Penn Array Receiver

Penn Array Receiver CDR Document 3: Software for the Penn Array

Version: 3
Date: 08oct03
Authors: Brian S. Mason
Keys:

Revision History

Ver.	Changes	Date	Author
0	First Draft	16 Sep 2003	Brian Mason
1	RP, NR comments	23 Sep 2003	Brian Mason
2	Add milestone list, minor revisions	25sep 2003	Brian Mason
3	Final typo fix	08 oct 2003	Brian Mason

Contents

Abstract

We present a status report on software for the Penn Array. We also outline our plan for delivering the software needed to commission the instrument, and indicate directions for the user software which will be needed after commissioning.

1 Introduction

Software, while not often technically risky, is a critical path item for many astronomical instrumentation projects. At the beginning of the Penn Array project the software resources necessary for the successfull completion of the project were not available. Consequently NRAO assumed responsibility for the software necessary for scientific operation of the Penn Array on the GBT, including the analysis software, and deferred the identification of software resources. Progress has recently started on this front. We present a status report and a plan for further action.

2 Laboratory Integration & Testing

The primary requirements for the software employed in laboratory operations with the receiver are convenience and flexibility. UPenn has employed standard packages such as LabView for cryogenics development and will continue to use this and other packages of their choice. This is likely to include the use of NASA's Java-based Instrument Remote Control (IRC) software for readout of the detectors in the lab.

Acceptance testing will be performed with software configuration of Penn's choice. Eventually responsibility for laboratory maintenance of the receiver will be assumed by Green Bank. Well before acceptance testing we as a collaboration will decide whether these procedures will be performed using Penn's laboratory software and equipment, or using the YGOR interface. If the former, the software Penn uses for lab work will be considered a deliverable.

3 Monitor & Control Software

3.1 The GBT Monitor & Control Architecture

This section briefly describes the existing GBT Control System, with an emphasis on describing how observations are executed. In GBT terminology an **observation** (for example a raster map of a source) is composed as a sequence of one or more **scans** (for example, a single raster row of a map). Both of these concepts are elaborated in more detail below.

The GBT observing system employs a highly modular, distributed architecture. From an observing (as opposed to engineering) viewpoint, the system is divided into four major components: Configuration & Observation Control; Scan & Device Control; and Data Monitoring and Analysis. Figure 1 illustrates the GBT software architecture (or at least a selection of components which are relevant to the Penn Array).

Scan and device control is provided by **Ygor** - the generic portion of the GBT monitor and control software. This provides the facilities for the four basic functions of control, monitoring, message/alarm handling and data production. Ygor is a telescope-independent, fully distributed, object-oriented system. Control functionality is provided by a base class, the **manager**, which provides a common control interface and which implements a core set of functionality required by all devices. A derived class, a **Device Manager** is used to add the functionality required for a specific device (e.g. a receiver, a backend or the antenna).

The core (and only!) observing primitive provided by Ygor is the **scan**. A scan is a contigous period of data collection defined by a finite set of variables. In advance of a scan, all participating devices (Managers) are configured by setting their control parameters as required. Managers are state machines: the state sequence for

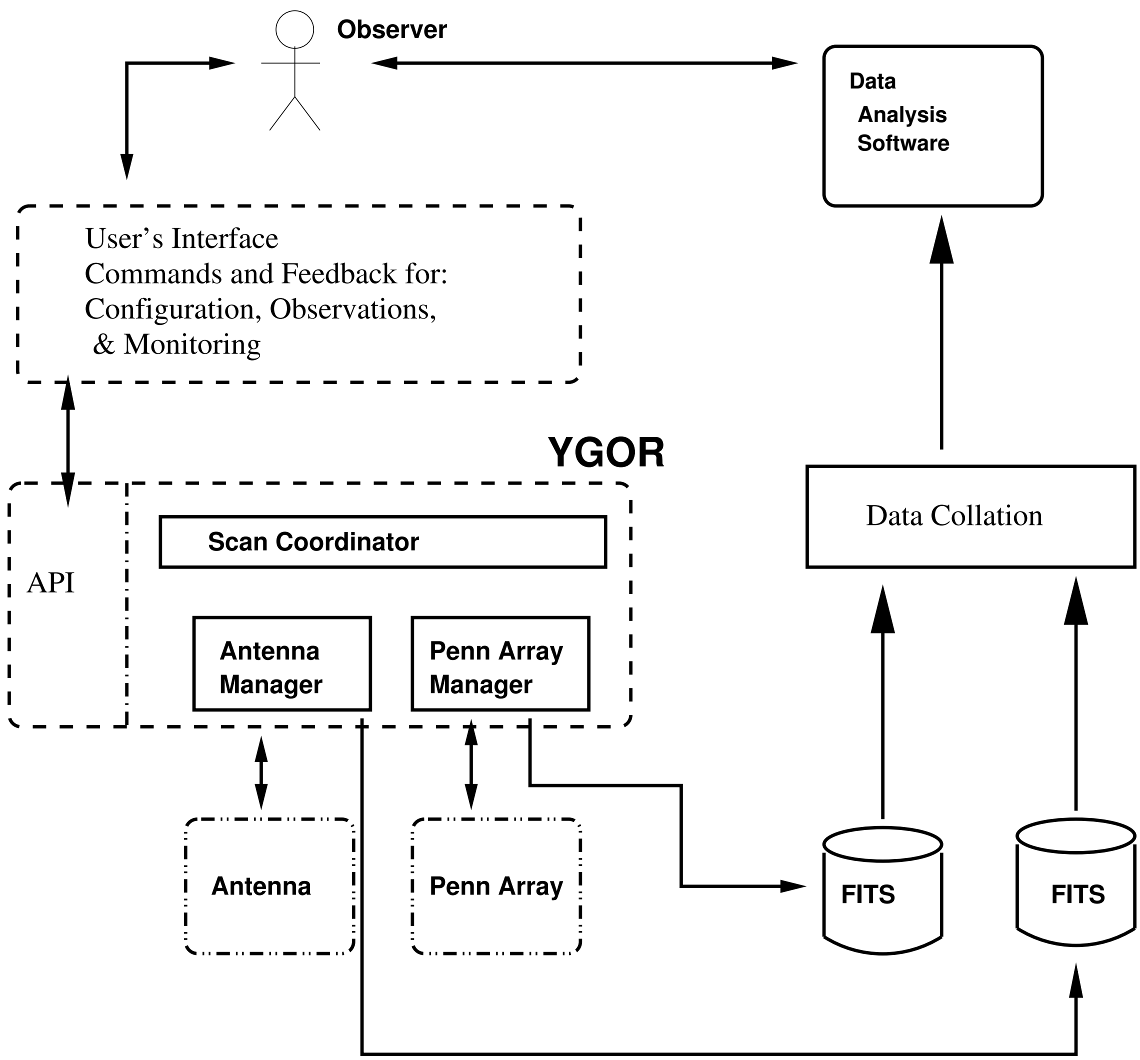

Figure 1: Schematic view of the GBT control system architecture.

a successful scan is Ready, Activating, Committed, Running (collecting data), Stopping, Ready. A co-ordinating Manager is able to synchronize the start of a scan across its "children" by requesting from each of them their Earliest Guaranteed Start Time (EGST). The co-ordinating Manager initiates the scan by using the EGST as the commanded start time for its children [1].

At the root of the Manager control hierarchy is the Scan Coordinator. As its name suggests, the Scan Coordinator is a co-ordinating manager specifically intended to co-ordinate scans within the context of a GBT Observation. It may be configured to include or remove specific Device Managers as appropriate into the scan execution; thus for example to change receivers, a new receiver Manager is included in the Scan Coordinator sub-system selection, and the previous one removed. The Scan Coordinator also has a number of key parameters (e.g. observing frequency, position); once set in the Scan Coordinator, these are automatically passed down to all its selected children.

The intention of the Ygor layer is to expose the full functionality of the underlying hardware to the higher level software, in a complete and consistent manner. At the Ygor layer, the system knows very little about astronomy, or indeed about performing observations. This additional knowledge is provided in the GBT observing system by the Configuration and Observation Control capabilities currently implemented in the "GO" (GBT Observe) package.

An observation at the GBT is defined as a sequence of one or more scans executed through a GO observing procedure. All the usual types of observation (point source observations with various switching schemes, grid and raster maps, calibration observations, etc) are supported. Some types of observations comprise multiple scans. In this case the system sequences these scans automatically.

An observation is executed as follows. First, the user configures the overall system as required. For the Penn Array this will entail: selecting the Penn Array, tuning the TES's, properly biasing the bolometers, setting the integration period, configuring the calibration signal, selecting a filterwheel, and ensuring the active surface is on. The user also selects the parameters of the observing procedure: for instance, central coordinates and map size.

Once the user presses the "start" button for the observation, the observing procedure takes control. The procedure sends additional configuration informationto the selected devices as required (in this case, the antenna position scan segment which defines the first scan), and commands the Scan Coordinator to start the scan. The Scan Coordinator negotiates the earliest guaranteed start time between its children (in this case, almost certainly determined by the antenna, which needs to slew to position), and instructs all selected Managers to start the scan at that time. From this point until the end of the scan, each Manager executes independently, synchronized by absolute time and the hardware switching signals. During this time, the backend Manager collects data, and writes it to a FITS file. Each manager writes one FITS file to disk per scan. Once the first scan is complete, the observing procedure automatically configures the selected devices for the second scan, and the process repeats itself.

In summary, the key points of the system are as follows. The system is extremely modular. A new device is added by providing a Device Manager; the common control interface ensures that all devices present a uniform interface to the higher level system. The core building block for any type of observation is a scan; co-ordination of the devices selected to participate in a scan is performed by the Scan Co-ordinator. Precise timing synchronization is acheived by distributing to each device IRIG or NTP, together with a 1 PPS signal. Observations are performed via observing procedures, which alternately (re)-configure devices and execute scans as required The result is a set of FITS file in the GBT archive, one FITS file per device per scan.

3.2 M & C Software for the Penn Array

The software infrastructure described in the preceeding section is already in place and in regular use for GBT observing. Several new items or capabilities will be needed:

- A device manager for the Penn Array
- TES/SQUID Configuration and tuning software

[1] Time is synchronized across devices by an IRIG timing signal and a 1 PPS hardware signal, or via NTP plus 1 PPS

- Support for new scan patterns
- Multi-feed pointing and focus procedures

These are described in the subsequent sections. See also the attached Gantt chart.

We considered using NASA's IRC software for array readout and control. This is a "pre-packaged" solution; active development in support of TES arrays within the IRC framework are underway at Goddard. However IRC software seems intended to provide much of the functionality of the existing YGOR infrastructure. We found it difficult to identify TES-specific capabilities of IRC and separate them from the general infrastructure IRC provides, and to identify a well-defined software interface to IRC which might suit YGOR. Rather than support a complete and independent system architecture specifically for the Penn Array, or supporting a comparatively functionless layer of software between YGOR and the array, we decided: a) to use the GBT YGOR infrastructure, and b) that the Penn Array YGOR device manager should communicate directly with the instrument at the device driver level.

3.2.1 A Device Manager for the Penn Array

The key function of a device manager is to present interfaces to both the YGOR system and the device (in this case the Penn Array). The behavior of YGOR device managers and their YGOR-side interfaces are standardized, so most of the work lies in defining interfaces to the device itself. As stated previously, these interfaces will be at the device driver level.

Control and readout of the array will be accomplished by a single commercial PC rackmounted in an RFI-shielding enclosure in the receiver room. It will have a fiber network connection and it is likely that the device manager will run on this PC. The PC provides all control and readout functionality for the Penn Array. There are several device-driver level interfaces which the Penn Array presents:

- **Array Readout and Control**: Configuration of the multiplexers, readout of array data, and control of the bolometers will be accomplished through a) a serial connection; and b) a custom (NIST-provided) fiber readout card. These interfaces are described in detail in the Detailed Design Documentation, although the DAQ interface document is only a working draft.
- **Cryogenics Readout and Control**: All housekeeping will be performed over a GPIB bus. Several devices (an HP DAQ box; a Lakeshore cryogenics readout unit; and a programmable power supply) will be daisy chained on this bus. All required functionality is low-bandwidth ($< 1\,Hz$). The GPIB interfaces are documented in the Detailed Design Documentation. These documents also specifies the operational procedures which will be needed, *e.g.*, how to cycle the He-7 fridge. The cryogenics and housekeeping interface can be regarded as final.
- **Calibration pulse control**: The calibration diode power supply will be controlled by a switchable relay over a serial line. Serial control will allow the cal pulse to be accurately registered with the bolometer timestream. This is a straightforward interface which GB has specified.

Refer to the Supplementary Material for block diagrams of the system as a whole.

Two main interface issues need more work. The first is that while we have a substantial amount of documentation on the DAQ interface, the Green Bank software group has some unanswered questions at the level that the DAQ interface document doesn't specify an interface to which they could write code now. NRAO is working with Penn and GSFC to answer these questions. The firmware on the custom fiber card is also being upgraded at present (see below). Pending the resolution of these questions, and documenting the improved capabilities of the fiber card, the DAQ interface is provisional. We plan to revisit and finalize the DAQ interface in the first and second quarters of 2004. There will soon be an operational MUX in the lab at Penn and this will be of great assistance in becoming familiar with the interface and filling in the gaps in our knowledge. This work will be done by NRAO and Penn personnell in consultation with GSFC.

Second, we do not have a concrete schema to timestamp the bolometer datastream with millisecond accuracy. Two options present themselves. We may implement all of the device drivers under RealTime Linux, with which Green Bank Software personnel have extensive experience. NRAO has achieved sub-millisecond accuracy with the combination of NTP and a 1 pulse-per-second TTL signal on other devices, so setting the PC clock to this accuracy will not be a problem, and RTLinux will enable interrupts to be serviced and timestamped quickly. Alternately we may be able to use an upgrade to the firmware on the custom fiber card which is now being designed. This upgrade will allow the data to be timestamped given acces to an external time reference. We will resolve the timestamping issue when we finalize the DAQ interface in Q1 and Q2 of 2004.

The Penn Array manager together with other managers such as the Antenna manager or the active surface manager provide control and readout of all instrumental degrees of freedom and monitor points. The following subsections (§ 3.2.2 through 3.2.5) describe the higher level software which will be needed for operations on the GBT.

3.2.2 TES/SQUID Configuration & Tuning

The device manager itself will only control the basic instrumental degrees of freedom. Special operational procedures are needed to operate SQUID-multiplexed TES arrays, and these will be executed by higher-level software which communicates with the device manger via YGOR. The primary procedures which are needed are setting the SQUID biases and the bolometer biases. GSFC has experience with these procedures and is implementing software to automate them. Lab experience with the trial and production detector arrays will be essential in specifying these procedures, and hence what realtime software is necessary for operations. An important consideration is flexibility, as we will likely need to fine-tune the operating procedures as commissioning proceeds.

Green Bank will implement these procedures at a level higher than the manager level within the GBT control system. While at odds with the traditional approach of including most or all device-specific functionality within the manager, this allows the needed flexibility. Fine-tuing these higher-level procedures (possibly over the internet) could begin as soon as a) the penn array, or the required components are integrated in the lab; and b) the device manager is complete.

3.2.3 New Scan Patterns & Antenna Modes

The Penn Array will primarily be used in variants of on-the-fly mapping. In order to eliminate $1/f$ and atmosphere noise while satisfying the scientific requirements of a range of observing projects, we will need scanning patterns which are not currently implemented by the GBT. These could include "daisy" patterns ,lissajous figures, and fourier-truncated billiard-ball patterns. As a conservative baseline scenario we assume *only* the capabilities of the primary— there is no plan to move the secondary in support of exotic observing modes[2].

We would also like to implement a truly "on-the-fly" antenna scanning mode which reduces antenna-related scan-start overheads. Currently the antenna is given a position, velocity, and acceleration which it must have at the start of a scan, and if it needs to it backs up to get a "running start" so that it can achieve this given acceleration limits and its present position. While useful for some types of observations this is not necessary for on-the-fly maps. It would be preferable to immediately begin moving towards the desired target, start the scan when the antenna was within some distance of the desired point on the sky regardless of velocity or acceleration. This change should *add* to the capabilities of the GBT and preserve existing interfaces.

We anticipate that this work will involve Don Wells, Brian Mason, and one individual from the Green Bank software division. Prior to tests on the antenna, Wells & Mason will prototype scan strategies with the simulation pipeline. These antenna mode tests could be done at present with existing receivers and the GBT continuum back-end together with AIPS++ continuum mapping analysis algorithms; we could also use the mapping algorithms under development for the Penn Array (§ 4) or the existing IDL pipeline. After testing these capabilities will need to be written into the GBT observing system.

[2]Note that currently the antenna manager delivers positions assuming the secondary is in the correct position. As long as the focus has been set, say by observing a source, this is fine; however were the secondary dislocated the positions would be wrong. PTCS may eventually address this shortcoming.

3.2.4 Pointing & Focus

High-frequency observations with the GBT will require periodic pointing and focus checks. These observations require a level of coordination not usually provided by analysis software: the telescope control system must signal when a pointing observation has been completed; the software must fetch the data; and the derived pointing offsets must be passed back to the control system.

Current GBT pointing data is analyzed in AIPS++ (via a facility known as IARDS). In principle we could use IARDS, however the AIPS++ pointing algorithms are essentially for single-beam (or dual switched-beam) receivers. Penn graduate student Michelle Caler has been working on pointing algorithms which make better use of the Penn Array's large field of view and full sky sampling. Prototypes of these algorithms are working with the IDL-simulated data. We will continue this development with the aim of fully specifying a pointing algorithm by the end of 2003.

The Green Bank software division is working on a Python framework for pointing and focus determinations. We intend to implement the algorithms in this framework.

3.2.5 Realtime Data Monitoring

Observers' inspection of the data falls into two broad classes: basic instrument health (does the dewar still have vacuum?), and data quality checks. The GBT system provides the ability to inspect instrument monitor points (vacuum gauge, selected data channels, voltage monitor points, etc.), for example, to plot them versus time.

It will be important to quantify how much hold time remains for each stage of the Helium-7 system. Penn will develop algorithms to estimate this based on laboratory measurements; Green Bank will implement this algorithm. This could be done within the manager or at a higher level. It will probably require that the GBT elevation be monitored even when the Penn Array is not being used. A full recycling may not always be needed, and Penn will provide guidelines for which procedures are needed when.

The best check on data quality is to reduce the data, e.g., make a map. The data analysis software (§ 4) will be used for this. The first step is to bring together the information from the various device FITS files (for the Penn Array the primary data will be distributed between one FITS file per scan for the antenna, and one for each scan for the Array itself). NRAO has tools to do this which will need minor modifications to accomadate Penn Array data.

3.2.6 Dynamic Scheduling and Site Assessment

Green Bank intends to provide these capabilities, although they will not be fully developed on the timescale of Penn Array commissioning. These capabilities are beyond the scope of the Penn Array project; however the remaining hold time of the He-7 system (§ 3.2.5) will be an important input to dynamic scheduling decisions, and the project is responsible for providing an estimate of this quantity.

4 Data Analysis

Three considerations drive our approach to the data analysis and analysis software. First the GBT has no tertiary mirror, and the primary and secondary drive motors are incapable of chopping at faster than $\sim 0.3\,\mathrm{Hz}$. This is inadequate to remove atmospheric fluctations. Since large arrays tend to be well-suited for wide-field imaging, and chopping reconstruction techniques limit the field of view to several chop widths, this is not a significant drawback; however alternate techniques to remove the atmosphere are needed. Second, many of the interesting problems require high dynamic range hence accurate relative calibration and removal of systematics. We aim to achieve the required relative calibration accuracy by using both the atmosphere signal itself, and an internal noise diode calibrator, as "flats" to determine relative detector gains; we also aim to use the instantaneous sky sampling offered by the array to distinguish between astronomical, atmospheric, and other signals. Finally the SQUID-multiplexed TES arrays are a new technology, and 3mm operations on the GBT will initially be in a prototype phase. We can expect to identify unexpected effects in the data, and our pipeline will require some flexibility until commissioning is finished.

Array detectors offer an advantage since the sky is highly correlated across the field of view, especially for a large aperture telescope such as the GBT, and changes fairly slowly. We plan to operate the telescope primarily in an on-the-fly mapping mode. This will modulate the astronomical signals on timescales more rapid than the instrument or atmosphere change, and most astronomical signals will appear in individual beams rather than coherently across the array. This strategy also reduces the flatfielding requirement by sampling each sky pixel with many bolometers. Since the Penn Array is fully sampled, on-the-fly mapping is reasonably efficient even for photometry of a known source. In this case one would slew less rapidly and over a smaller field of view. The on-the-fly technique applied to fully sampled arrays also relaxes pointing requirements. While pointing is formally beyond the scope of the Penn Array project and Green Bank aims to deliver full 3mm capability, it is as a practical matter helpful to have an extra margin.

The main steps which will need to be done in the simplest analysis are:

1. **Deconvolve** the bolometer time response if necessary (depends on the slew speed);
2. **Identify cosmic ray hits** and remove them;
3. **Fit for at atmosphere term** across the array— for example a low order polynomial in two spatial dimensions;
4. **Use the atmosphere fit to calibrate detector gains**;
5. **Subtract the atmosphere fit**;
6. **Apply cal diode scaling** to the residuals, correcting for changes in the bolometer response with time and referencing the data to appropriate units;
7. **Grid the data onto the sky**, making a final astronomical image.

We currently have an IDL-based pipeline which performs all of these steps except for the cal diode scaling. This pipeline was developed by Ed Chapin and David Hughes (INAOE) for BLAST data. It has been adapted to suit the expected properties of Penn Array data from the GBT, and includes a simulator with a realistic atmosphere model and basic instrument properties (detector time response; photon noise; $1/f$ noise; detector-to-detector gain variations). An advantage of the Chapin/Hughes IDL pipeline is that it is quite easy to modify. We are confident that we could use it to analyze commissioning data from the Penn Array very soon were that needed. This would be sufficient to calibrate the data, image bright sources, construct beammaps, and estimate noise levels. We have not demonstrated that this pipeline removes the atmosphere and other artifacts to the level needed for much real science, i.e., that the noise will continue to integrate down for many minutes in the final maps. Since this pipeline makes only minimal use of the characteristic differences of the astronomical and atmospheric signals it seems likely that better methods are available.

We are therefore exploring two alternate classes of algoritms:

- **Iterative Imaging**: this is similar in spirit to the existing IDL pipeline, except that the final imaging step (# 7) is then used to construct an explicit model of the sky is then subtracted from the data. We are exploring a variant of the CLEAN algorithm to estimate the sky model, although other approaches such as maximum entropy could also be used. The residuals are then passed back to the atmospheric fit (step # 3) and the analysis proceeds. This loop is repeated until satisfactory convergence, and should improve the separation of astronomical and systematic signals (i.e. image artifacts). A CLEAN-type approach to constructing the sky model also has the advantage of naturally including a beam deconvolution. Since the Penn Array's illumination of the primary has a sharp cutoff, this may be helpful for imaging crowded fields of point soruces. Bill Cotton (NRAO) has implemented a prototype Iterative pipeline for the Penn Array.

- **Linear Inversion Imaging**: In recent years the Microwave Background community has invested tremendous effort in solving the problem of making wide-field images from bolometer data, and some of these methods will be applicable to the Penn Array (for imaging deep high-z galaxy fields, or SZ clusters). This class of approaches typically requires a large matrix inversion and so is computationally expensive, but in principle makes the best use of all available information to separate the astronomy from systematics and instrumental effects. Since typical GBT observers will not need the covariance matrix of pixels on the sky— a driving consideration for the CMB problem— a number of shortcuts are available, and we expect that this approach will be feasible and useful for a range of interesting problems. In particular a full matrix inversion is not necessary as indirect methods may be employed. Dale Fixsen and collaborators at GSFC have implemented a large-scale linear inversion algorithm for SIRTF data. The Fixsen code has been applied to $350\ \mu$m on-the-fly data from SHARC-II with good results. Their algorithm runs on a desktop PC in a few minutes for modest datasets.

All three of these algorithms will be tested with simulated data.

Our development plan is as follows.

1. **Simulate science cases**: These will be a selected set of projects which embody different scientific requirements: say, mapping a star forming core, a deep high-z galaxy field, and an SZ cluster. The Chapin/Hughes IDL simulations will be used to generate a set of canonical datasets; these datasets will be used (below) to compare algorithms. The simulations will, as closely as possible, mimic the properties of a real observation. They could for instance comprise a calibrator scan, followed by twenty minutes of mapping, etc. We will also use these simulations to explore the utility of different scan patterns. Brian Mason (NRAO) and graduate student Michelle Caler (UPenn) will be responsible for this work.
2. Continue development of the Iterative Imaging (**SDclean**) package, with the goal of using it on first-season prototype science data. Bill Cotton (NRAO) will do this work.
3. Prototype **Linear Inversion Imaging** on simulation data. We hope to use the Fixsen package as a starting point. Since the inversion algorithms are more difficult to modify, we will postpone much of this work until after astronomical commissioning when we know what the main instrumental effects to be dealt with are. This work could involve contributions from Brian Mason, Don Wells, and Michelle Caler.
4. **Test algorithms with simulated data** to assess their merits.
5. **Commission the Instrument** using a modified IDL pipeline, together with the Cotton SDClean package.
6. **Produce a user package:** While all of our software will be available to the astronomical community, it may not be particularly user friendly. Since a general announcement to the user community will not be made until after preliminary commissioning is completed, and the IDL and SDclean packages will be available in any case, this should not be a problem. After commissioning we will consider how to provide easy-to-use software for the user community. This approach also allows us to factor unexpected properties of the bolometer data into the analysis software.

The accompanying Gantt chart shows our analysis software development plan.

A concern is that one of the key players in this plan (Bill Cotton) has no official commitment to the project, contributing through a small fraction of his research time. We are exploring obtaining an official allocation of his effort.

5 Plan

Major milestones in (and related to) the software development plan are:

- **New Antenna Mode Requests Documented (19dec03)**: the scheduling of this work package is quite flexible; the work and subsequent testing could be done in collaboration with the PTCS group.

- **Simulated Datasets (15jan04)**
- *MUX Operational in Lab (01feb04)*
- **Memo Specifying Calibration Strategy Written (21apr04)**
- **DAQ Interface Finalized and Documented (07may04)**: this includes resolving OS and timestamping issues
- **FITS file specification memo (04jun04)**
- *Full Receiver Integration at Penn (14jun04)*
- **Iterative Imaging analysis of simulated datasets (09jul04)** including assessment of noise levels.
- **YGOR Manager Complete (06aug04)**: at this point, in principle, control of the receiver will be possible via YGOR over the network, so some tests could proceed remotely.
- **Linear Inversion Imaging of a subset of Science Cases (03sep03)**
- **High Level Software Ready for Commissioning (22oct04)**

Refer to the Gantt chart for more details.