HI experiment with horn antenna

April 4, 2023

1. Introduction

This document presents the procedures involved in the 21 cm hydrogen line experiment with horn antenna. Procedures for horn antenna assembly, electronics assembly, software setup, data analysis and calibration are outlined. The codes for the entire calibration and data analysis process are given in the Appendices.

2. Horn antenna

A horn antenna consists of a waveguide and a flare. We will look at both these components in the following sections.

2.1. Waveguides

2.2. Horn design

3. Horn assembly

The antenna has following items.

1. Four panels, two narrow and two wide.
2. Mount base.
3. Mount U section.
4. Elevation rods and couplings.
5. Nuts and bolts.
6. Panel mount N-type connector with solder cup.
7. Aluminium tape

The procedure for antenna assembly briefly enumerated below.

1. Keep all the necessary equipment eg., screwdrivers, spanners, aluminium tape, with you.
2. Cut a 2mm thick copper wire with length 4.9 cm. The copper wire should be un-coated.
3. Solder the probe to the N-type connector.
4. Fix the connector on the waveguide with nuts and bolts.
5. Start with the narrow panels. The elevation rods need to be fixed to these panels.
6. Once the rods are fixed, put all four panels in an inverted pyramid and fix them together with nuts and bolts.
7. Put aluminium tape on the inside four corners of panel joints to cover the entire joint and cover the openings. This is to avoid external radiation from entering the horn through these openings.
8. Bolt the waveguide to the pyramid.
9. Put aluminium tape to cover waveguide joint.
10. Put The U section on the base and fix the screws at the coupling pipes.
11. Mount the pyramid on the U section and fix the elevation couplings.
12. Check the joints and azimuth elevation motion of the antenna.

4. Electronics

The front end electronics consists of two low noise amplifiers [1] (LNA) and a band pass filter [2]. The back end consists of a software defined radio receiver (SDR) [3].

Each LNA has a gain of 20.5 dB which makes the total gain for two LNAs

41 dB. This translates to a linear gain of

$$10^{41/10} = 12589 \tag{1}$$

The noise figure for each LNA is around 0.55 dB. Noise figure (NF) is defined as

$$NF = 10log(F) \tag{2}$$

where F is noise factor. Noise factor is measure of how much a signal degrades as it passes through a device. It is defined as the ratio of signal to noise ratio (SNR) at the input to the signal to noise ratio at the output.

$$F = \frac{S_{in}/N_{in}}{S_{out}/N_{out}} = \frac{SNR_{in}}{SNR_{out}} \tag{3}$$

Here, S is the signal power and N is the noise power. For a multi stage system the noise factor is given by the Friis formula.

$$F = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + \frac{F_4 - 1}{G_1 G_2 G_3} ... \tag{4}$$

$F_1, F_2 ...$ and $G_1, G_2 ...$ are the noise factors and gains for stage 1, stage 2 and so on. Using equations (2) and (4) the net noise figure for two amplifiers is then about 1.3 dB. The LNA is the first stage of the front end. This is because the signal is very weak and need to be amplified without adding too much noise to the signal during amplification. Once the electromagnetic waves falling on the antenna are converted to electrical signals by the antenna probe, every component in its path will add its own noise to the signal. This is apparent from equation (4). The low noise amplifiers are specially designed to add very little noise to the input signal.

The band pass filter operates from 1350 MHz to 1450 MHz. It attenuates signals that don't fall between this band. The purpose of the filter is t reduce aliasing noise and suppress any strong out-of-band signals. This improves the SNR of the signal.

The front end electronics connects directly to the N-type connector on the waveguide.

4.1. Assembly

The assembly process is as follows.

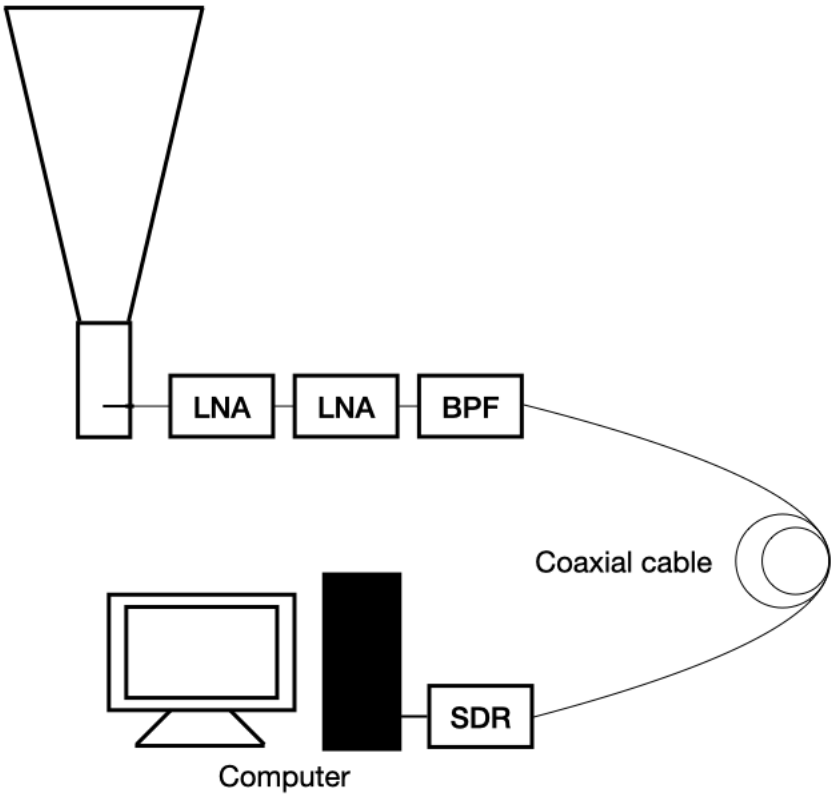

Figure 1: Experiment setup

1. Solder the power supply wires to the low noise amplifiers. Care must be taken while handling the LNAs as they are prone to damage by

electrostatic discharge. It is recommended to have an electrostatic protection by grounding yourself while touching LNAs.

2. Connect the two LNAs followed by the filter with appropriate connectors.
3. Connect the assembly to the antenna.
4. Connect a long (10 m) RG58 coaxial cable to the filter.
5. Connect the other end of the cable to the receiver.
6. Plug the receiver in the computer or laptop.
7. Connect the LNA power cables to the battery and turn it on.

4.2. Software Defined Radio (SDR)

Traditional radio communication systems had hardware components which performed specific tasks. The entire systems were in form of hardware. Improvements in modern computing and digital signal processing has enabled deployment of several tasks in software. For example, analog filters are replaced with digital filters. The hardware filter need not be modified. Instead, the filter response can be changed by changing the filter code. If a few or most of the components of the communication system are designed in software then the system is called a software defined radio or SDR for short.

The rtl-sdr dongle used in the current experiment has capabilities to turn it into a SDR. The dongle gives raw digital samples to the computer which can interpret and process them in any way required. For example, FM demodulation can be done entirely in software, which is how some of the software packages can give FM reception in software even though the dongle itself does not do any FM demodulation. For digital signal processing and SDR enthusiasts, a good hands on tutorial can be found here [4].

The rtl-sdr dongle has 8 bit analog to digital converter with IQ sampling.

Some of its hardware parameters that can be set with software commands are as follows.

1. Receiver can be tuned to frequencies ranging between 24 MHz to 1766 MHz.
2. Maximum sampling rate can be set to 3.2 Ms/s.
3. Ther internal LNA gain can be set upto 50 dB.

5. Antenna pointing

5.1. Source location

The source location can be determined using stellarium [5]. Clicking on any star in the app will show the star's coordinates. A basic process of getting the source coordinates is given below for getting pointing information for sources in the galactic plane.

1. Set the location in the application for correct sky map. This can be done by dragging the mouse pointer to the bottom left corner of the application screen and bringing up the options. click on the *Location* option to set the location.

2. Bring up the options again by dragging the mouse pointer to the bottom left corner of the application screen. Click on *Sky and viewing options* and choose *Markings* on the top of the window. This will bring up the markings window. Click on the three boxes in front of *Galactic equator*. The galactic equator is now visible in the map.

3. Check the time of display at the bottom right corner in the application. It has to be current date and time for correct sky coordinates to be displayed. Current time can be choosing the downwards pointing arrow below the date time. If the arrow is white then the application is following the current date time from computer.

4. Zoom in on the location you want to point to in the sky and click on a star closest to the location. This will show all the coordinates of the star. You can see the *Az./Alt.* coordinates which will be changing with time as the source moves through the sky. These are the coordinates that are used for pointing the antenna to the source.

Once the source azimuth and elevation are noted the antenna needs to be turned to those azimuth and elevation.

5.2. Azimuth and Elevation pointing

Antenna pointing can be done by putting elevation (El) and azimuth (Az) scales on the antenna mount and calibrating the scales with the north direction (for Az) and the horizon (for El). A drawback that some setups might face is that the mount has to be fixed to the ground after the scales are calibrated. If the mount moves during observation, offsets are introduced in the antenna pointing. Therefore, re-calibration is required.

One solution to the problem is to use mobile phone for pointing the horn. Mobile phones have several sensors, some of which can give absolute Az and El pointing. A magnetometer can sense earth's magnetic field and give a pointing to the north and an accelerometer can measure the local acceleration due to gravity (g) and give the vertical angular measurement w.r.t. the g vector. These measurements can be used to determine the Az and El angles.

There a re several mobile applications which can do this, eg. SatFinder [6] which gives the Az and El angles for the direction in which the camera is pointing. We can get the pointing by placing the mobile phone at the back of the horn and turn the antenna to the desired location. The pointing procedure is as follows.

1. If using SatFinder application:

   1.1 Start the application. Give app the necessary permissions (app needs GPS access to get your location for correct pointing).

   1.2 Go to AR mode in options.

   1.3 The app should now show camera output with azimuth and elevation pointings at the bottom left corner.

2. Put the mobile phone at the back plate of the horn waveguide. Note the Az and El angle in the mobile app.

3. Turn the antenna to the azimuth pointing first by keeping the elevation angle zero. This is done to reduce the azimuth pointing error in some mobile phones.

4. Fix the antenna azimuth angle by tightening the azimuth coupling screws.

5. Turn the antenna vertically while holding the mobile on the back plate.

6. Fix the elevation coupling screw once the desired elevation is obtained.

One important point to note is that the magnetic sensor the mobile phone reacts to feromagnetic metals. Since aluminium is paramagnetic the sensor will work. But in case it is used with metals like iron then the sensor will give wrong values if the mobile is too close to such metals.

6. Data acquisition

There are several software packages and tools available for RTL SDR. A list containing large number of such software can be found here [7]. The device driver needs to be installed to use most of the software. For the current experiment we will use a command line tool *rtl power* which comes with the rtl-sdr driver. The installation process if as follows.

6.1. Linux

Detailed instructions for installation on linux systems csn be found here [8]. Installation on linux systems has following requirements.

- libusb (*sudo apt install libusb*)
- cmake (*sudo apt install cmake*)

Installation from the source can be done as follows.

1. Download source code from `https://osmocom.org/projects/rtl-sdr/wiki/Rtl-sdr`.
2. Run following commands in terminal to complete installation.
   2.1 cd rtl-sdr/
   2.2 mkdir build
   2.3 cd build
   2.4 cmake ../
   2.5 make
   2.6 sudo make install
   2.7 sudo ldconfig

You may need to blacklist the rtl sdr kernel driver to use the sdr. To do this, create a file at */etc/modprobe.d/blacklist-rtl2832.conf*. Add the following lines to the file.
blacklist rtl2832
blacklist dvb_usb_rtl28xxu
blacklist rtl2832_sdr
blacklist rtl8xxxu

This needs to be followed by the following commands or restarting the com-

puter. *sudo rmmod rtl2832_sdr*
sudo rmmod dvb_usb_rtl28xxu
sudo rmmod rtl2832
sudo rmmod rtl8xxxu
Details of blacklisting can be found here [9].

6.2. Mac OS

1. install MacPorts from here [10].
2. run the following command in terminal to install rtl sdr.
   sudo port install rtl-sdr

6.3. Data command

After installation the installation can be checked by running the following command.
rtl_test -t
If the command runs without error then the installation is successful. The following command can be used in the terminal to record and save the data to file.
rtl_power -f 1419.5:1421.5:4k -g 50 -i 60s -1 test.csv
The command give a power spectrum with parameter given in the command.
rtl_power -h command gives the usage of the command.

- **-f** defines the *start frequency:stop frequency:bin size* of the spectrum.
- **-g** sets the gain of the receiver.
- **-i** sets the integration time.
- **-1** means we capture only one spectrum.
- Finally the spectrum is stored in a file named *test.csv*. The files are stored in the directory from which the command is run

The files are stored as .csv files. The file name should be changed for every pointing. It is recommended to give file names as the source names like ground.csv (when antenna points to ground), zenith.csv (when antenna points to zenith), 90.csv (for galactic longitude 90 in galactic plane), etc.

7. Calibration

Calibration involves converting the power and frequency axes of the data to temperature and velocity axes respectively. These quantities relate to physical properties of the source. We will see both these calibrations separately in the following sections.

7.1. Temperature calibration

Temperature calibration converts the power units to temperature (K). Consider an antenna pointing to source with temperature T. If the antenna is terminated with a matching load. The the power P absorbed by the load through the antenna is proportional to the source temperature. This relation is given by $P = kT$ where k is the Boltzmann's constant.

Let us select two sources two sources with known temperature, i.e., ground or a nearby wall with temperature T_g and sky with temperature T_{sky}. We can write the following equation

$$\frac{P_g}{P_{sky}} = \frac{T_g + T_r}{T_{sky} + T_r} \tag{5}$$

In the above equation P_g is the power measured by the receiver when the antenna is pointing to ground and P_{sky} is the power measured by the receiver when the antenna is pointing to sky. When pointing to ground the measured power has contributions from the ground temperature T_g and the receiver or radiometer temperature T_r. Similarly, when pointing to sky the contributions are from the sky temperature T_{sky} and the receiver or radiometer temperature T_r. We can consider the ground temperature to be 300 K and the sky temperature to be 5 K. The sky temperature has contributions from the cosmic microwave background and the earth's atmosphere.

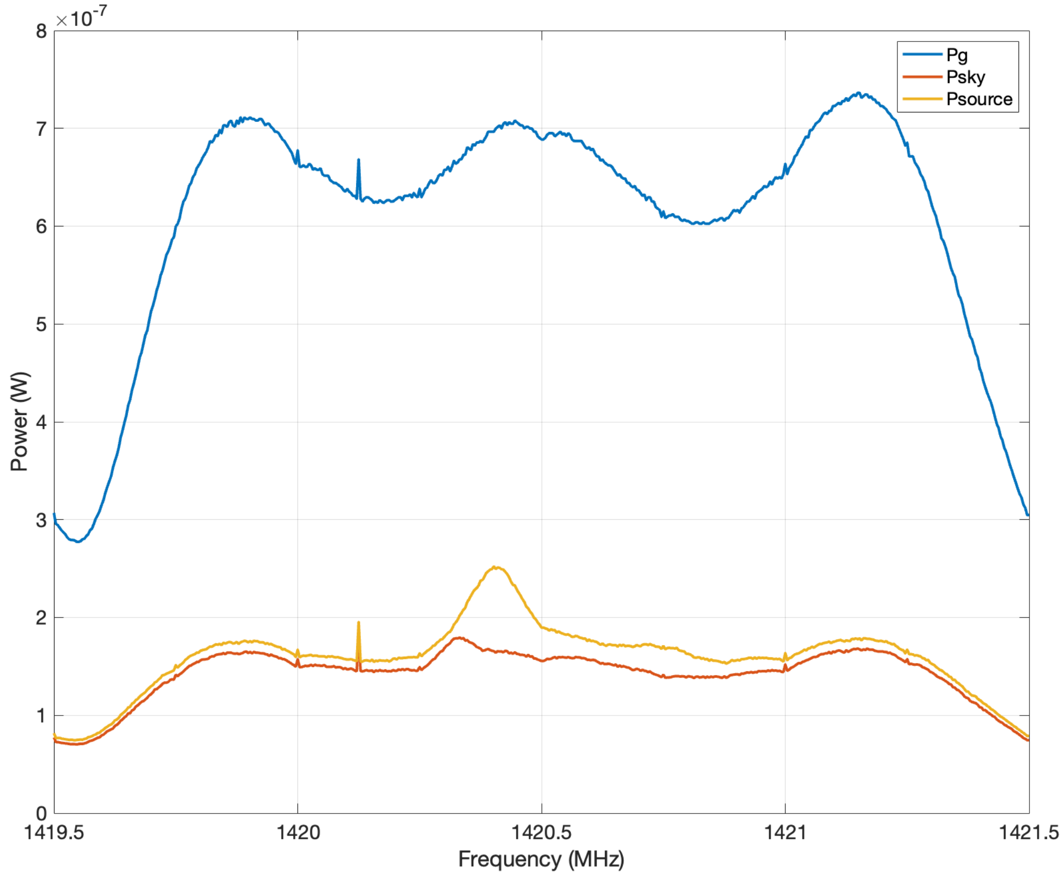

Figure 2: The measured powers P_g,P_{sky} and P_{source}.

We can simplify equation (5) to get the unknown receiver temperature T_r.

$$T_r = \frac{T_{sky}\frac{P_g}{P_{sky}} - T_g}{1 - \frac{P_g}{P_{sky}}} \tag{6}$$

Once we know T_r we can find the temperature of an unknown source T_{source} with measured power P_{source}. We put T_r back in equation (5) replace one of the powers in equation (5) with the measured source power and replace corresponding temperature with T_{source}. This gives

$$T_{source} = \frac{T_g + T_r}{\frac{P_g}{P_{source}}} - T_r \tag{7}$$

Equation (7) gives calibrated brightness temperature of the source.

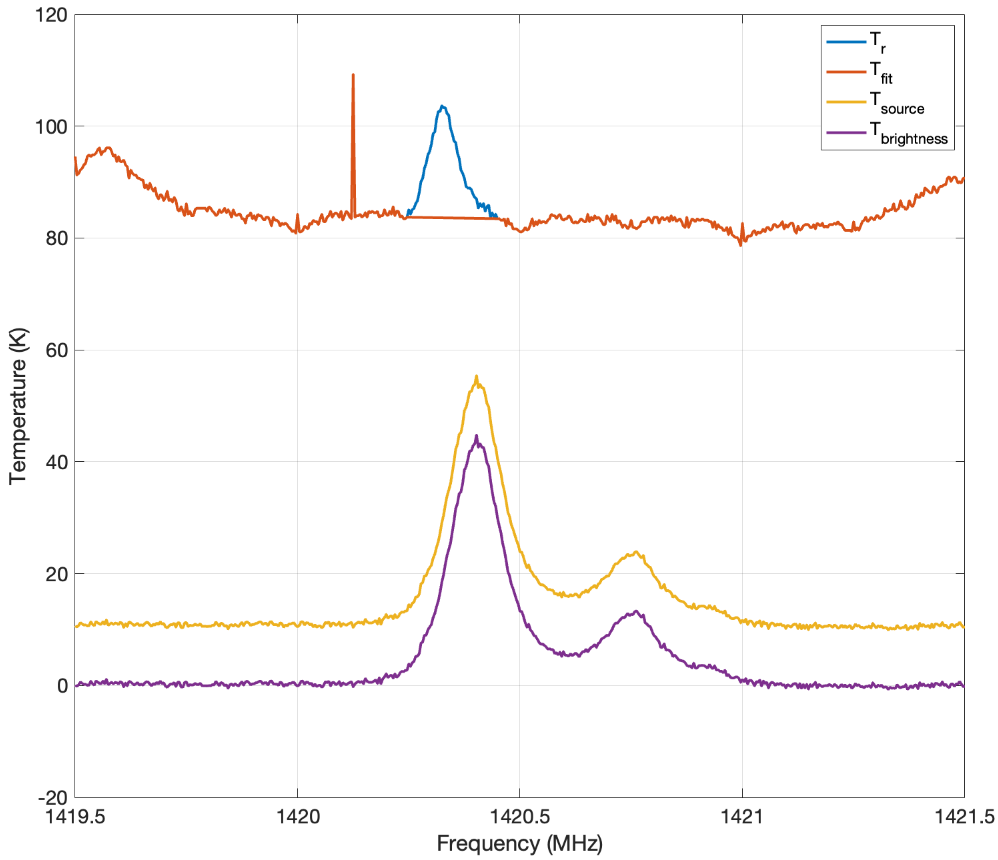

Figure 3: The temperatures T_r,T_{fit},T_{source} and $T_{brightness}$.

The sky temperature measurement is taken when the antenna is pointing to sky away from the galactic plane but near zenith. Since HI line is present in all directions it is difficult to measure sky temperature without the line feature appearing in the data for sky temperature. The same feature is repeated in T_r. We need to get rid of the feature from T_r as this will excess temperature from HI. We can do this by fitting a line to HI region of T_r baseline as seen in figure 3. The curve T_{fit} is a line fit to T_r where the HI line is present. We can now use T_{fit} in source calibration equation (7) instead of T_r.

7.2. Velocity calibration

To convert the plot to velocity we use the Doppler equation (8)

$$v = c(1 - \frac{f}{f_0}) \tag{8}$$

where c is the speed of light, f is the frequency axis of the plot and f_0 is the rest frequency of the hydrogen line. A correction factor needs to be added to the velocity obtained in equation (8). This correction factor takes into account the rotation of earth, the revolution of earth around the sun and the peculiar motion of the sun w.r.t. the local standard of rest. See Appendix A for the python code to obtain this correction velocity. Once the correction is applied, the velocities are now w.r.t. the local standard of rest (LSR).

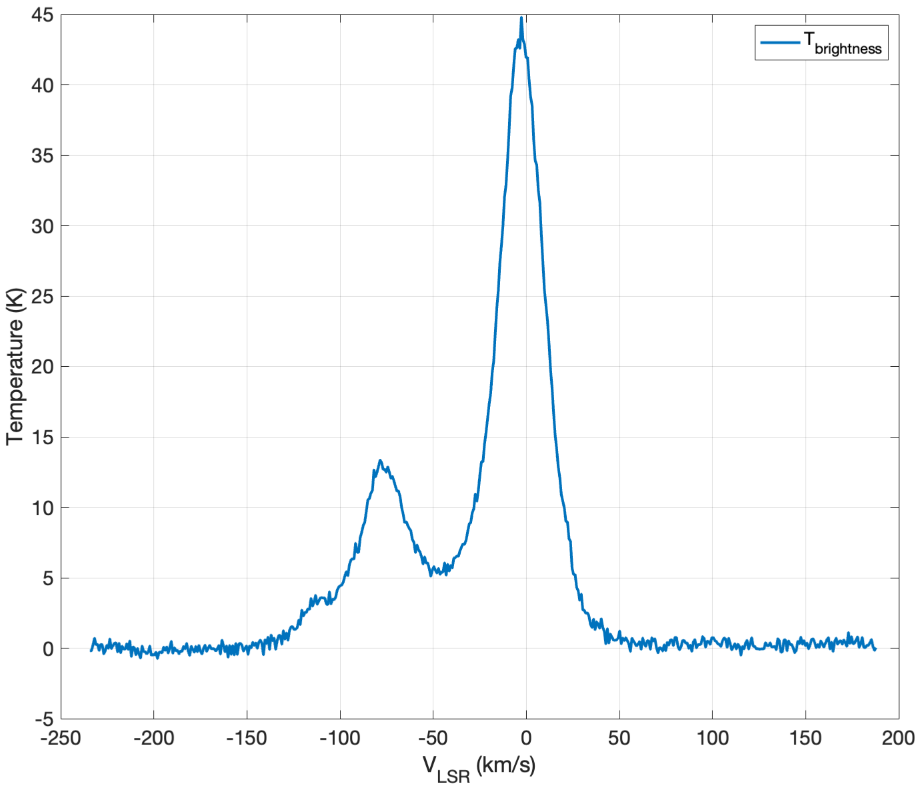

Figure 4: The frequency converted to velocity.

Velocity calibration is equivalent to frequency calibration (8). If a calibrated single tone transmitter with transmission frequency equal to or close to the HI line rest frequency then that signal can be fed to the receiver and frequency offset and stability of the receiver can be tested. This is would be the laboratory method of frequency calibration. For the purpose of this experiment, we can compare our observations to standard surveys to check for velocity as well as temperature calibration. Eg., the Lieden-Argentime-Bonn (LAB) [11] survey can be used for this purpose. We convolve the antenna beam with the survey data to get comparable results. Figure 5 shows one such comparison. We can see a good agreement between our observations and the survey data. This is an independent way to calibrate the velocity and temperature.

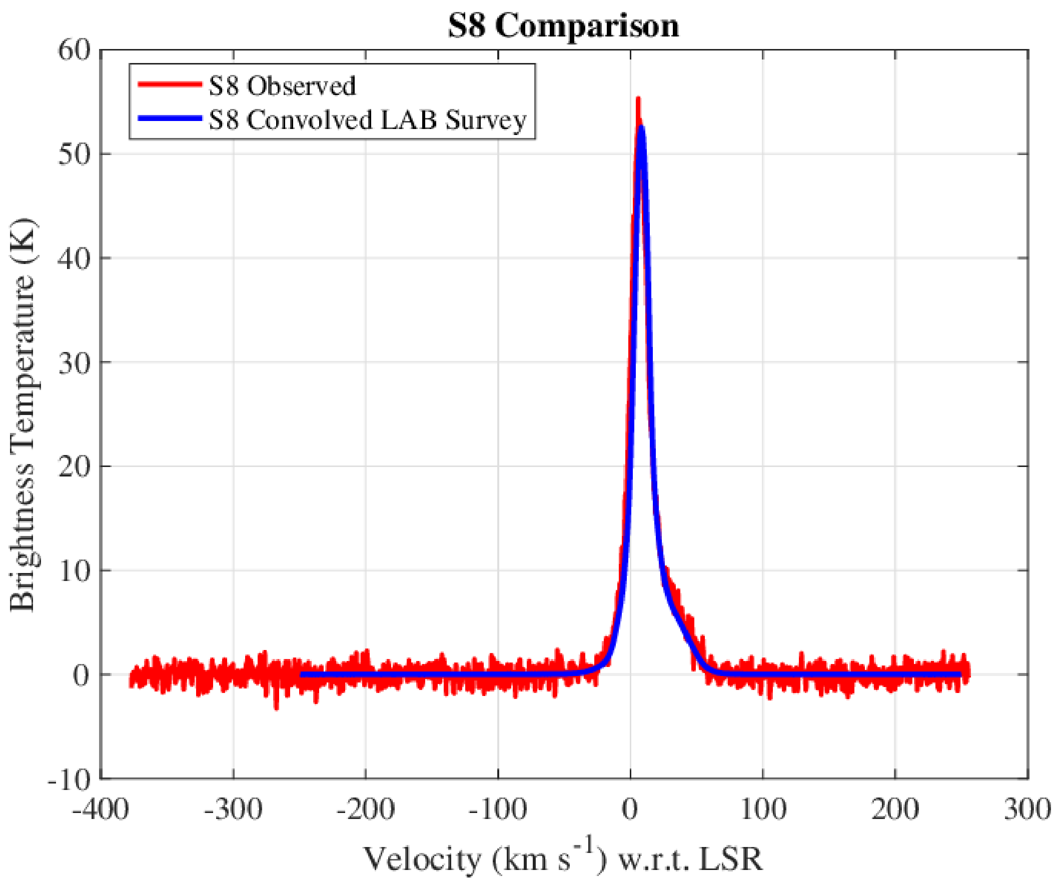

Figure 5: Observation of HI in the S8 region in the Orion constellation compared with the LAB survey. (Figure reproduced from Mhaske et.al, [12])

7.3. Notes on calibration

- Calibration requires ground and sky measurements. These measurements can be ideally repeated for every observation or once every hour or so.
- LNA as well as system noise contribution is sensitive to temperature variations. Hence, calibration must be done for every set of observations.
- While pointing to the wall or ground for ground measurement, the wall should not be too close and also should not be perpendicular to the antenna. This should be followed to avoid formation of standing waves in between the wall and the antenna which can introduce errors in calibration. Ideally a microwave absorber should be used instead of ground reference. But if the wall is in far field and the antenna is not directly pointing the wall, the procedure works for the purpose of the current experiment.

8. Data analysis

8.1. Fitting

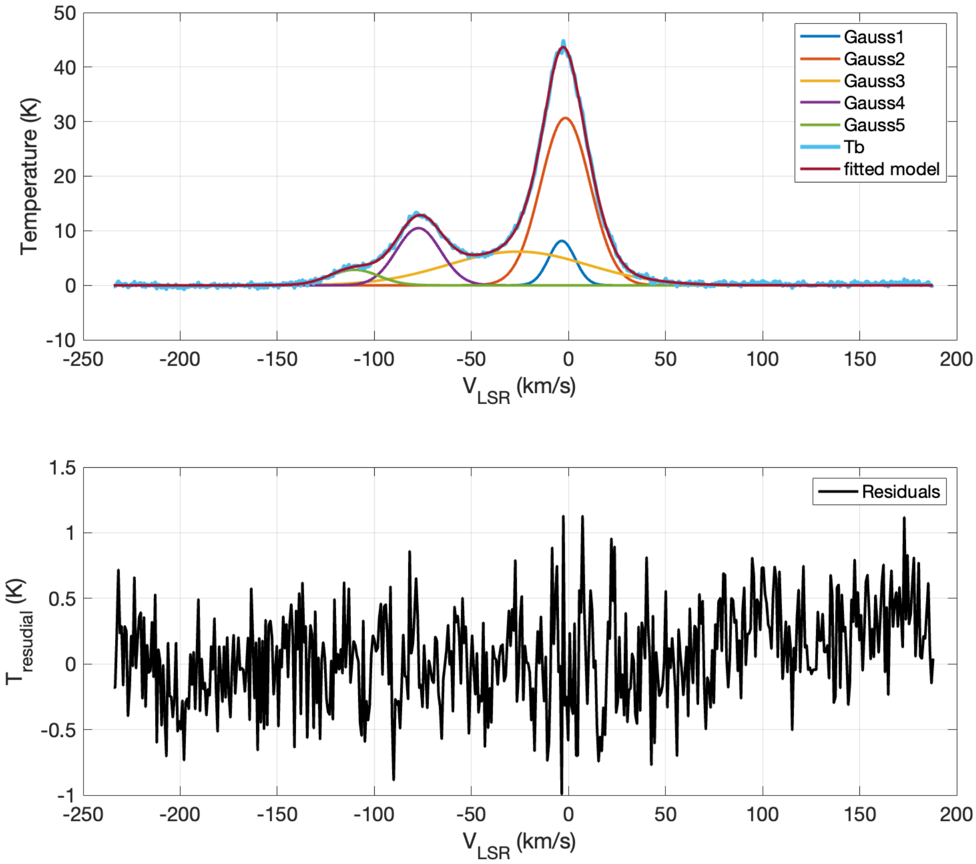

Figure 6: HI line fitting with gaussians and the residuals.

The calibrated plots can now be fitted with gaussians. There are multiple peaks in the plots. Hence, multiple gaussians are fit to the data. As seen in figure 6, we create a model of the line profile by fitting multiple gaussians. Subtracting the model from the data gives us the residuals. The software for fitting is left to the user. Here, the MATLAB fitting tool was used to generate the model. We use the model in the following sections to determine the rotation curve and the Galactic structure for the Milky Way Galaxy.

9. Results

The Galactic HI line profile has features like line width, Dopplar shift and multiple peaks. These are indicators of the dynamics of the sources in the line of sight. We can use these parameters to estimate the rotation curve and the structure of the Milky Way Galaxy. The first estimates of the rotation curve and the structure of the Milky Way Galaxy using HI line were done by Hulst et al. [13].

9.1. Rotation curve

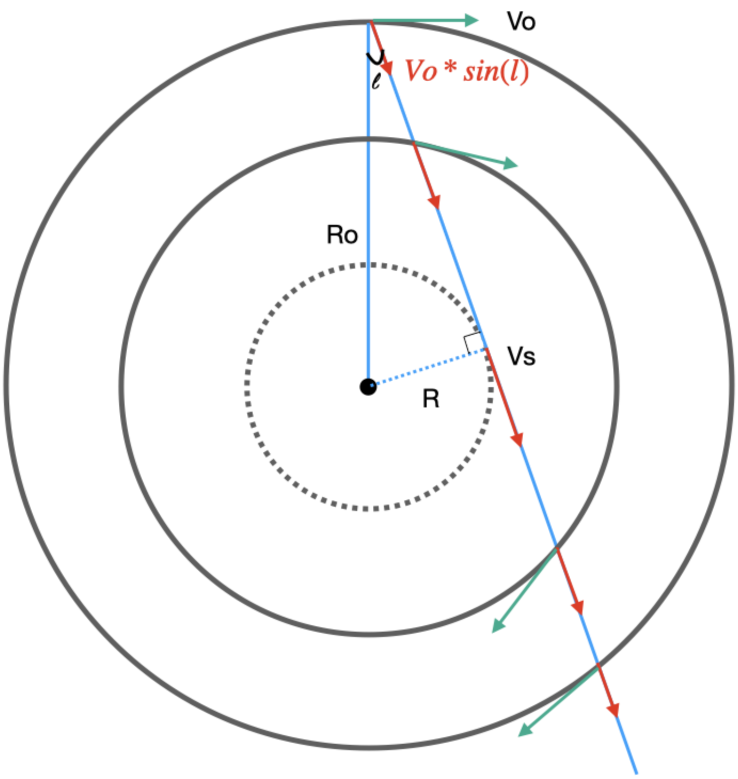

Figure 7: A model for getting the rotation curve.

The rotation curve of the Milky Way Galaxy can be determined using the model created by fitting gaussians. Consider figure 7 where V_o is the velocity of the Sun w.r.t. the Galactic center, R_o is the distance of the Sun from

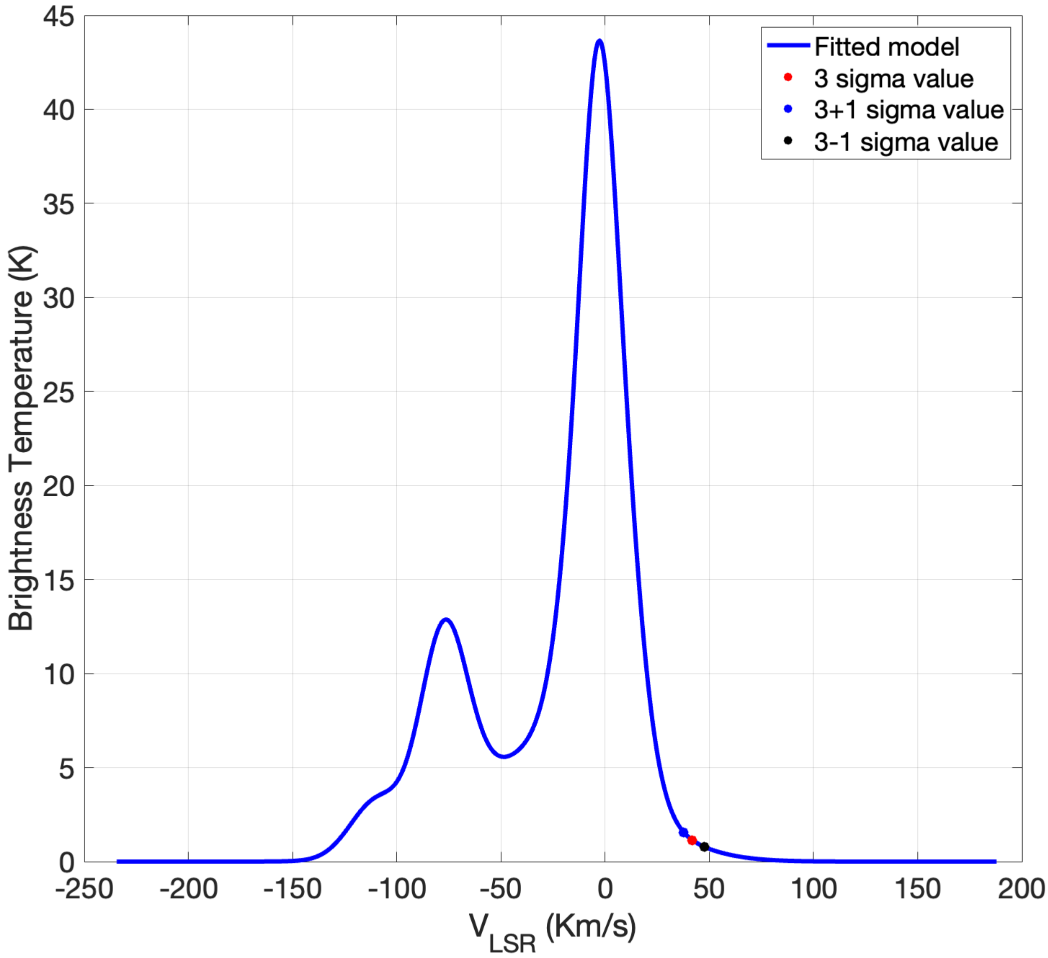

Figure 8: Selecting the most redshifted point from model with 3 σ point and ± 1 σ error bars.

9.2. Galactic structure

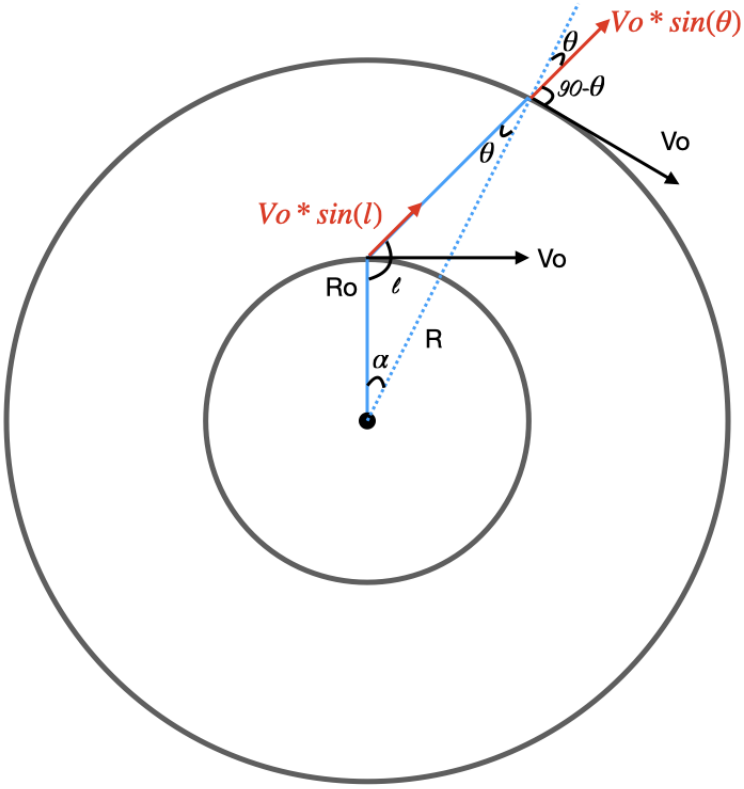

Figure 9: Figure showing the method of plotting galactic structure.

A. AppendixA: Python code for getting correction velocity

The code needs RA and DEC (J2000) (HH MM SS.ss Deg Min Sec.ss) of the source and the Julian date (YYYY MM DD HH MM SS.ss). This information is available in Stellarium. The code also needs coordinated (latitude, longitude and altitude above sea level) for the observatory. These details for IUCAA are entered.

```
#!/usr/bin/env python
# coding: utf-8

from __future__ import print_function, division
from PyAstronomy import pyasl
import math
```

```
import datetime

# Coordinates of telescope
longitude = 73.8253
latitude = 18.5593
altitude = 554

# Coordinates of source (RA2000, DEC2000) RA_hr RA_min RA_sec
   DEC_deg DEC_min DEC_sec. Note DEC must be signed + or -.
hd1 = "03 07 07.85 +58 19 30.20"
obs_ra_2000, obs_dec_2000 = pyasl.coordsSexaToDeg(hd1)

# Time of observation converted to Julian Date
dt = datetime.datetime(2019, 2, 12, 15, 49, 43)
jd = pyasl.jdcnv(dt)

# Calculate barycentric correction (debug=True show
# various intermediate results)
corr, hjd = pyasl.helcorr(longitude, latitude, altitude,
   obs_ra_2000, obs_dec_2000, jd, debug=True)

#print("Barycentric correction [km/s]: ", corr)
#print("Heliocentric Julian day: ", hjd)

# Calculate LSR correction
v_sun = 20.5 # peculiar velocity (km/s) of sun w.r.t. LSR
   (The Solar Apex. Nature 162, 920 (1948).
   https://doi.org/10.1038/162920a0)
# solar apex
sun_ra = math.radians(270.2)
sun_dec = math.radians(28.7)

obs_dec = math.radians(obs_dec_2000)
obs_ra = math.radians(obs_ra_2000)
```

```
a = math.cos(sun_dec) * math.cos(obs_dec)
b = (math.cos(sun_ra) * math.cos(obs_ra)) + (math.sin(sun_ra)
   * math.sin(obs_ra))
c = math.sin(sun_dec) * math.sin(obs_dec)
v_rs = v_sun * ((a * b) + c)

v_lsr = corr + v_rs
print("LSR correction [km/s]: ", -v_lsr)
print("Positive value means receding (redshift) source,
   negative value means approaching (blueshift) source")
```