UW-Madison SRT Manual

Snezana Stanimirovic

September 4, 2024

1 Introduction

The Astronomy Department at UW operates two Small Radio Telescopes (SRTs), one located on the roof of Sterling Hall and another one located at a UW facility in Pine Bluff. Both telescopes are operated remotely from the undergraduate classroom (3521 Sterling Hall). There are two designated computers (called Observer) to connect and operate each SRT.

The SRTs were developed by the Haystack Observatory (MIT) and have been assembled and calibrated for class use since 2007. A printed copy of the original Haystack SRT Manual is located in the classroom. This manual is very helpful for understanding how to work with the software and contains example command scripts.

2 Connecting to SRTs

Use the Microsoft Remote Desktop on two OBSERVER computers. Click on "dome" or "pbo92". Passwords: "srt", "srt". After finishing, **stow** the SRT and **Exit** from the interface. Then go back to the Microsoft Remote Desktop and **disconnect**.

3 Preparing observing files

When observing, it is common to write command scripts as this saves time and avoids typing mistakes. The observing file needs to be on the SRT computer (pbo92 or dome) in the directory called **SRTvsrt** (you will see other observing files there). You can do this in two ways: (a) you can write a script and send it to yourself as an e-mail, then on the observing computer open your e-mail and download the script; (b) there are many existing scripts in the observing directories, you can copy and edit one for your need. To open an existing file use the right-hand mouse button and select "Edit". This will open a Notepad window and allow you to edit the file. Please make sure the observing file has .cmd extension. Click on the file name and make sure it is listed as a Windows

Command Script[1].

To create .cmd file on one of iMacs in the lab, please follow these steps.

1. Launch “TextEdit” using the search bar in the top right corner of the home screen
2. Under TextEdit, select Preferences from the drop-down menu. Under Format select “Plain Text”.
3. After editing your command file, hit Command+S to “Save As”. Navigate to the uwastro directory under Macintosh HD/Users . If you are preparing a file to run on the Simulator, save it as “srt.cmd” as this is the only file name that works in the simulator.
4. Unselect the option entitled ”If no extension is provided, use ’.txt’ Select Save. This should save the text as exclusively plain text in a .cmd.

4 Simulator: testing your observing scripts

iMacs in the lab run the off-line version of the SRT software, or Simulator. This program is useful to familiarize yourself with the SRT graphical interface, and test your observing scripts.

To access: go to any iMac except OBSERVERS, click on the SRT icon (bottom of the screen), select simulate options and “Run Program”.

To run a test observing file in the Simulator, your file needs to be called “srt.cmd” and needs to be in the directory: /Users/uwastro . Click on “Rcmdfl” on the top console, then enter srt.cmd in the command window at the bottom of the panel.

5 Observing

After loging into one of the SRTs, your observing file needs to be on the SRT computer in the directory called **SRTvsrt** (you will see other observing files there). Start the SRT interfate software by clicking on the SRT icon.

To run your command file click on “Rcmdfl” at the top console and enter the file name (with .cmd extension) on the command line window (bottom of the screen). As lines are read from the file the text will be echoed on the message board in green. Once done, copy (e-mail) the data file to your computer[2].

[1]When using a Notepad you can Open a new file, and copy your commands from your e-mail for example. When you try to save the file you will as several options for Save as, one of them being Windows Command Script.

[2]The sync program is running between the SRT computer and the directory XXX

Troubleshooting: (a) In the case that the command file stops while observing and does not execute commands as expected, please stow the telescope, exit for the graphical interface and then login and try again. Re-starting the interface usually clears the problem. Sometimes the interface gets stuck on the previous observing script and does not want to accept a new one. The solution is the same, close down and re-start the control interface.

(b) During observations please pay attention to the power counts on the screen. Those should be relatively constant when tracking a single object and have smooth rising/falling while going over the source. If the power counts are all over the place please note that down in the logbook and try observing at a different time. The Sterling SRT is surrounded by tall buildings and have problems detecting radio waves if the source is at a low Elevation (where the waves bounce off the buildings a lot).

(c) When turning on the system, please pay special attention to any error messages that may appear within the Message Board below the Total Power Chart Recorder. They may indicate that the system has failed to communicate through the coaxial cable (i.e. gnd 0 radio 1). If movement of the dish stops prematurely, "antenna drive status.." or "lost count" may appear in the Motion Status Display Area. If this occurs please turn off the control system and ask for help. Be sure to return the dish to the stow position each time before powering down and quiting the java applet.

6 SRT Hardware

6.1 Antenna & Motors

The small radio telescope, SRT, is a 7.5 foot (2.3 m) diameter parabolic dish and receiver designed by the Haystack Observatory and sold by Cassi Corporation. Complete specifications for the antenna, mount and receiver are available at the SRT website[3]

The SRT has a focal length of 85.7 cm and the f/D = 0.375. It has a beam width of approximately 7.0 degrees[4]. The dish is mounted on a two-axis azimuth/elevation mount. It is supported by an aluminum frame constructed from C/Ku band mesh that reflects all incident microwave energy if the surface holes are less than 1/10th of the incident wavelength. The system is controlled using a computer running a java applet that communicates with the controller, which contains a Basic-Stamp microcontroller. The microcontroller in turn controls the motor functions. A very basic block diagram of the system is shown in Fig. 1.

[3] The SRT memo series is: https://www.haystack.mit.edu/haystack-memo-series/srt-memos/ . Documents up to 2012 are for the original SRT project where telescope parts were provided by Cassi. In 2012 the new SRT project was developed to build telescopes from commercial parts. The Sterling and PBO SRTs are from the original project (Cassi).

[4] The first minimum of the antenna diffraction pattern is given by $1.22\lambda/d = 7$ degrees

6.2 Receiver

The SRT uses a phasing-type single-sideband scanning receiver. Fig. 2 shows a block diagram of the radio receiver and subsequent signal processing stages, while Figure 3 shows a picture of the antenna feedhorn. Radio power arriving from the sky is focused by reflection to the antenna feedhorn. Signals then pass through a low pass filter, a low noise pre-amplifier and a mixer. The signal is digitized and sent back to the controlling computer over a serial RS-232 link. The central frequency of the synthesized local oscillator (LO) used in the mixer is user selectable from within the JAVA program. This permits the investigator to look at different spectral regions around 1420 MHz.
1 - L-Band probe
2 - Low Noise Amplifier
3 - Universal Male F to F Coupler
4 - Receiver mounting bracket
5 - Video Port - Analog signal
6 - Power and Communication - Digital Signal
7 - Band Pass Filter
8 - Feed Horn Extension

6.3 Electronic Noise Calibration

Electronic noise calibration is performed using a noise diode, whose intensity and spectral distribution is approximately equivalent to a 115 Kelvin blackbody[5]. Compare this to the known signal strengths of some radio sources in the sky, shown in Figure 4. The electronic noise calibrator is a diode (Noise/Com NC302L) connected to a small dipole antenna attached to the center of the SRT dish. A small (6mA) current is sent through a controlled bulk avalanche mechanism, resulting in a wide band of frequencies. The dipole emits the signal which is detected by the receiver. To generate the correct wavelength signal, the dipole was fabricated to be 1/2 wavelength, or approximately 10.5 cm. To minimize signal from behind the dipole, the poles are set 1/4 wavelength, or approximately 5.25 cm, above the circular plate at the center of the SRT dish.

In practice, to calibrate the telescope you just have to point the antenna at an empty part of the sky, change the frequency to the hydrogen frequency, and hit the 'Calib' button on the software interface. The software outputs a single number called T_{sys} which represents the background noise level in the device. The software measures the ratio of the received power when the noise diode is turned on and then off again. Complete details of the SRT calibration are available at http://www.haystack.edu/edu/undergrad/srt/receiver/SRT_calibration.html and http:// web.mit.edu/8.13/www/calibrator_report.pdf.

6.4 SRT Software

Please take some time to look at the panel on the right-hand side of the software window. Familiarize yourself with the information that is shown there.

[5]Noise diodes that our SRTs have are very old and have likely decreased in its intensity.

The SRT is continually acquiring and processing RF signals incident upon the dish. The upper right plot in the software window shows individual power spectra updated every few seconds. The plot to the left in RED, shows an integrated power spectrum which may be cleared by pressing the "CLEAR" button.

During the calibration, you can point the telescope at any clear part of the sky (avoid buildings).

You can execute commands via the toolbar located at the top of the console. Clicking on any of these buttons either initiates an automatic response or the software waits for text input by the user.

Azimuth and elevation locations are set using the Horizon Coordinate system. Azimuth and elevation are entered in degrees. The controller moves the telescope to the required position. If the tracking modeis on (button text is YELLOW), when the telescope reaches the position, the controller continues to move the telescope to compensate for the Earth's rotation so as to keep the telescope pointing to the specified position in the sky. To turn tracking off, simply click on the button and it should turn RED.

Try the following to get started: Enter "180 40" in the text window and then press the Azel button. The dish should slowly move to point due south. Press STOW to return the dish to it's "parked" position at Azimuth=0, Elevation=2. Press the "RECORD" button to stop writing to the log file and open it up using a text editor.

It is useful to begin a data file ('.rad' file) which provides you with a continual record of your experiment. You can do this using the RECORD button. Hitting the record button again will cause the recording to stop.

Instead of using individual buttons from the toolbar, in some parts of the experiment it is useful to write a script (a '.cmd' file) and load the script to be run. Please see the Haystack SRT manual for sample scripts.

The program calculates the Local Sidereal and Universal Time based on the computer clock settings within the information sidebar on the right-hand side. Since Universal Time = Eastern Time + 5 hours (under Standard Time) and = Eastern Time + 4 hours (under Eastern Daylight Savings Time), you must make sure the box "Automatically adjust clock for daylight savings" is checked when setting the computer's clock. It also indicates the antenna direction in both Equatorial (RA and DEC) and Galactic ('l' and 'b').

7 Observing file syntax (MORE TO BE ADDED)

For examples of how observing scripts look like please check the Haystack Observatory SRT Manual (available on our Canvas course page). Note: all commands start with a space after : , except the integration time command, e.g.

```
: record test.rad
: freq 1420.4 4
: noisecal
: galactic 130. 0.
:300
: roff
```

For running the calibration diode, function “calibrate” is obsolite. We will use “noise-cal” instead.

8 The srt.cat (system) file:

The text file srt.cat, located in SRTCassi, is the primary configuration file for the SRT java interface program that is used to control the telescope. The srt.cat file may be updated, but upon doing so, please stow the telescope, shut down the control program, and restart it so that it recognizes any updates made to the srt.cat control file. The srt.cat file contains keywords which are case sensitive. Lines that are preceded by a ” are ignored. Blank lines are ignored.

Keywords:

AXISTILT* - Use to set the azimuth and elevation axis tilt. Default is 0 0

AZEL* - Allows user to catalog a fixed catalog location by name.

AZLIMITS - This may be set to allow movement in a clockwise position from stow from 90 to 270. At the midpoint the telescope will face due south. Degree limits should be set slightly below the physical limitations of the telescope. It is recommended to use 95 and 265. The telescope movement is clockwise as seen from above through increasing degree.
BEAMWIDTH* - Antenna beamwidth in degrees, default is 7.0.

CALCONS* - Gain correction constant. This is the ratio of temperature (K) per count. Change this during calibration (should be 1 to 1 ratio). Default is 1.0. This will be changed by software after calibration occurs and value will be available in the status bar.

COMM* - Communication port, default 1, use 0 for linux.

COUNTPERSTEP* - Counts per step for stepped antenna motion. The default is no stepped motion.

CURVATURE* - Optional correction for curvature in spectrum. The default is 0.

DIGITAL - Indicates that a digital receiver is being used. Comment out if using the analog receiver (set via jumper removal).

ELBACKLASH* - Optional correction for elevation backlash to improve pointing in flipped mode. The default is 0.

ELLIMITS - This may be set to allow movement in a clockwise position from stow from 0 to 180. At the midpoint the telescope will face up. Degree limits should be set slightly below the physical limitations of the telescope. It is recommended to use 10 and 175. The telescope movement is clockwise as seen from above through increasing degree.

GALACTIC - Set a galactic coordinate by location and name.

MANCAL* - Calibrates vane. 0 indicates auto-calibration, 1 indicates manual calibration. Is not a requirement of the digital setup. Moon - Add the Moon to the catalog.

NOISECAL - Calibration temperature of the noise diode located at center of dish. Temperature is in K.

RECORDFORM* - Adds tabs to separate columns in output file. The default is space delimited. "VLSR" adds vlsr and "DAY" forces a file change at each new day.

SOU - Set a source in the catalog by location and name. For negative declination us "-" in from of dd.

SSAT* - Satellite ID and location (satellite name then longitude west).

STATION - Used to set latitude (in degrees), longitude (in degrees), and station name.

Sun - Add the Sun to the catalog.

TLOAD* - Load temperature. Default is 300 K.

TOLERANCE* - Counts of error which can accumulate before command to drive stow occurs. Default value is 1.

TSPILL* - Antenna spill over temperature in K. Default is 20 K.

9 Data file format

The extension is .rad for an output data file. The data file can be opened in Python. The data values are separated by spaces. The file has the following formatting structure.

Any comment line starts with an asterisk. The first line of an output data file

```
* STATION LAT= 37.2 DEG LONGW= 80.4
```

gives the observing site location. Calibration results produced while data are being recorded are also put on a comment line, such as:

```
* tsys 215 calcons 0.98 trecvr 195 tload 300 tspill 20
```

Data lines start with a UT date and time for the line (yyyy:ddd:hh:mm:ss), followed by the the pointing information: azimuth, elevation, azimuth offset, elevation offset, first frequency of the spectrum being recorded (in MHz), spacing between frequencies (in MHz), observing frequency mode number, number of frequency bins, and then the calibrated temperature value for each frequency bin (units of K).

Thus a typical line might look like:

```
2005:148:10:55:41 92.0 5.0 0.0 0.0 1419.75 0.00781250 1 64 4.7 5.9 10.5 20.1 40.1 70.
```

Field 0 - time (yyyy:ddd:hh:mm:ss)
Field 1 - azimuth(deg)
Field 2 - elevation(deg)
Field 3 - azimuth offset(deg)
Field 4 - elevation offset(deg)
Field 5 - first frequency bin (in MHz)
Field 6 - digital frequency separation (in MHz) = freq. channel width
Field 7 - digital spectrometer mode
Field 8 - number of frequency channels
Field 9 – data value (in calibrated temperature units) at first frequency channel
field 10 – data value at second frequency channel

10 Reading the data file in Python

You can read .rad file in Python in many different ways, I like using functions "open" and "readline" . Here is a simple example of what can be used to read spectra from a .rad file. As the .rad file contains lines that start with *, using a text editor you need first to remove those lines. Then read a line from the file, and split into individual data points. Take the last 64 data points from each line as your spectrum, and save all spectra into a 2D array.

```
spectra = []
frequency = []
counter = 0
file="test41.rad"
for line in infile:
    print(infile.readline())
    line_list = line.strip().split()
    str_spec = line_list[-64:]
    specific_spec = []
    for item in str_spec:
            specific_spec.append(float(item))
    plt.plot(specific_spec)
    plt.ylabel('Brightness Temperature(K)')
    plt.xlabel('Bin number')
    plt.show()
    spectra.append(specific_spec)
infile.close()
```

Note: the number of channels (64 in this case) will depend on the exact spectrometer mode used. Only mode 4 has 156 frequency channels, while all other modes have 64.

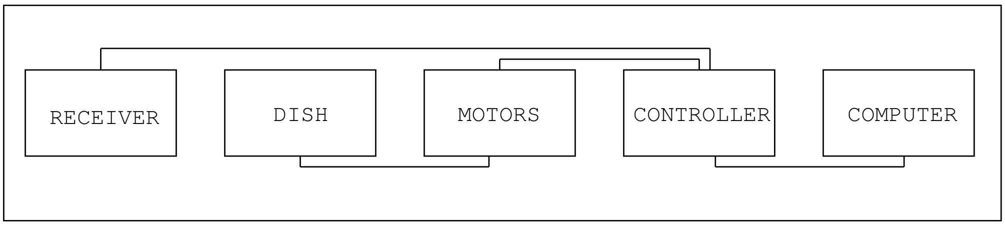

Figure 1: A basic block diagram of the SRT.

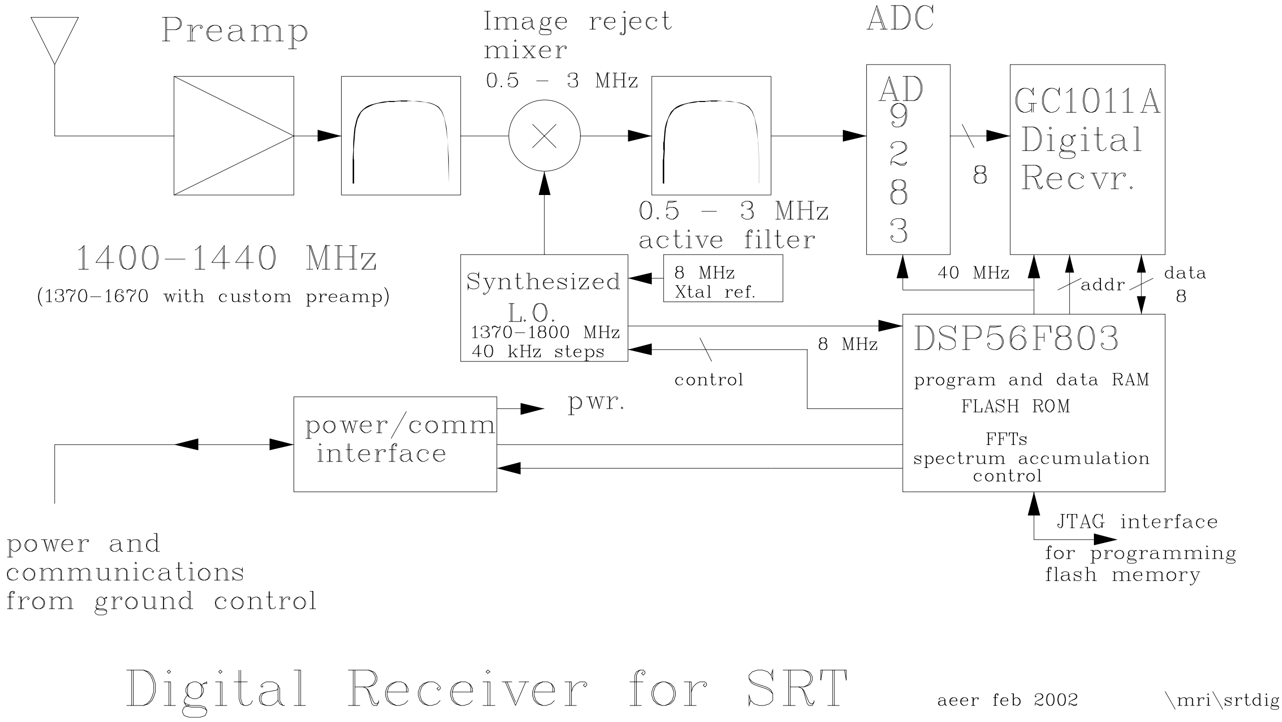

Figure 2: A basic block diagram of the SRT receiver.

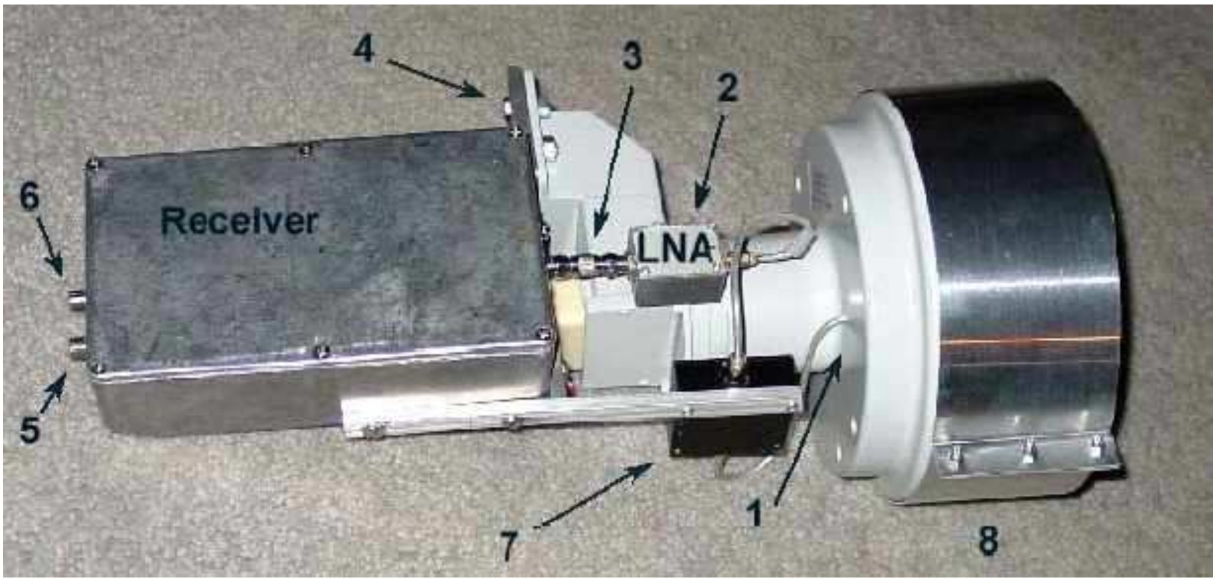

Figure 3: A basic block diagram of the SRT receiver.

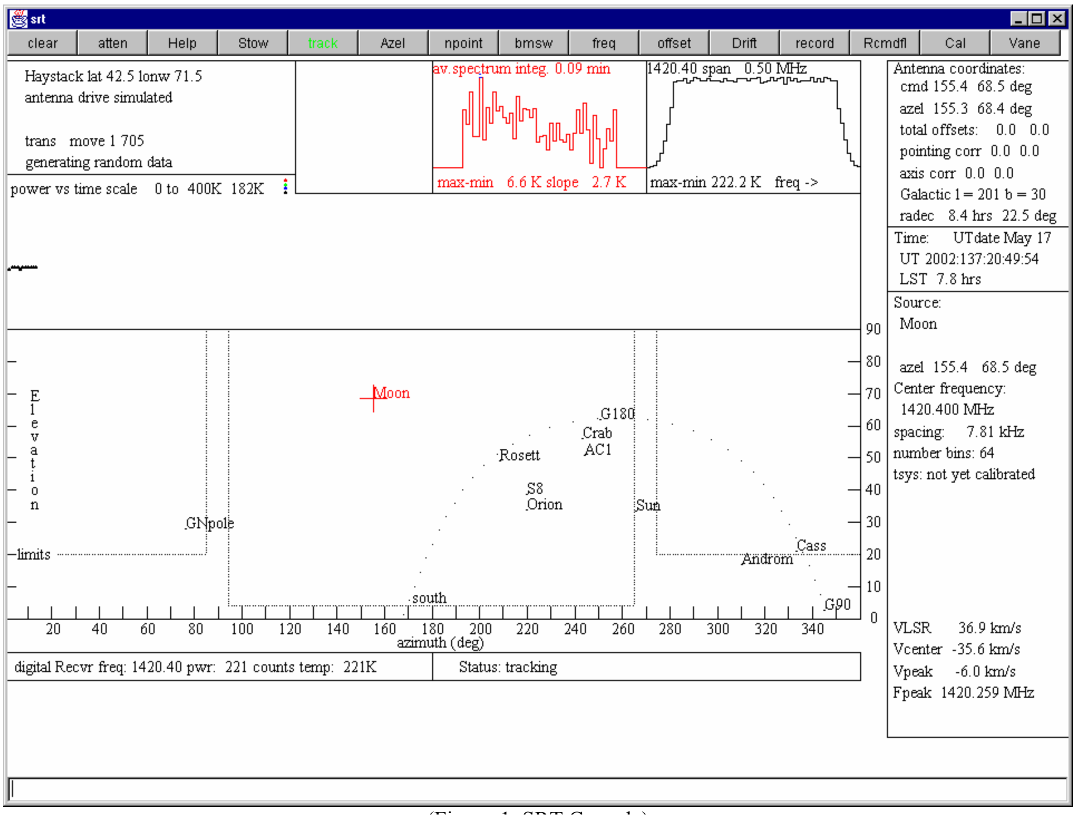

Figure 4: SRT console.