Hands-on Observing Project:
Continuum Maps with the GBT
06jul03 Brian Mason

1 Introduction

The goal of the project is to map a large region ($\sim 5^\circ \times 1\frac{1}{2}^\circ$) of the Galactic Plane at L-band (1.4 GHz)with the GBT and choose a region for follow up observations at X-band (9 GHz). The X-band map will be smaller but of higher resolution; you should be able to make a followup map of about $40' \times 40'$ at X-band. The expected noise levels in each map are a few milliJanskys, which is sensitive enough to detect objects such as Supernova Remnants or HII regions associated with massive stars, both of which are plentiful in the Galactic Plane. The initial (L-band) map should be centered at Galactic longitude $\ell = 84^\circ 30'$ and Galactic latitude $b = 0$ (*i.e.* the Galactic Plane). The initial map should be "long" in the longitude direction, and skinnier in the latitude direction, since our goal is to map the Galactic Plane.

2 Observing Setup

You will need several tools, and I'd suggest setting these up on separate desktops (you have 8 virtual desktops). The tools, and how you'll start them, are:

1. The Map Calculator: located at `fghigo/bin/mapcalc`, this will help you establish the parameters for your map and predict its sensitivity.
2. GO: The GBT Observer's interface. This can be started just by typing `go`. You will use this to execute the maps.
3. The Config GUI: This provides a high-level interface to set up the system for observing. Start it by typing `CONFIGGUI`.
4. CLEO: This provides the lowest-level interface to the system, and we will use it to verify that CONFIGGUI has set up everything correctly as well as apply a few tweaks if needed. Start it by typing `cleo`.
5. IARDS: This is an AIPS++ based realtime display. We'll mostly use it to check the results of our pointing and calibration data. start it by typing `IARDS`.
6. AIPS++ will be used to analyze the map data as we get it to make sure that things are going ok. This online reduction will also enable us to choose an interesting target for X-band. Start it by typing `aips++`. You may need to first source the setup file by typing

   `source /home/aips++/stable/aipsinit.csh`.

3 The Observing Procedure

For each map you should do the following:

1. Establish your map parameters in `mapcalc`. The key numbers you need to actually obtain the map through `GO` are: the slew speed (Xrate), the row spacing (one pixel); you should also note the number of rows needed, the time for one row ("stripe") and the total time required for the map. Use an integration time ("On") of 0.6 sec and zero for "Off".

2. Use `CONFIGGUI` to configure the system. Choose the receiver, rest frequency (1420 MHz or 9000 MHz), bandwidth (20 MHz or 80 MHz), switch mode (total power), switch period and Obstype (continuum). Defaults for the others should be fine

3. Double check the setup in `CLEO` (devices to check: receiver, IF rack, & DCR). In most of these we are checking for red boxes. We may need to balance the IF's in the IF rack. In the DCR window set up a couple of stripcharts to track the output.

4. Find a calibrator source. Use `GO` to track it, and then execute a **peak** procedure. Use `IARDS` to check the results.

5. Track your map center and set up the map (RALongMap or DecLatMap) in `GO`. Make your map!

Generally keep an eye on the stripchart and try to understand what's going on (is anything broken?). Make note of the IARDS calibration results, although they are repeatable later. *Make note of your project code and the full directory path to the data!* Noting the starting scan number of each map could also be useful.

After a few scans you should check the map. The "summary" command, when given the data path, will tell you how many scans you have and what they are: this is very useful. To check your map:

- Run `gbtmsfiller` from the command line to make your measurement set (input) for `AIPS++`. Designate minscan and maxscan using numbers from "summary"

- Fire up aips++ and make a map, following the directions from the web page ($gc.setdata()$, $gc.contcal()$, $gc.makeimage()$, $im := image, im.view()$...). In the x-band map, use `baseline=T`. In the "real" analysis you will also do a sky coverage correction.

Note that the map is in units of antenna temperature (*i.e.*, total power estimated to have been observed at each point in the sky). However *gc.contcal* must be given the correct T_{cal} value (see "Analysis Questions" handout)— for now you can assume unity, which is the default. This in turn can be scaled to Jy/beam via the GBT gain Γ (in K/Jy); with knowledge of the beam size, that in turn can be scaled to a physical intensity scale (Jy/Sr or Jy/arcmin^2).

Brian or Frank will be available for help if you need it.