A Proposed Green Bank Interferometer Control System

J. Ford and F. Ghigo

May 5, 1999

1 Introduction

The GBI is still useful monitoring certain sources. Recently, the USNO 20 Meter operations were moved to the new Jansky Lab control center. It is required that the control of the GBI be moved to the control center as well, in order to allow economical operation in an RFI shielded environment. In order to accomplish this, the DDP-116 must be replaced with a system that allows more distributed control, using serial interfaces. We have chosen to implement the control system for the GBI using the Ygor system as a base.

2 The GB Interferometer

The Green Bank Interferometer, in its current incarnation, consists of two eighty-five foot polar mounted radio telescopes fitted with cooled S/X band receivers. These telescopes are controlled by a central computer, the DDP-116, which also controls the digital delay rack system. Figure 1 shows a block diagram of the system.

2.1 Existing Telescope Hardware

The Interferometer telescopes consist of a polar mount with focus and rotation drives for the prime focus receiver. The declination axis is driven by one motor-brake-clutch assembly, and the polar axis is driven by a different arrangement of motors and gears. This whole works is currently controlled by the DDP-116 computer from the Interferometer Control Building.

The dual polarization S/X Band receivers are cooled with standard helium refrigerators. Control and monitoring of these receivers is through a standard MCB control board.

2.2 Backend

The IF signals from the receivers are brought back to the interferometer control building where they are converted to baseband and piped into the digital delay

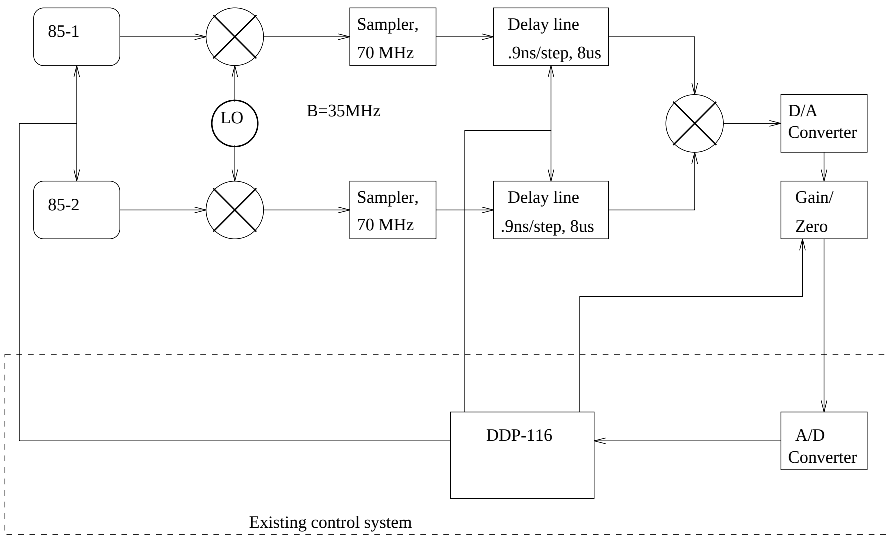

Figure 1: Green Bank Interferometer Block Diagram

rack, sampled, and multiplied. The multiplier outputs are converted back to analog signals and sent through a variable gain stage and an autozero stage before being sent to the A/D system on the DDP-116. Fringes are fit to the data, and the outputs stored.

2.3 Control Computer

The Control Computer consists of the DDP-116 computer connected to the telescopes and the digital delay rack. Wiring from the interferometer control building travels down the interferometer line and controls the telescopes.

3 New Control System Hardware

New hardware for the system consists mostly of interface cards and a new control computer. There will be very few changes in the basic operation of the Interferometer hardware itself. Figure 2 shows a block diagram of the new control system.

3.1 Telescope Hardware

The control relays for the system will be replaced with a system similar to that already in place for 85-3. An MCB interface will drive these relays and control

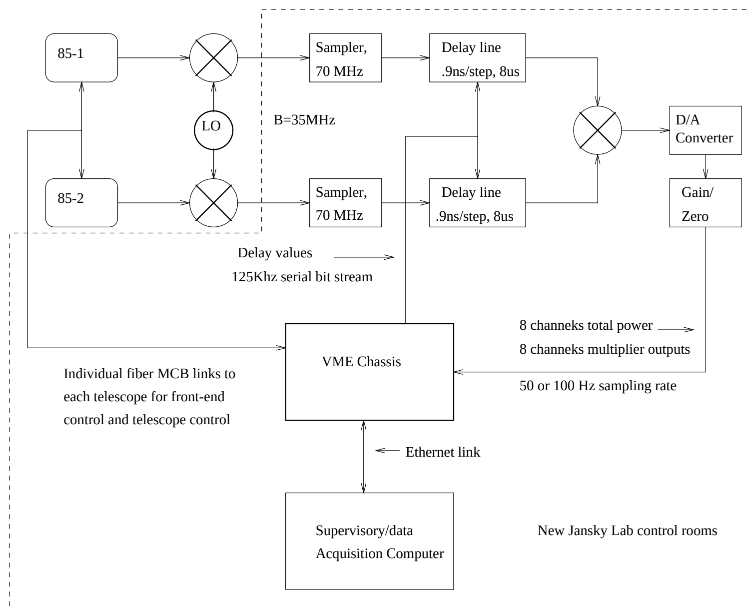

Figure 2: New System Block Diagram

the motor drives and clutches that control the speed of the axes. The receivers are also fitted with an MCB interface card.

3.2 Computer System

A new control computer based on the Motorola MV167 Single Board Computer will be installed in the new Jansky Lab Addition control room, and will control the interferometer telescopes via fiber-optic links to the telescopes. The configuration of the VME computer chassis is show in Figure 3.

3.2.1 Single Board Computer

The real-time needs of the telescope are met using a Motorola MVME-167 68040 based SBC, the same one found on the GBT. See Table 1 for the features found on the MVME-167.

3.2.2 Digital Delay Rack Interface

The digital delay rack interface is built using a parallel I/O module and a FIFO-buffered serial output module. These are both off-the-shelf IndustryPack modules from SBS Greenspring Modular I/O. The IP-BSDO FIFO-buffered serial output module is composed of a 1Kx16 FIFO chip and a Xilinx FPGA

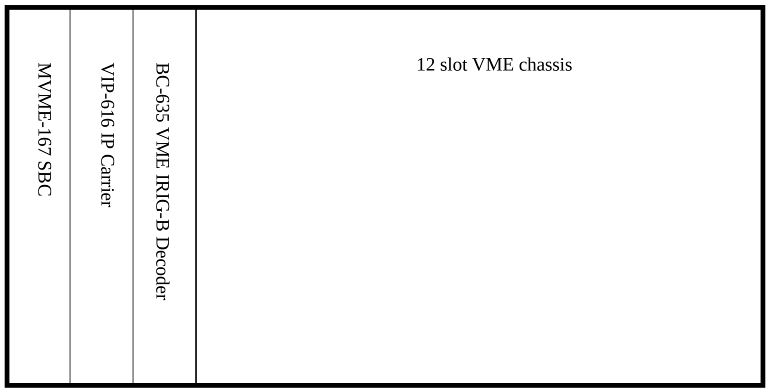

Figure 3: Computer System Hardware

logic chip. The module implements a protocol compatible with the digital delay rack. See the Appendix for a data sheet on this module.

Miscellaneous I/O, including the digital signals needed to drive the gain and autozero stages, are provided by the Acromag IP480 Digital I/O module, which provides 24 60 volt open drain I/O points. A data sheet for this module is included in the Appendix.

3.2.3 Data Acquisition

The data is acquired by the computer after it is correlated by the delay rack and multiplier. 8 Total power and 8 Multiplier outputs are brought into the computer with a 16 bit analog to digital converter. the analog to digital converter module is also an IndustryPack module, this one from Acromag, the IP330. The IP330 allows the automatic scanning of selected channels to be triggered by a signal, or a timer. This method will be used to ensure all samples are sampled at a consistent interval. A data sheet on this module is included in the Appendix.

4 New control system software

4.1 GBI software design

4.1.1 Hardware device managers

In the language of the GBT M&C system, a device Manager is a piece of software that provides at control interface to a particular piece of hardware. These device managers are shown in Figure 4 and are described below.

Antenna manager The 85-foot polar mounted antennas will be controlled through a standard MCB interface card. The software task will monitor en-

Table 1: MV-167 Characteristics

68040 CPU, 33 MHz
8 MB RAM
SCSI Interface
10 Mbit/s Ethernet Interface
4 Serial Ports
1 Parallel Port
4 Counter/Timers
Demonstrated MTBF:
Mean 147,507 hours (16.8 years)
90% Confidence 85,522 hours(9.8 years)

coders, limit switches, brake status, and feed focus and rotation. It will command the position, brakes, and feed position. It will take the commanded coordinates and implement the pointing servo loop. It will provide tracking status to other programs. Monitor and status data will be passed along to a logging task. Alarm conditions will be checked for, and alarm messages sent out.

Receiver manager Each receiver will have a monitor/control task which will collect monitor data and set receiver modes. The receivers have standard MCB interfaces. Monitor data includes things like cryo temperatures, gate voltages, and lock levels. These will be passed along to the logging task. Error conditions will generate alarm messages. Control functions are provided to turn the noise cals or LOs on and off.

Backend manager There will be an interface board which will provide delay values to the Digital Delay Rack. Another digital interface will provide gain settings and zero-offset signals to the variable gain and auto-zero stages. A software module will send commands to these devices through the VME bus. The module will calculate predicted interferometer delays to be delivered to the DDR. Timing is critical. The delays have to be sent to the DDR at predetermined times within about 1 ms tolerance. In the present system they are sent at the beginning of every even numbered second.

An A/D will sample the data from the multipliers at a 50 or 100 Hz rate. The data consist of sixteen 16-bit numbers for each sample (8 multiplier outputs and 8 total power outputs). Data from the A/D needs to be stored in memory in an efficient way, perhaps using a double-buffering or circular-buffering scheme. Data will accumulate in the buffer for a predetermined interval such as 10, 20, 30, or 60 seconds. For each accumulation interval, a fringe pattern is fit and the

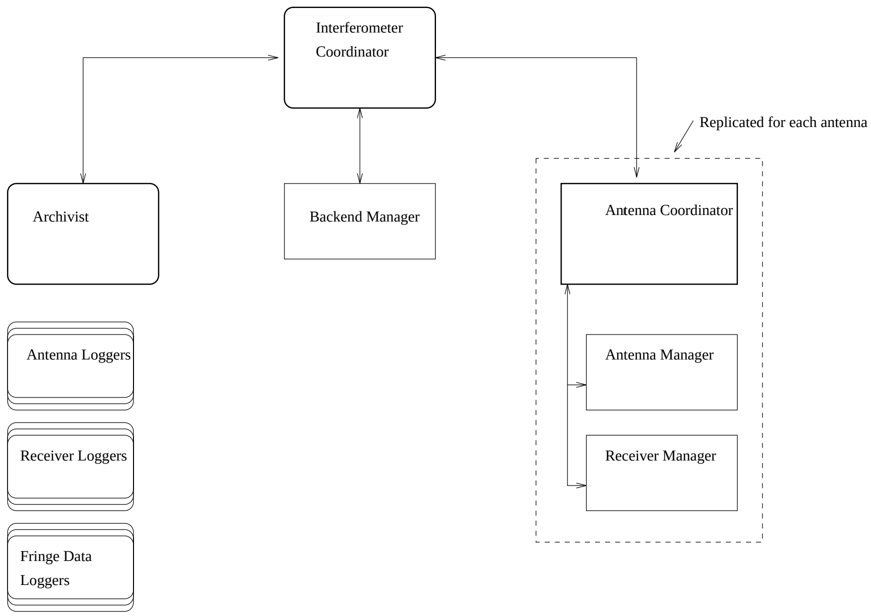

Figure 4: Software Block Diagram

resulting amplitude and phase stored in a file. The time at which each datum is acquired needs to be known to about 1 ms, and stored with each output record.

If a new correlator is built, the sample rate may increase, and the number of data points per sample will increase to over 100. The analog data acquisition system will be replaced with a digital interface, probably directly through the VME bus, or a similar interface.

4.1.2 Required software modules

This section describes the first-level breakdown of software functions required of the system. These modules may be implemented in many different ways, and this is not meant to constrain the design in any way.

General considerations The system should allow "n" telescopes, which may be configured as "m" sub-arrays. Each sub- array can follow a separate observing program. With the present system this arrangement would allow for two of the 85-foot telescopes to be used as an interferometer, while the third runs a different observing program as a single dish. In the future, a new correlator may allow two or more interferometer sub-arrays including the 85s and other telescopes on site.

The number of telescopes and sub-arrays will be specified in a configuration file that the program will read on startup. i.e., the system will have no idea how many telescopes and arrays there are until it reads the configuration file.

The system should be insensitive to parts of itself malfunctioning or missing. For example, if one antenna is off line, the rest of the system should continue to operate as if everything were just fine. This is important for testing individual parts of the system in the absence of others. It is also important to keep observing even when some error conditions exist. Of course the errors can cause alarm messages if necessary. But only the human operator should decide to shut down part or all of the system.

Many of the software modules will implement both a "manual" mode and an "automatic" mode. In the automatic mode, everything is driven by an observation file. In the manual mode, an operator can control the position of an individual antenna, and can do such things as setting telescope motors individually, doing system temperature calibrations, or doing a pointing measurement.

Initialization All tasks read configuration files when they start up. Initially a "master" task starts up and reads the array configuration file. An observing control task is spawned for each sub-array. An antenna control task and a receiver control task is spawned for each telescope. When an antenna control task starts up, it reads the appropriate antenna-specific information from a file. This includes the device IDs, bus addresses, and pointing model parameters. Likewise, other tasks read parameters specific to their functions from configuration files. There should be a provision to change any configuration file and tell the system to re-initialize, without having to stop and restart everything.

Observing control This program follows an observing task file which contains a list of names of observation control files and the start and stop time and date for each. The Observation Control File contains a list of sources to be observed, their position, observing mode, integration time, stop time, etc. The observing parameters are passed to "procedure control". Then the program waits for the next source change.

Procedure control Given the desired parameters for the current observation, this implements a few standard observing procedures: "track" - move to a sky position and track a source; "move" - move to a given HA and Dec, and stop; "cal" - do a system temperature calibration; "point" - scan across the source position and determine a pointing offset. The desired sky positions are passed to the position demon.

Time daemon This knows the date and UT time, and calculates sidereal time. These values are kept current and are provided to whatever modules need them.

Positions daemon This module is responsible for converting positions passed from the procedure control module to position of date. For planets or comets, the position is kept updated often enough for the required accuracy. Pointing model corrections are applied. It provides the current commanded positions to the antenna controllers.

Data acquisition and fringe Fitting The system will still acquire data from the DDR through an analog to digital converter in the VME computer. The IP330 module provides 16 16 bit differential inputs that are tied to 8 multiplier outputs and 8 total power outputs. These inputs are accumulated for a predetermined interval, then fringes are fit and the amplitude and phase are stored in a file. The time that the data was acquired will be stored with the data.

Data writing and flagging The fringe data is written to disk along with the time of observation and data flags. The flags indicate any error conditions that may pertain to each data record, such as an antenna off source, a warm receiver, bad fringe fit, etc. The data file will be an aips++ compatible interferometer fringe data format file.

Loggers Antenna and receiver status information is logged. There is a short term log that can be examined on line in which status is shown at some short interval, such as every 10 seconds, 30 seconds or such. About an hour's worth of short-term log is kept on the system. Older data is dropped. The long-term log is written at longer intervals such as every 30 minutes, and goes to a FITS log file. Some data can be logged synchronously with observing scans, such as tracking errors. The logger module reads a configuration file that gives the logging intervals for the various parameters.

This logging system will likely be implemented using the standard GBT logger, which writes binary FITS files. The decision of short versus long term logs is one of disk space. The short-term logs can be written to a scratch space on disk which is erased by a cron job each day. In the event of difficulty, the logs could be accumulated and analysed for errors.

Alarms When any module notes an alarm condition, a message is sent to this module. The alarm message is logged. The operator is alerted. In some cases a telephone call-out is triggered. When the alarm module starts up, it reads a configuration file that gives all the alarm thresholds and criteria for triggering a call-out.

4.2 Reused GBT M&C software

Much of the new control system will be built of existing M&C software modules. The following list of subsystems will be used in their entirety. A few extensions will be made for the specific requirements of the GBI, and a few areas are not yet fully implemented, such as fine control of the message system (alarms).

4.2.1 mcTime

The mcTime subsystem synchronizes the computer clock with the site time system. The synchronization is carried out using IRIG-B on the GBT. For the GBI, the thought is to modify the daemon to use the ntp protocol with a 1 PPS tick disciplining the internal computer clock. This combination should give better than 500 microseconds accuracy of the clock. By contrast, the IRIG-B system can provied synchronization to some 10's of microseconds.

4.2.2 Message system

The Message system will be used unchanged to provide the alarm processing for the GBI. The GBT Message processing module is not yet finished, and so a module to process the alarms to provide call-outs is needed.

4.2.3 Watchdog system

A system is in place on the GBT Single-board computers to monitor the health of the SBC and the tasks running on it. The Watchdog system can be adopted without change for the GBI.

4.2.4 TaskMaster system

TaskMaster is a daemon that runs on the Unix systems that monitors the health of all the telescope control and data acquisition processes running on that computer. If any of these are killed, then the daemon restarts it and mails the "owner" of the process, informing the ofwner of the demise of the process, and the reason why, if it can be determined.

4.2.5 Monitor system

The Monitor system is the basic way in which periodically sampled data gets out of the system. It is sent across the network already time-tagged, so that data can be collected and post-processed easily from any workstations. The monitor system supplies data to both loggers and user interface programs. This subsystem will be used unchanged from the GBT incarnation.

4.2.6 Data logger system

The data logger program for the GBT is a generic program that can log samples from any data point on the telescope into a fits file. The same logger program will be used for the GBI, unchanged.

4.2.7 User interfaces

The GBI can make use of any of the GBT interfaces. The command-line Glish interface, the GUI Glish/Tk system, or the GUI tcl/Tk system are all transparently supported. Others are possible as well. (Perl and Python come to mind)

4.2.8 Other libraries

The rest of the generic M&C libraries are used in various capacities. Recently, an effort has begun to separate out the generic control system components from the GBT specific ones. This has not yet been accomplished, but a start has been made, and seems straightforward. Here is a listing of the generic libraries developed for M&C and thier descriptions. More information on these may be found at http://info.gb.nrao.edu/GBT/MC/doc/GBTsoft.html.

ConfigIO handles reading and writing configuration files in a standard manner.

Control is the mother of all libraries. It contains the manager classes.

Coords handles coordinate transformations.

DataDesc is a library for accessing internal manager information.

DeviceAccess works with the above DataDesc library.

DynamicLoader is a library for loading shared objects at run-time.

FSM is a finite state machine library.

FitsIO encapsulates Pence's CFITSIO library to handle telescope data.

GServer is a Glish RPC server

IF provides support for calculating the frequency characteristics of an IF channel.

IFDevice

MCBInterface provides packaged classes for acessing words or parts of words on the MCB.

Matrix is a matrix math package.

MesgMuxIF is the interface to the system message multiplexer.

MesgProc contains all of the container classes for Messages.

Message implements the application or event-generating side of the Message system.

PVA is a base class to encapsulate Position/Velocity/Acceleration and duration.

RPC++ is a C++ Interface for remote procedure calls, including a system interface for the select system call.

Scan provides a framework for specification of arbitrarily // complex movements of a mirror.

ScanSpec is used to sequence through XXXSegment lists, in real-time.

Sequencer flushes system defined vxWorks semaphores at their defined interval.

TaskLib implements vxWorks task management calls on Solaris

TimeKeeper is used by the coordinate transformation libraries to keep track of local time.

TimeStamp is a class for producing and manipulating Time reference tags or TimeStamps.

TimeStampUtl contains a routine to set the clock on a Unix machine.

util is a module with utility functions and classes for getting the environment, parsing text, etc.

5 Conclusion

The new GB Interferometer control system will be a good test of the flexibility of the system designed for the GBT. The antennas are much different, the user requirements are much different, and the system will be run unattended much of the time. The system will be put together over just a few months this summer by a summer student and Frank Ghigo, with help from the GBT M&C group, mostly John Ford and Mark Clark. The actual move of the control system will probably take place later this year, before work on the GBT outfitting starts in earnest. There is also the possibility that a new correlator will be built for the system, which would have better reliability and performance than the digital delay line currently used. This correlator may be built by a German group from a design concpt by Rich Lacasse.