal lines must be connected between these subsystems, usually with a back-end as the driver and the others as slaves. Since one of several back-ends may act as the master, the Monitor and Control computer must inform all back-ends who is master and who must only "listen" to the switching lines. This is a brief description of the timing and coordination requirements for these functions.

Frequency switching - The L.O. currently proposed for the GBT provides for many presettable frequencies which may be selected with digital switching lines. The Monitor and Control computers will have to load the frequencies into the L.O. synthesizer and set up the appropriate back-end to properly sequence the digital lines. Because the synthesizer takes some milliseconds to settle on a new frequency, switching rates will be less than 10 Hz or so, and switching edges must be accurate only to about 100 microseconds.

Load or beam switching or noise adding radiometry - Front-end reference switching must be faster than time scales of receiver gain fluctuations and shorter than continuum integration times. Modern receivers are stable on time scales less than 100 milliseconds, but integration times can be as short as 10 milliseconds. Switching edge resolutions better than 100 microseconds should be adequate.

Calibration signal - For intensity calibration purposes the front-end noise source need only be switched fast enough to follow receiver gain fluctuations, usually only a few Hz. However, pulsar measurements require synchronization of the calibration with the pulsar period. The calibration signal can also be a handy check on the overall system timing, so its turn-on time should be predictable to the level of the best expected pulsar timing accuracies. This will be somewhere in the range of 100 nanoseconds for the GBT. The fastest known pulsar period is 1.6 milliseconds, so switch rates will be on the order of 1 kHz or less with typical duty cycles of 10 to 50%.

Back-end switching signals must not be required to be synchronized with any software clocks within the Monitor and Control system. The only external time synchronization should be absolute time, e.g., UTC or LST.

Appendix A

Spectral Processor Keywords Specifications

Below is a rough description of the purpose of each keyword and the units where applicable. The index N is the spectral processor rack number (only N=1 available for now) and M is the IF channel number (at least one and, hopefully two available for first run).

atod_input_lev_N_M	AI _m	Sets the noise level at the input to the flash A/D convertors. The unit is the number of quantization intervals per noise rms level, normally between about 1.0 and 4.0. Lower numbers give more large signal handling room in the A/D but compromise sensitivity and baseline stability because of quantization effects. The IF attenuators will be set to produce an input level within about 0.5 dB of the one specified.
atod_lev_mode_N	AM	Determines when the attenuator settings are changed. There are two choices: immediately, or at the start of every setup.
atod_offset_N_M	AO	Flash A/D offset level in microvolts (usually in the 100's of microvolts range). This value is normally set automatically, but it can be set directly through this parameter.
balance_N	BA	Specifies whether to set the IF attenuators before the next setup starts. The choices are "Yes" or "No."
bandwidth_N	BW	Sets the total bandwidth in each IF passband. All bandwidths within one rack must be the same. Legal selections are 40, 20, 10, 5, 2.5, 1.25, 0.625, 0.3125, 0.15625, and 0.078125 MHz. The total bandwidth for one rack is 40 MHz. In other words, the product of the number of IF's (num_spectr) and the bandwidth of one IF must be 40 MHz or smaller.
blank_state_N_M	BS _m	Specifies the state of the blanking control line for each switching phase. The states can be "on" (blanking active) or "off".
bypass_stage_N_M	BP _m	Sets the status of each board in the spectral processor rack to either normal or bypass for hardware diagnostic purposes. For normal observing all of these parameters must be "No". m = 1: window board m = 2-12: FFT stages 0-10 m = 13: real corrector stage m = 14: square/cross real m = 15: square/cross imaginary m = 16: square/cross switched
cal_duration_N	CT	Specifies the duration the cal signal in fraction of a pulsar period in the frequency/time matrix observing mode.
cal_phase_N	CP	Specifies the pulsar window phase for starting the cal signal in the frequency/time matrix observing mode.
cal_state_N_M	CAM	Specifies the state of the cal control line for each switching phase. The states can be "on" or "off".
clip_level_N_M	CL _m	Sets the maximum level that can be instantaneously applied to the input of the slow baseline integrator. This clipping prevents severe over-charging and, hence, slow recovery of this integrator. This parameter is specified in fraction of the total IF power plus noise peak level with the given bandwidth. If this parameter is zero, the clipper level is set right at the top of the noise peaks. A value of 1.0 allows a 3 dB in noise total power before clipping.
clock_freq		Sets the high resolution synthesizer frequency when it is being used to drive the spectral processor

CF	master clock. The normal value is 160 MHz. and all legal values are less than or equal to this frequency.
clock_freq_deriv CD	Sets the rate of change of clock_freq in units of Hz/Sec. When this value is not zero, 'clock_freq' is the frequency at 'utc_start'.
clock_source CS	Controls whether the spectral processor master clock is driven from its internal 160 MHz oscillator or from the external synthesizer.
dead_fft_stage_N_M DSm	List of hardware FFT stages that are unusable (0 - 10) Here M is an index not associated with IF number.
default_setup DF	Specifies the set of default keyword values to be loaded at startup of pulsar. The legal values are the same as for processor_mode_N. Both are set up to the same mode.
dispersion_measure_N DM	Specifies the pulsar dispersion measure for spectral processor modes associated with dedispersion. The unit is parsecs/cm**3.
excise_N_M EXm	Turns on or off automatic RFI excision based on total power threshold pulse detection for each IF channel. Legal values are "Yes", "No" where "Yes" means that excision is active.
fast_time_const_N_M FCm	Specifies the response time of the total power pulsed RFI threshold detector. The optimum value depends on the nature of the RFI but is often roughly equal to the characteristic pulse length of the interference. Legal values are 1us, 3us, 10us, 30us, 100us, 300us, 1000us, 3000us, and 10000us.
first_drop_chan_N_M FDm	Specifies the beginning of a range of frequency channels which are not to be stored by the accumulator if any averaging in the frequency dimension is requested. This parameter is paired with 'last_drop_chan_N_M'. Here M is an index not associated with IF number.
iff_N_M IFm	Center frequencies of IF passbands. After taking into account the baseband offset and other conversions in the IF drawer, these parameters set the synthesizer frequencies in the IF drawers or the frequency of the high resolution synthesizer if an external IF LO is selected.
if_lo_source_N_M LSm	Controls whether an IF drawer has its frequency set by its internal synthesizer with 10 kHz resolution or by the external synthesizer with 10 Hz resolution.
if_sideband_N_M ISm	Selects the active single-sideband convertor sideband. With "Upper" sideband, increasing frequency at the IF corresponds to increasing frequency at baseband. With "Lower" sideband, increasing IF produces decreasing frequency at baseband.
last_drop_chan_N_M LDm	Specifies the end of a range of frequency channels which are not to be stored by the accumulator if any averaging in the frequency dimension is requested. This parameter is paired with 'first_drop_chan_N_M'. Here M is an index not associated with IF number.
lst_start LB	Start time in Local Sidereal Time. This is converted directly into utc_start_M. The UTC derived is actually UT1, so there will be a slight offset in the start time according to the current UT1-UTC value
lst_stop LE	Stop time in Local Sidereal Time. This is converted directly into utc_stop. The UTC derived is actually UT1, so there will be a slight offset in the start time according to the current UT1-UTC value.

max_pwr_N		Specifies the largest signal that can be present in the output spectrum without overflowing the 16-bit accumulator input. The data from the square/cross multiplier is a 32-bit word, but only the top 16 bits are presented to the accumulator. This parameter sets left-shifting of the two 16-bit input words to the multiplier to between two and five bits. Greater shift means a smaller maximum power. The unit for this parameter is the fraction of total noise power in the passband. This can be less than or greater than one.
MP		
multiplier_mode		Sets the multiplier following the real correction to square the output of each complex FFT, cross multiply the outputs from racks A and B, or do both if the bandwidth permits the multiplier to run twice as fast as the FFT. Legal values are "Square", "Cross", and "Sqr/Cross".
MM		
num_chan_N		Number of spectral channels assigned to each IF input. For normal operation this parameter is nearly redundant with "num_spectr", the exception being 2 IF's with either 256 or 512 channels. The num_spectr/num_chan selections are (8/128, 4/256, 2/256, 2/512, and 1/1024). If a smaller number of channels than is permitted by 'num_spectr' is specified, frequency averaging will be done in the accumulator before sending data to the MassComp. A smaller number of channels could be chosen to get spectra more rapidly since the data rate into the MassComp is a limiting factor. The number must be a power of two.
NC		
num_ffts_per_bin_N		Sets the number of FFT's in each time bin in the frequency/time matrix mode. This parameter is used only when the matrix parameters to match the pulsar period are not determined automatically.
NF		
num_phase_N		Sets the number of time bins in the frequency-time data matrix. If this value is negative, the best number is chosen within the constraints of the master clock frequency. Otherwise, the user is responsible for specifying numbers that keep the clock within a usable range.
NP		
num_spectr_N	NS	Number of IF inputs to each rack (1, 2, 4, or 8).
observ_freq_N		Sky frequency of the center of the passband being observed. Must be specified to convert a dispersion measure into a time delay.
SF		
observ_mode		Specifies the general type of observing to be done with the spectral processor or any other back-end for that matter. Only two types refer to the spectral processor: "Line" and "Pulsar." "Line" specifies straight spectrum averaging for fixed integration periods possibly with cal and front-end switching. All other accumulation modes fall under "Pulsar."
OM		
phase_start_N_M		Specifies the beginning phase of each switching phase in fractions of a switching cycle (0.0 to 1.0). Each phase will have associated states of the cal, sig/ref, and blanking signals specified by CA, SR, and BS.
PSm		
processor_mode_N		Specifies the configuration of the spectral processor and, hence, sets the context in which all other parameters are interpreted. Only the first two characters are required.
PM		
		"DedispTimeSamples": dedispersed intensity vs time output
		"TimeSyncSpectra": spectra accumulated at selected phases of a pulse period for many pulses
		"SigAvgDedisp": dedispersed intensity vs time folded for many pulse periods - synchronous signal averaging
		"SpectraBurst" (not implemented): no averaging of spectra - data collected in short bursts whose average

rate can be handled by the MassComp input
 "FreqTimeAvg": continuous series of spectra that have been partially averaged in time and frequency
 "VoltageTimeSamples" (not implemented): untransformed time samples - data collected in short bursts whose average rate can be handled by the MassComp input
 "StdSpectLine": most common spectral line mode - front-end cal, sig/ref, and blanking control and synchronous spectral integration in a variety of phase combinations
 "PowerTimeSamples" (not implemented): untransformed but squared and averaged time samples - data may be collected in short bursts whose average rate can be handled by the MassComp input
 "PulsePhaseSpectra": spectra accumulated at selected phases of a pulse period - really just a hardware variation on "TimeSyncSpectra"
 "SyncFreqTime": accumulation of spectra at adjacent and equal width time intervals over a full pulse period for many pulses

pulsar_period_N		The period of the pulsar to be observed in UTC seconds, normally specified for the mid-point time of a scan.																					
PP		This is used to set up synchronous accumulation in some pulsar modes.																					
pulsar_period_deriv_N		The derivative of the pulsar period, normally specified for the mid-point time of a scan, in UTC seconds per UTC second. This value may be used to track a pulsar period in a long scan of synchronous averaging by updating the clock rate at short intervals. The update interval may need to be as short as a second depending on the magnitude of the derivative.																					
PD																							
pulse_offset_N		Pulse window start time offset in fractions of a pulse period. The value must be between > -1.0 and < 1.0, and a positive value means an earlier start time.																					
PO																							
rf_sideband_N_M	RSm	Front end sideband for each IF channel																					
sample_time_N		Specifies the total accumulation time of one data dump to the spectral processor MassComp. This time will be adjusted an integral number of accumulation cycles in whatever observing mode is being used.																					
ST																							
scan_duration	SD	Can be used in place of utc_stop.																					
source	SO	Source name, e.g. 0329+54																					
slow_time_const_N_M		Specifies the response time of the baseline comparison level for the total power pulsed RFI threshold detector. The optimum value depends on the nature of the RFI and is usually more than 10 times the characteristic time scale of the pulsed interference. RFI detection occurs when the fast-time-constant response exceeds the slow-time-constant response by the 'threshold_N_M' value. Legal values are 100us, 300us, 1000us, 3000us, 10000us, 30000us, 100000us, 300000us, and 1000000us.																					
SCm																							
status_N		Rack (FFT engine) status. AR and BR, in combination, specify overall spectral processor configuration as follows (x = don't care, z = any other value):																					
AR, BR		<table border="0"> <tr> <td>AR</td> <td>BR</td> <td>Configuration</td> </tr> <tr> <td>Yes</td> <td>No</td> <td>Rack A only</td> </tr> <tr> <td>No</td> <td>Yes</td> <td>Rack B only</td> </tr> <tr> <td>Yes</td> <td>Yes</td> <td>Racks A & B operating independently</td> </tr> <tr> <td>BR</td> <td>x</td> <td>Racks A & B set up identically*</td> </tr> <tr> <td>x</td> <td>AR</td> <td>Racks A & B set up identically*</td> </tr> <tr> <td>z</td> <td>z</td> <td>Racks A & B set up identically*</td> </tr> </table> <p>*when AR=BR the parameters for rack A are copied to rack B and parameters for rack B (except DS [dead fft stages], CL, EX, FC, SC, and TH [rfi excision parameters]) are ignored.</p>	AR	BR	Configuration	Yes	No	Rack A only	No	Yes	Rack B only	Yes	Yes	Racks A & B operating independently	BR	x	Racks A & B set up identically*	x	AR	Racks A & B set up identically*	z	z	Racks A & B set up identically*
AR	BR	Configuration																					
Yes	No	Rack A only																					
No	Yes	Rack B only																					
Yes	Yes	Racks A & B operating independently																					
BR	x	Racks A & B set up identically*																					
x	AR	Racks A & B set up identically*																					
z	z	Racks A & B set up identically*																					

switch_period_N SP	This is another name for pulsar_period_N when using a switching cycle that is not associated with a pulsar. To avoid the confusion of redundancy the spectral processor uses the same variable for both.
sig_ref_state_N_M SRm	Specifies the state of the sig/ref switch control line for each switching phase. The states can be "sig" or "ref".
taper_N_M TAm	Specifies the weighting function to be applied to the A/D amplitude-vs-time series before it is transformed. A small number of taper functions are stored in the spectral processor and will be referred to by name in this keyword. The index M may be 1 or 2 since two taper functions may be loaded into the hardware to be selected from with the keyword 'taper_select_N'. In spectral processor hardware memos the taper function is called a window. Currently available tapers are "Box" (equal weighting of all input samples), "Halfbox" (equal weighting in center half of samples, zero on the ends), and "Cosine."
taper_offset TO	Selects whether the data sampling for the time series to be transformed in racks A and B are started together or are offset by half of a series length. If both racks look at the same IF signal, the offset provides enhanced sensitivity when spectra from the same IF are added together. The penalty is that there are half as many spectral channels to spread around the frequency and polarization dimensions. A value of "Yes" puts the offset into effect.
taper_select_N TS	Selects one of two taper functions in hardware for each rack. In other words, it tells which value of M in "taper_N_M" is active. Values are either 1 or 2.
t_cal_N_M TCm	Calibration source in Kelvins. This parameter is simply passed into the data header for later data reduction.
threshold_N_M THm	Specifies the level difference between the outputs of the fast and slow integrators that will flag an RFI pulse. This is specified in fraction of total IF peak noise excursions from the average DC detected level with the given IF bandwidth and fast and slow integrator time constants. This number must be ≥ 1.0 .
test_source_N TX	Selects the data source to the input buffers. This is used only for test purposes and must be set to "Normal" to receive data from the flash A/D convertors. The selections are "Normal", "Long" (pseudo-random noise with a very long repetition period), "Fixed" (constant input value given by 'test_seed'), and "Short" (pseudo-random noise with a short repetition period).
test_seed_N TZ	Seed value for pseudo-random test noise (TX=Short or Long) or the fixed value when TX=Fixed.
utc_start UB	Determines the precise time of taking the first data sample in the first time series to be sent through the FFT pipeline at the beginning of a scan. The minimum resolution of this time is 200 ns or the flash A/D sample period for the bandwidth in use, whichever is greater. If the accumulation cycle parameters are determined by the pulsar period, this time may be adjusted to the next pulse mid-phase.
utc_stop UE	Sets the time to stop sending accumulated data to the spectral processor MassComp to the nearest integer second.

Appendix B

Sample functions derived from the 140 foot control system

This list is representative of the kinds of primitive functions that the GBT control system will need. A more complete list of the 140-ft control functions is in the memo by Bob Vance. The procedures given at the end of that memo are extremely valuable examples of the complex tasks that the GBT will perform. For simplicity we will use the same function names that are used at the 140-ft, even though some of them are not very descriptive. The GBT needs this functionality, but numerical examples are not meant to be requirements.

SPOBS -- Make a frequency-switched spectral observation. Starts taking autocorrelator data while switching the LO at a 1 Hz rate, and periodically turns on a noise source to calibrate the data. This function operates independently of telescope position so does not know if the telescope is even on source. SPOBS runs for a length of time given by the value of a variable TDUR.

Every "integration time" (a variable chosen by the observer that is typically 20 seconds to 1 min), SPOBS calls other functions to update the frequency setting of the local oscillator and to update the focus position.

STALL -- This function causes the system to go into a wait state until the telescope's actual position agrees with the commanded position (within a given tolerance) for a period of a few seconds. STALL then passes control to the next command in the file.

ONTPO -- Make a total-power spectral observation. Starts taking autocorrelator data at a fixed LO frequency and periodically turns on the noise source to calibrate the data. Data taken with this function are flagged with an "on source" code. The function OFTPO is identical to ONTPO except that it flags the data as "off source". Neither function checks that the telescope is on position. ONTPO runs for a duration given by the variable TDUR.

Every "integration time" (a variable chosen by the observer that is typically 20 seconds to 1 min), ONTPO calls other functions to update the frequency setting of the local oscillator and to update the focus position.

MODFOCUS -- A variable which, if true, causes the prime focus receiver to move in and out of focus by $\pm 1/8$ wavelength in synchronism with the data taking cycle to reduce standing waves in spectra. This variable causes focus modulation every time the focus routine is called by functions SPOBS and ONTPO (typically every minute).

CDOB -- Begins to collect continuum data and continues to do so for a specified amount of time. The data samples are averages over an interval selected by the observer (e.g., 0.4 seconds). CDOB activates the A/D which passes data to it every 50ms.

CDCL -- Uses CDOB to produce sets of continuum measurements with and without the noise source on, that it then averages and uses for a system calibration.

MTP0 -- Move To Position Zero. This commands the telescope to move to the specified position (with zero offset from this position) in the specified coordinate system. The move is made at the maximum rate in both coordinates. When the telescope reaches the desired position it maintains itself at that position. Because of this, if the coordinates are ra and dec, it will track the position; if

they are az and elev, the telescope will sit still.

MTP0 initiates the move and then turns control over to the next statement in the command file. If the position is illegal, the function will halt the command file execution and signal an error.

MFPWR -- Move From Position With Rate. This commands the telescope to move from the current position in a direction specified by the values of the rate variables. This function initiates the move and then turns control over to the next statement in the command file. The telescope movement continues at the specified rate until a HALT command is reached. It is possible to drive the telescope into a limit with this command.

Appendix C:

These are wish lists for data monitoring functions for three types of observing.

1. Continuum:

The monitoring needs for continuum observations are largely given by the chart recorder described in the main memo, but with a bit more sophistication and a few extra choices: any available data function (switched difference, total power, gain, system temperature, etc.) for any receiver channel may be assigned to any of the plots in the window with a menu and selection cursor. A new selection will be plotted as if it had been on the screen for the duration of the displayed time.

2. Spectral Line:

The fundamental plot for spectral line observing is frequency or velocity on the horizontal axis and intensity on the vertical axis. The wish list is as follows:

1. Each spectrometer IF will be plotted in its own box with labeled ticks on the horizontal and vertical axes.
2. Several plot boxes may be shown at one time as allowed by the size of the graphics window.
3. In monitoring mode, every assigned plot is updated automatically when new data are available.
4. Available plotting functions are total power, $((\text{sig} - \text{ref}) / \text{ref})$, and $(\text{sig} / \text{ref})$. The reference may be a previous total power scan or the reference part of a load or frequency switching cycle. Accumulated signal for the current scan or individual records may be displayed. In total power mode the reference used may be selected by the observer or operator.
5. The user may assign any function for any spectrometer IF to any plot box.
6. The vertical scale for each plot on the screen can be rescaled and have its zero point set independently, either automatically or by specifying a scale factor, or with GUI tools.
7. The default horizontal scale is something like one point per pixel, but this can be expanded or contracted under user control. A contracted scale would plot two or more data point at the same horizontal position.
8. Two horizontal and two vertical line cursors in the form of a box with settable diagonal corners is available for measuring the height and frequency or velocity of a feature in a plot. Up to ten vertical arrows are available for marking features in the spectrum to be monitored visually from one plot to the next. These arrows may be set with a mouse or by specifying a frequency, velocity, or channel number.
9. The horizontal scale may be selected to be velocity, sky frequency, intermediate frequency, or channel number. Any horizontal scale may be reversed.
10. A reasonable number of previous scans are kept in a circulating history buffer so that the observer may quickly review some of the most recent data. A forward/reverse sequencing button is provided for thumbing through this data.
11. Continuum functions such as gain, system temperature, and total power may be displayed in parallel with spectral line data as a check on receiver performance. These can be derived from simultaneous operation of the continuum back-end or possibly from a simpler total power detector and analyzer. This monitor data can be displayed in a separate window using the continuum display options.

3. Pulsar:

At this time there are four pulsar modes available.

Pulse synchronized spectra - This is essentially the same as the spectral line observing with the same requirements for spectrum plots except that there may be many more spectra, corresponding to different phases in the pulse period, in each record. The observer will want to choose which phases are used for reference and which for signal in the $((\text{sig} - \text{ref}) / \text{ref})$ or $(\text{sig} / \text{ref})$ displays.

Fast sampling - This is generally the same as continuum data with the same display requirements except that the data rate is usually much higher

than can be practically displayed. The best that can be done is to display small chunks of the data stream at user specified intervals.

Signal averaging - The displays for this type of data are the same as for total power spectral line except that the horizontal axis is pulse phase instead of frequency. The same plot, cursor, marker and scaling options apply. The horizontal axis may be labeled in either pulse phase, relative time, or absolute time (UTC).

Frequency and pulse phase data matrix - Here we have intensity as a function of two independent variables, sky frequency and pulse phase, which requires a grey scale plot in addition to the plots used with signal averaging and total power spectral line. The wish list is an expansion on what is already available on the spectral processor as follows:

1. Each spectrometer IF has its own grey-scale plot with as many plots as can be fitted in the available display window. The user may select which IF's are displayed and sequence through the IF's under manual control.

2. The grey-scale plot vertical axis is sky frequency increasing from bottom to top. The horizontal axis is pulse phase. The size of the plot will be set by the display resolution and the number of frequency and pulse phase samples. Grey-scale zooming is not necessary for monitoring.

3. The plotted data are normalized to remove the receiver passband shape by dividing each row (frequency channel) by the off-pulse mean. The current normalization process requires finding the pulse peak phase after dedispersion, but the median for each row may be just as good and computationally less intensive.

4. The default intensity transfer function sets the peak values at full white and black. The user may chose and fix the white and black normalized intensity values to avoid having interference influence the transfer function.

5. The horizontal and vertical axes are shown as a box with labeled ticks. The vertical axis may be labeled with sky frequency, intermediate frequency, baseband frequency, or channel number at the user's discretion. The horizontal axis may be labeled with pulse phase (0.0 to 1.0), time bin number, or relative time (0.0 to pulse period).

6. A cross-hair cursor is available for determining the coordinates and normalized intensity of any feature in the grey-scale plot. Up to ten vertical and horizontal arrows are available as markers of features in the plot to be watched from one data record to the next. These may be set with a mouse or by specifying a coordinate in the available values for the particular dimension (pulse phase, bin number, time, etc.).

7. The data matrix may be averaged over all pulse phases to create a total power spectrum that may be displayed with any of the spectral line total power plotting options.

8. The data matrix may be averaged over all frequencies using a dedispersion algorithm to create a pulse profile identical to a signal averager data record. This pulse profile may be plotted with any of the options available for the signal averager data. The dedispersion algorithm should use linear interpolation, but, if this takes a significant amount of computing time, a simpler nearest-point algorithm may be an option.

9. A peak-fitting routine may be activated to determine the arrival time and fractional bin number of the dedispersed pulse profile to be recorded in a continuous log as a check on the timing stability of the pulsar processor.

10. Continuum functions such as gain, system temperature, and total power may be displayed in parallel with pulsar data as a check on receiver performance. These can be derived from simultaneous operation of the continuum back-end or possibly from a simpler total power detector and analyzer. This monitor data can be displayed in a separate window using the continuum display options.