Observations of the Universal Dialtone Radio Astronomy Laboratory

Andrew Sheridan Sharks In Space Professor Aaron Parsons

February 27, 2018

Abstract

We record electromagnetic signals from hydrogen at two locations in the sky using the horn antenna on the roof of New Cambell Hall. We aim the horn at zenith and in the galactic ecliptic by performing spherical coordinate rotations. By correcting for the motion of the Local Standard of Rest we find HI velocities of these hydrogen sources to be 10.4, 29.0, 48.1, and 72.6 $\mathrm{km\,s^{-1}}$. By evaluating our system temperature we find the corresponding intensities to be 37.3, 179.0, 20.0, and 7.1 K.

1 Introduction

Hydrogen is the lightest and most abundant element in the Universe, composing about 75% of all matter. Due to its simple structure hydrogen has received a great deal of study. When physicists began exploring the quantum mechanical structure of neutral hydrogen, HI, it was discovered that the ^{1}s1 ground state would split into two extremely close 'hyperfine' energy levels. The split is due to the slight energy difference between the states of parallel and anti-parallel magnetic dipole orientation between the proton and the electron. The energy difference between the hyperfine levels is on the order of 5 µeV.

In a cloud of HI gas that has elements which have been excited to the higher hyperfine state, when the electrons decay to the lower state the resulting photon has a frequency of approximately 1421.4 MHz and a wavelength of approximately 21 cm. This wavelength allows these photons to penetrate both dust in space and the atmosphere on Earth, allowing us to observe the distribution of gas in the Universe. Because stars and galaxies are largely composed of hydrogen, a radio telescope that observes the HI 21 cm line can create maps of the Universe. The photon frequency is tightly dictated by the hyperfine energy difference, so any observed shift in frequency may be attributed to either the temperature of the HI source, or its motion.

We detail the signal chain used to capture data as well as methods to validate and smooth the collected data. We analyze HI signal data collected from zenith, adjusting for the instrumentation temperature and the velocity of the Local Standard of Rest. We repeat this analysis on HI signal data collected at coordinates in the galactic ecliptic.

2 Apparatus

The signal chain we employ is sketched out in Figure 1. The antenna collects radio frequency signals and passes them to a double sideband mixer (DSB). The horn antenna constructed by Carl Heiles & Co (1) on the roof of New Campbell Hall collects high frequency RF signals. The signal is passed through a K&L Microwave Inc 5B120-1380/160 Bandpass Filter (2), a 160 MHz range centered on 1380 MHz. From (2), the filtered signal is amplified in two 20 dBm amplifiers, (3) and (4). The amplified signal is combined (5) in a double sideband mixer with a local osicllator, (6), an Agilent N9310A RF Signal Generator set to 1230 MHz.

The signal has been shifted from 1420.4 megaHz to 190.4 megaHz, and is fed into a sideband separating mixer (SSB). The output of (5) is passed into a Telonic 190-22 Bandpass Filter (7), which has a 22 MHz range centered on 190 MHz. This is then amplified in (8), a MiniCircuits Amplifier ZFL-1000LN, and split in the MiniCircuits Splitter ZFSC 2-1 (9). The split signal is sent to a sideband separating mixer composed of two mixers (10) and (12). The second input to the SSB is the output of a local osillator (14), a Keysight 9310A RF Signal Generator set to 190 MHz. The Keysight signal is split in the Technical Research Manufacturing 90° splitter (15).

The signal has now been shifted down to 0.4 megaHz. The output from the SSB is passed through a variable bandpass filter (13), a Krohn-Hite Model 3202R Filter set to 2 MHz. Finally, the signal from the bandpass filters is collected using a PICO Technology PicoSeries 2000 digital sampler.

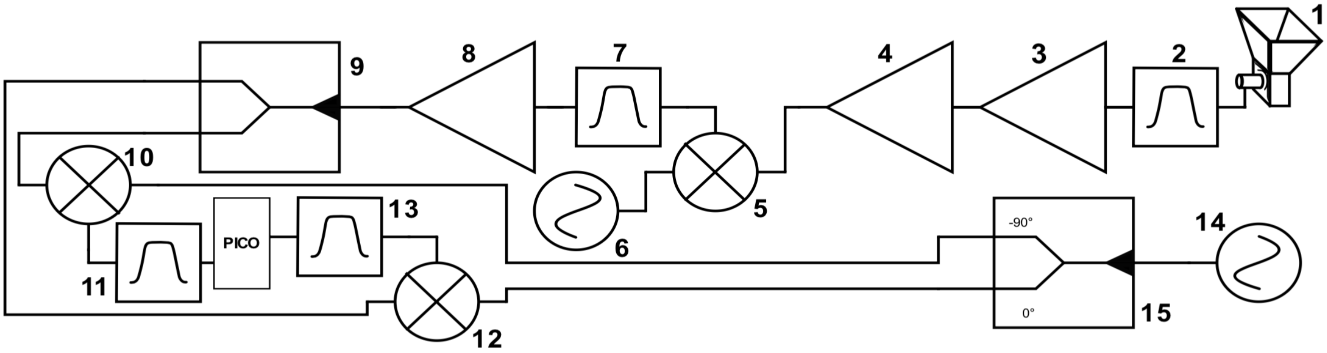

Figure 1: The signal chain employed. The signal captured by the horn antenna is amplified, filtered, mixed, and shifted. The signal data is collected with a PICO sampler.

3 Method

To validate our signal chain we first ensure that the signal we aim to receive will be passed through our band pass filter. The HI signal we wish to measure lies at approximately 1420.4 megaHz. Our first local oscillator is set to 1230 MHz using the `ugradio.agilent.SynthClient().set_frequency()` function. The second local oscillator is manually fixed to 190 MHz. When used in conjunction these shift the signal to 0.4 megaHz:

$$\begin{aligned}\nu_{\mathrm{sig}} &= \nu_{\mathrm{HI}} - \nu_{\mathrm{lo}_1} - \nu_{\mathrm{lo}_2}\\ &= (1420.4 - 1230 - 190)\ \mathrm{MHz}\\ &= 0.4\ \mathrm{MHz}\end{aligned}$$

Since the expected signal frequency of 0.4 MHz falls inside our final band pass filter range of ±2 MHz, the signal should not be filtered out. Before we view signal the we check that our sampling rate is not too low.

The PICO sampler has a maximum sample rate of 62.5 MHz. This maximum rate can be reduced by adjusting the `divisor` parameter in the `ugradio.pico.capture_data()` function. By passing an integer into `divisor`, we divide the maximum rate by that integer. From the resulting value, we must keep the associated Nyquist frequency sufficiently high. By letting `divisor` = 10, we meet this criterion.

$$\text{Nyquist frequency} = \frac{1}{2}\frac{\text{Max Rate}}{\texttt{divisor}} > \text{bandpass}$$

$$\frac{1}{2}\frac{62.5}{10}\ \text{MHz} = 3.125\ \text{MHz} > 2\ \text{MHz}$$

3.1 Saturation

In addition to setting an appropriate sample rate, we must adjust the `volt_range` parameter in `capture_data()`. This setting represents the range of amplitudes that the PICO sampler will correctly read. Because the sampler has only 256 distinct sample levels [1], setting this value too high results in severely quantized data, while setting this value too low causes the readings to saturate the sampler. The data must also be converted from the raw integers reported by the sampler to millivolts, which we do using our function `scale()`. Since we expect our signal to be low amplitude, we begin with `volt_range` at its lowest setting, and collect some sample data.

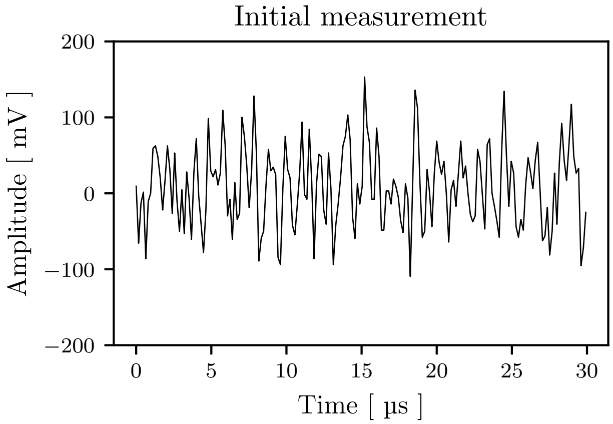

Figure 2: The first 30µs of collected amplitudes. We convert the sampled data to millivolts to check that our signal is not saturating the PICO sampler. The amplitudes fall inside our 400 mV$_{\mathrm{pp}}$ range so we conclude the setting is appropriate.

We increment `volt_range` until saturation is no longer visible. In Figure 2 we show how a setting of 200 mV affects the sampled signal. We see that in the first 30µs the signal does not saturate the sampler. When checked rigorously over 32000 elements, we find that ≈ 0.05% of the samples were at the maximum value.

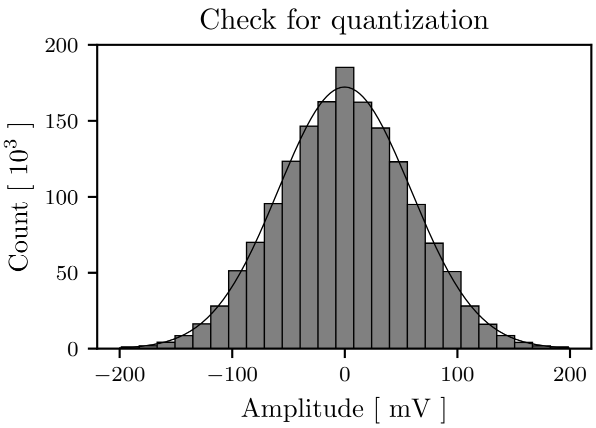

Figure 3: The first 1.6 million collected amplitudes. We create a histogram to check that our signal is being detected in a wide range of the PICO sampler bit depth. The even spread of values and lack of quantization suggest our signal chain is functioning correctly.

3.2 Quantization

The sampler has a limited number of distinct levels in which can report amplitudes. If we are using only a small subset of these levels, our data will exhibit severe quantization. In an alternative check of our sampling method, we collect 50 blocks, each represents 32000 samples. In Figure 3 we see that the first 50 blocks do not exhibit quantization. We perform a least squares to a Gaussian distribution on the histogram using `curve_fit()` from `scipy.optimize`. For the Gaussian function

$$g(x) = Ae^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

our fit returns the amplitude A, mean μ and standard deviation σ. For this initial set of blocks, we find $A = 172 \times 10^3$, $\mu = -0.177$mV, $\sigma = 58.9$mV. The fitted line closely matches our histogram, we conclude the noise we collect is mostly Gaussian. By collecting vast amounts of data we will be able to filter out this noise, revealing the low amplitude HI signal.

3.3 Sampler A/B correction

To ensure that the sampler is connected to our signal chain correctly, we input two test signals and check that they fall into the correct sideband. Our signal chain acts as a LO that shifts any incoming signal by 1420 MHz. The sampler settings result in a power spectrum frequency width of 6.25 MHz centered at zero. Signals higher and lower than 1420 MHz will fall into different sidebands. The upper sideband represents frequencies greater than zero, while the lower sideband represents frequencies less than zero. Using a HP Synthesized CW Generator we input signals at 1419 MHz and 1421 MHz into the horn antenna. The resulting signal frequencies should be

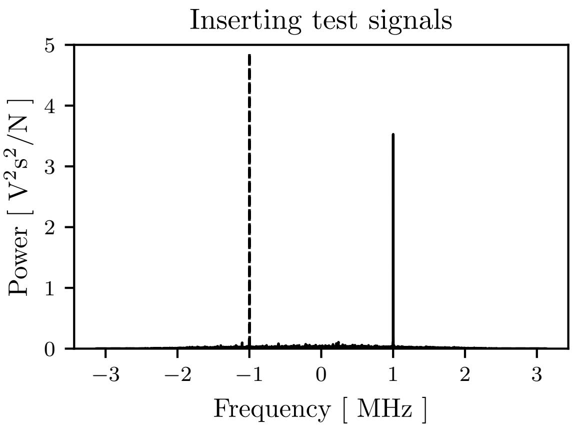

Figure 4: Power spectra for two test signals, 1419 MHz (left, dashed) and 1421 MHz (right). We compute the power spectra to check the PICO sampler input order. The lower frequency signal falls in the lower-sideband, and the higher frequency signal falls in the upper-sideband, confirming the input order is correct. The factor of 1/N in the units is a product of the transforms `ortho` parameter.

$$\begin{aligned}\nu_{\text{sig}} - \nu_{\text{lo}} &= \nu_{\text{PICO}} \\ (1421 - 1420)\text{ MHz} &= 1\text{ MHz} \\ (1419 - 1420)\text{ MHz} &= -1\text{ MHz}.\end{aligned}$$

The PICO sampler we use for data collection has two inputs, labeled **A** and **B**. When we pass `dual_mode = True` to `capture_data`, both of these input streams will be captured. We combine the two outputs into a complex array. We collect 50 complex blocks and compute the Fourier transform of each using `numpy.fft.fft()`, setting the optional parameter `norm = "ortho"`, which scales the transform by $1/\sqrt{N}$ where N is the number of elements. We compute the power spectrum of each block by taking the absolute square of each Fourier transform. Using `numpy.mean()` we find the mean power of the 50 blocks, shown in Figure 4. The frequencies are in the expected locations, confirming the inputs to the PICO sampler are in the correct order.

3.4 Noise reduction

The HI signal we aim to view has a width of approximately 10kHz. Each bin in our frequency space has a width determined by the number of elements, and the sample rate. With 16000 elements in our frequency space, the width of each bin is

$$\Delta\nu = \frac{6.125\text{ MHz}}{16000} \approx 0.38\text{ kHz}.$$

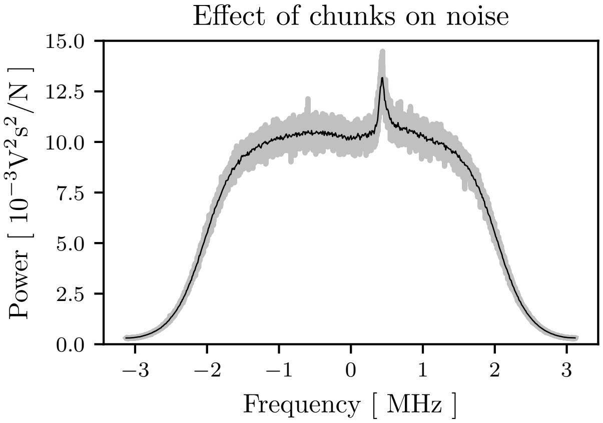

Figure 5: Mean power spectra of 1000 data blocks, computed with 25 chunks per block (black) and no chunks (grey). By breaking blocks of data into smaller chunks, we decrease both frequency resolution and noise.

By decreasing the number of elements, we increase the width of each bin and decrease the resolution. We have our sample rate at a fixed value, and so we change the number of elements in each power spectra. Since our data comes in complex blocks of 16000 samples, the number of elements we change to must be a multiple of 16000. If we divide each block into 25 chunks of 640 elements, we arrive at a bin width of

$$\Delta\nu = \frac{6.125 \text{ MHz}}{640} \approx 9.57 \text{ kHz}.$$

We collect 1000 blocks of data with the horn directed towards zenith. In Figure 5 we compare the effect of changing the number of chunks. With 25 chunks per block, the peak near 0.4 MHz is clearly defined, and the noise is reduced significantly. Based on this we maintain the number of chunks at 25 throughout.

4 HI at zenith

We point the horn straight up, at zenith, and collect two sets of 1000 complex blocks, computing the mean power of each set. One set is collected with the DSB LO set to 1230 MHz which will keep the HI signal in the upper sideband. Another set is collected with the SSB LO set to 1231 MHz, which will move the HI signal into the lower sideband. From these we extract two shapes, S_{offline} and S_{onfline}. By taking the lower sideband portion of the 1231 MHz signal and appending to it the upper sideband portion of the 1230 MHz signal we produce the shape S_{offline}, Figure 6. We then take the lower sideband of the 1230 MHz signal and append to it to the upper sideband of the 12301MHz signal to produce S_{online}, Figure 7.

We note that in Figure 6 there is an amplitude shift at the zero frequency point. This occurred whenever

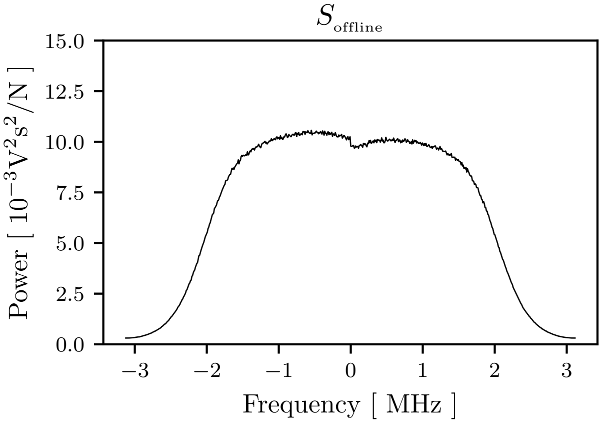

Figure 6: The shape resulting from our signal chain. By combining the lower sideband and upper sideband of two signals, we can derive the signal shape generated by our instrumentation noise. We use this to reveal the corrected shape of the HI signal.

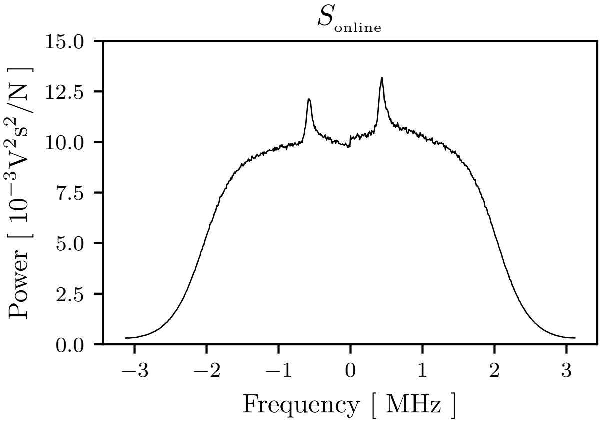

Figure 7: The uncorrected shape of the HI signal. By combining the lower sideband and upper sideband of two signals, we can derive the HI signal shape generated on top of our instrumentation noise. We use this to reveal the corrected shape of the HI signal.

we collected data using two different DSB LO settings. We suspect the 2 MHz bandpass filter may not be symmetric about zero, though there may be alternative explanations. Due to this amplitude shift, we focus our attention on the upper sideband.

4.1 HI shape

To find the corrected HI signal shape, we divide the uncorrected HI signal shape by the instrumentation shape:

$$\text{HI shape} = \frac{S_{\text{online}}}{S_{\text{offline}}}$$

The resulting shape in Figure 8 represents the relative strength of the HI signal compared to our instrumentation.

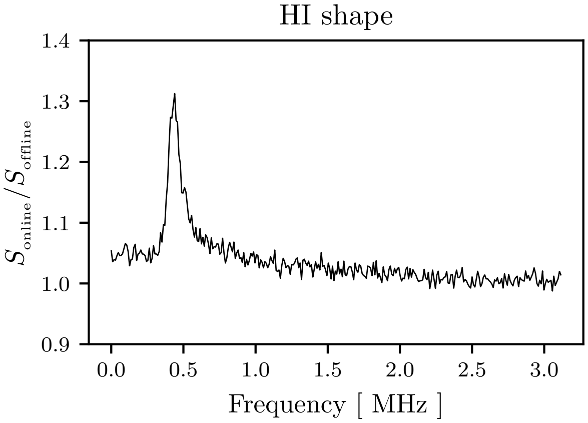

Figure 8: The shape of our HI signal. We divide the two extracted signal shapes, S_{online} and S_{offline} and isolate the upper-sideband. The result is the shape of our HI signal.

4.2 Intensity Calibration

To find the intensity of the HI signal, we must calibrate our measurements to our instrumentation. We collect two sets of 250 blocks, each with the DSB LO set to 1230 MHz, and find their smoothed lines. For one set, S_{coldsky}, we aim the horn at zenith. For the other set, S_{300K}, we aim the horn at two average-sized human beings. Human beings are rather warm compared to the sky, and we take their relative intensity as $T_{\text{cal}} = 300$ K. To find the intensity of the instrumentation we combine the S_{coldsky} and S_{300K} spectra

$$\begin{aligned} T_{\text{sys, coldsky}} &= \frac{\sum S_{coldsky}}{\sum \left(S_{\text{300K}} - S_{coldsky}\right)} \left(T_{\text{cal}} - T_{\text{sys, coldsky}}\right) \\ &\approx 136.3 \text{ K.} \end{aligned}$$

And to find the calibrated HI signal in Figure 9 we simply multiply the HI signal by the system intensity:

$$\text{HI signal calibrated } = \text{HI signal} \times T_{\text{sys, coldsky}}$$

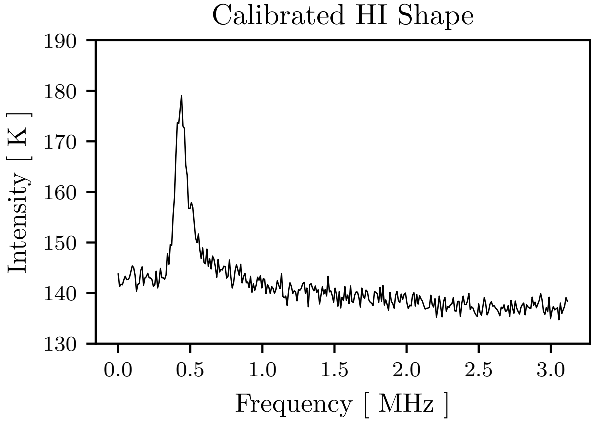

Figure 9: The calibrated shape of the HI signal. By multiplying the HI shape by $T_{\text{sys, coldsky}}$ we arrive at the calibrated shape. If we were so inclined, we could infer many properties of the hydrogen cloud from the intensity, such as cloud density.

4.3 Hydrogen velocity

All that remains is to connect the observed frequency of our HI signal to a velocity. In this two step process we first make a coarse adjustment to our frequency range. By astronomical convention, velocity away from the observer is positive, so

$$\frac{v_{\text{dop}}}{c} = -\frac{\Delta f}{f_0}$$

where c is the speed of light, f_0 is the actual predicted frequency of the HI signal, 1420.4 MHz, and Δf is the frequency difference:

$$v_{\text{dop}} = -c \left[\frac{f_0 - (1420 + \text{freqs})}{f_0} \right]$$

The second step is a small correction to the velocity v_{dop}. Using `ugradio.doppler.get_projected_velocity()` we take into account the Earths motion and correct to the Local Standard of Rest (LSR), which takes as parameters the celestial coordinates where we are pointing the horn, the Julian date of the observation, and the coordinates of the observatory. From this wonderful piece of software we find that the velocity correction to be:

$$v_{\text{cor}} \approx -20.7 \text{ km s}^{-1}$$

When we add this correction to our v_{dop}, we arrive at can plot the intensity of our HI signal versus velocity, Figure 10. We extract the velocity associated with the peak intensity of 179.0 K and find the observed source of hydrogen is moving away from us with a velocity:

$$v_{\mathrm{HI}} \approx 29.0\ \mathrm{km\,s^{-1}}$$

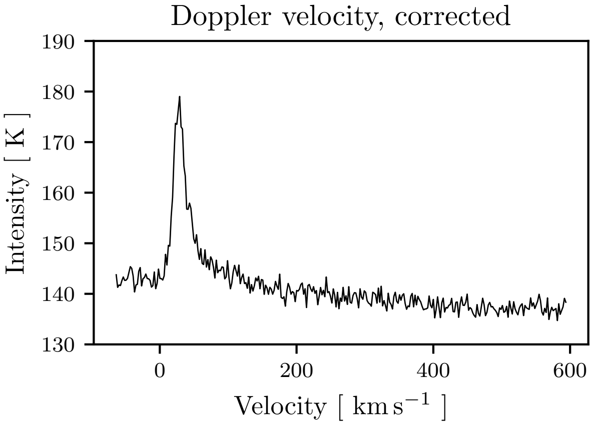

Figure 10: The shape and corrected velocity of the HI signal at zenith. We correct to the LSR to obtain the corrected velocity, and find that the cloud has an approximate velocty away from us of 29.0 $\mathrm{km\,s^{-1}}$.

5 Hydrogen in the direction of the galactic center.

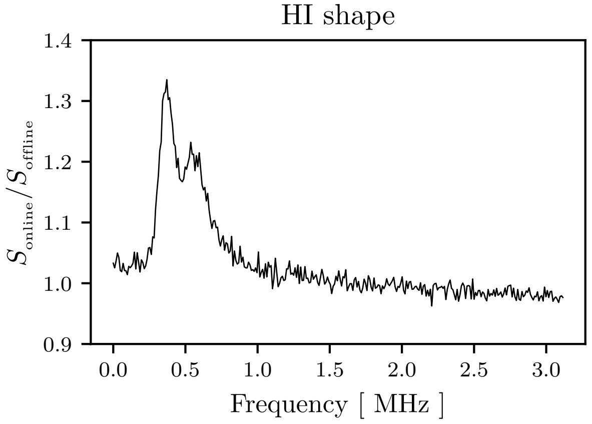

Figure 11: The shape of the galactic HI signal. By dividing S_{online} by S_{offline} the relative strength of the galactic HI signal is revealed.

To collect HI signals from the galactic ecliptic we convert the galactic coordinates $(\ell, b) = (120°, 0°)$ to local azimuth-altitude coordinates. We covert coordinates via spherical rotation matrices, Appendix refsec:coords. We point the horn to the galactic coordinates $(\ell, b) = (120°, 0°)$ and collect 1000 blocks. Following the same prescription as §4.1 and §4.2, we find the calibrated HI signal shape, Figure 11. To eliminate the curved baseline we perform a quadratic fit on the curved baseline present in the HI signal, using `numpy.polyfit()` and `numpy.polyval()`. By subtracting off this quadratic baseline and multiplying the shape by $T_{\mathrm{sys,\ coldsky}}$, we arrive at an adjusted calibrated shape for our galactic HI signal. We find a new velocity correction:

$$v_{\mathrm{cor}} \approx -17.4\ \mathrm{km\,s^{-1}}$$

With this correction we arrive at a plot of the intensity of our galactic HI signal versus velocity Figure 12. The galactic HI signal appears to have multiple peaks. We suspect the peaks are an indication of a Gaussian distribution of velocities, perhaps different clumps of a cloud moving towards or away from us. We compute a least squares fit of a polynomial to the adjusted galactic HI signal shape, using `ugradio.gauss.guassfit()`. This polynomial is overlaid on the galactic HI signal shape in Figure 13. The fitted polynomial in Figure 13 is the sum of three normal Gaussian distributions. Each of the three distributions represent some different HI signal, perhaps from different gas clouds, or portions of the same cloud moving at different velocities. To see how these combine into our polynomial, we plot them each separately along with their sum, Figure 14. For each Gaussian we can extract the peak intensity, mean velocity, and a range of velocities 5.

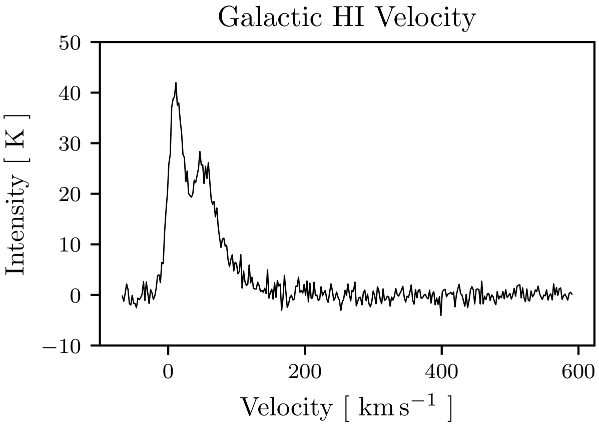

Figure 12: The adjusted shape of the galactic HI signal. By following the procedure as before, we find the intensity and velocity of the galactic HI signal.

6 Conclusion

In this experiment we were able correctly trace a signal down our signal chain and predict its behavior. For example, by adjusting one of the local oscillators we can

Intensity [K]	Mean Velocity [$\mathrm{km\,s^{-1}}$]	Velocity Range [$\mathrm{km\,s^{-1}}$]
37.3	10.4	0.3 - 20.5
20.0	48.1	31.1 - 65.0
7.1	72.4	37.0 - 107.8

Table 1: Intensity and velocities of the three HI sources observed at galactic coordinates (ℓ, b).

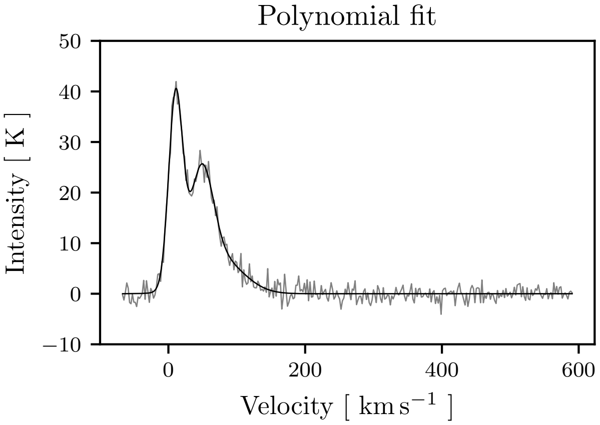

Figure 13: Polynomial fit to the adjusted galactic HI signal. We compute a least squares fit to the data and show it is well described by a polynomial. The fit paremeters will let us extract velocity ranges for different elements in the HI cloud we observe.

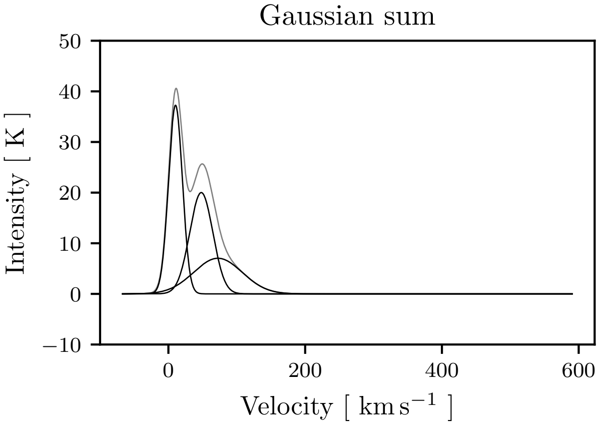

Figure 14: Three Gaussian distributions (black) and their sum (grey). We confirm that the polynomial fit to the HI signal is the sum of three Gaussian distributions.

shift the signal to a desired frequency. By breaking the data into small chunks we were able to significantly decrease noise while maintaining a minimum resolution. When aiming the horn antenna at different locations we are able to discern distinctive HI signal peaks near the theoretical frequencies. Using provided lab software we computed the velocities of those HI peaks to be 10.4, 29.0, 48.1, and 72.6 $\mathrm{km\,s^{-1}}$.

7 Acknowledgments

We would like the thank Professor Aaron Parsons and his graduate student instructors Deepthi and Kara for their patience in explaining the methods used to capture and analyze data. Also, discussions with numerous other students were essential to resolving coding issues.

In our group, Sharks in Space, Steven guided data collection and provided sanity checks, Amanda diligently maintained organization with Google Sheets, while Hayley determined what would go in plots and I created the code to collect and parse data.

We each analyzed the data individually. Any similarities in plots are due to the use of the same data set and a mutual atheistic.

The code is available on my account at ugradiolab, though I warn it is in my opinion rather fragile and poorly documented. The data analysis was performed partly in Python 2 and partly in Python 3. In retrospect I did not stay as personally organized as I should. Also much of the data had to be retaken multiple times due to both machine and human error. If I was to repeat this experiment I would perform automated checks on the collected data with every collection, not only on the first sets of collected data.

Appendix A Coordinate Transformation

To convert from one coordinate system to another we employ rotation matricies as outlined in [2]. By multiplying the $\vec{x}$ in one coordinate system by a rotation matrix R, the result is $\vec{x}'$ in the new coordinate system.

$$\vec{x}' = R\vec{x}$$

We initialize $\vec{x}$ by converting the longitude *long* and latitude *lat* in one coordinate system to rectangular coordinates.

$$\begin{bmatrix} \cos(lat) \cdot \cos(long) \\ \cos(lat) \cdot \sin(long) \\ \sin(lat) \end{bmatrix}$$

The rotation matrices R can be multiplied together to move between multiple coordinate systems. In the notation below, the transformation fraction subscript on the rotation matrices follows the format

$$\frac{\text{numerator}}{\text{denominator}} = \frac{\text{ending coordinate system}}{\text{starting coordinate system}}.$$

The following matrices were employed.

$$R_{\frac{ha,\delta}{\alpha,\delta}} = R_{\frac{ha,\delta}{(\alpha,\delta)_2}} \cdot R_{\frac{ha,\delta}{(\alpha,\delta)_1}}$$

$$R_{\frac{ha,\delta}{\alpha,\delta)_2}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{\frac{ha,\delta}{(\alpha,\delta)_1}} = \begin{bmatrix} \cos(\mathrm{LST}) & \sin(\mathrm{LST}) & 0 \\ -\sin(\mathrm{LST}) & \cos(\mathrm{LST}) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{\frac{ha,\delta}{\alpha,\delta}} = \begin{bmatrix} \cos(\mathrm{LST}) & \sin(\mathrm{LST}) & 0 \\ \sin(\mathrm{LST}) & -\cos(\mathrm{LST}) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{\frac{ax,alt}{ha,\delta}} = \begin{bmatrix} -\sin(\phi) & 0 & \cos(\phi) \\ 0 & -1 & 0 \\ \cos(\phi) & 0 & \sin(\phi) \end{bmatrix}$$

$$R_{\frac{\ell,b}{\alpha,\delta}} = \begin{bmatrix} -0.054876 & -0.873437 & -0.483835 \\ 0.494109 & -0.444830 & 0.746982 \\ -0.867666 & -0.198076 & -0.455984 \end{bmatrix}$$

For example to convert from galactic coordinates to local azimuth altitude we do

$$\vec{x}_{az,alt} = R_{\frac{ax,alt}{ha,\delta}} R_{\frac{ha,\delta}{\alpha,\delta}} R_{\frac{a,\delta}{\ell,b}} \vec{x}_{\ell,b}$$

While in reality we multiply the matrices together, we can instead imagine the fractions canceling, to produce the resulting R

$$\vec{x}_{az,alt} = R_{\frac{ax,alt}{\cancel{ha,\delta}}} R_{\frac{\cancel{ha,\delta}}{\cancel{\alpha,\delta}}} R_{\frac{\cancel{a,\delta}}{\cancel{\ell,b}}} \vec{x}_{\cancel{\ell,b}}$$

$$\vec{x}_{az,alt} = R_{ax,alt} \vec{x}$$

REFERENCES

[1] Pico technologies spec sheet. https://www.picotech.com/oscilloscope/2000/picoscope-2000-specifications. Accessed: 2018-02-20.

[2] Carl Heiles and Deepthi Gorthi. Spherical/astronomical coordinate transformation, February 2018.