Waves to Ones and Zeros:
Data Acquisition in Radio Astronomy

Ryan Thornton (Marmion Academy)

Dr. Christopher Stoughton (Fermi National Accelerator Laboratory)

Abstract

I designed the data recording and analysis of the QuarkNet Radio Telescope, a telescope designed measure hydrogen's emission of radio waves of around twenty-one centimeters. The telescope is meant to map the hydrogen in the arms of the Milky Way galaxy, and is composed of a two-meter disk, a feed horn configured to twenty-one centimeter waves, an Airspy receiver to collect the data and send it to the Raspberry Pi, two linear actuators to rotate the dish, and a Raspberry Pi 3 to control the Airspy and motors and to record data. We intend the QuarkNet Radio Telescope to be inexpensive enough for enterprising high school teams to repeat the procedures and build a telescope.

I created modules in the GNU Radio program with the programing language Python to average data using the Welch method and record the data to hdf5 files. To store the data, I set up a data storage system to transfer data from the Raspberry Pi 3 to the virtual server acting as a control and storage system. I then created the data analysis program using Python to display the data in a waterfall plot format.

Introduction

Background

Radio astronomy as a field came about due to the accidental discovery of Karl G. Jansky. While trying to design an antenna for Bell Telephone Laboratories' transatlantic telephone plans, Jansky found noise he could not explain. After accounting for thunderstorms in the area

and the Sun, Jansky identified the source of the noise as the Milky Way. He published his findings in 1933 with little attention. In 1937, Grobe Reber built the first parabolic radio telescope, and published the first radio frequency sky map in 1944. Reber configured his telescope to search for a smaller wavelength than Jansky's, 1.87 meters rather than 14.6 meters. Radio astronomy developed after World War II, becoming a prominent method of observing the universe.

Radio telescopes gather radiation that penetrates the Earth's atmosphere, direct it to a central feed horn, and use an electronic filter to amplify and record the radiation. Even through amplification, radio signals received from space are minimal compared to ambient noise. Interferometry involves arraying two or more telescopes together allows the strength and precision of the signal to be increased simultaneously. Interferometry allows telescopes with a small collecting area to be more effective (Miller 1998).

Context of the Work

The QuarkNet Radio Telescope (QRT) functions as a small and inexpensive telescope to map neutral hydrogen in the universe. Neutral hydrogen emits radiation when it changes dipole states with a frequency of around 1,420.405 MHz (and a wavelength of 21.106 cm). This known frequency allows radio astronomers to detect and map hydrogen in the universe.

The QuarkNet Radio Telescope is designed to be small enough and inexpensive enough that it can be replicated by high schools. This will allow telescopes to be set up in an array and map the sky with greater precision. The QuarkNet Radio Telescope and its high-school-built counterparts will map the hydrogen in the universe and observe various astronomical phenomena, including stars and galaxies.

The current phase of the project is to create a functioning movement system and data acquisition system. It serves to pioneer the way for future QuarkNet Radio Telescope-like telescopes to be built and added to become an interferometer.

Methods

Equipment

The telescope contains an 8 foot parabolic dish, a feed horn, an antenna, a preamplifier, an Airspy receiver, a Raspberry Pi 3, 2 linear actuators, and a relay. The feed horn is a small, 3-D printed tube lined with copper tape configured for 21 cm radio waves, and directs the waves to a quadrupole antenna at its center. The antenna is an array of four copper poles that allows for polarized electromagnetic waves to be collected and transmitted to the Airspy as an electric signal. The antenna is connected to a preamplifier, which boosts the low noise signal. This signal goes to the Airspy, which measures the beats against its local oscillating frequency. A radio receiver can measure accurately up to half of its sampling frequency, called the Nyquist frequency, on both sides of its local oscillating frequency. The Airspy has a sampling frequency of 10 MHz, which means it can measure signals within 5 MHz below and above the local oscillating frequency. To include the 21 cm line in an accurate area, we set the local oscillator to run at 1.42 GHz. The linear actuators and relay move the dish to various positions.

The Raspberry Pi 3 runs a flavor of GNU/Linux called Raspian, an operating system configured for the ARM architecture of the Pi. On top of the kernel, the Raspberry Pi has the programing language Python version 2.7 with various modules, a program called GNU Radio with modules, and hosts an Secure Shell server. GNU Radio is an open source framework designed to make signal processing applications. It comes with built-in modules called blocks to assist in signal processing, and allows users to make their own modules using the programing languages Python or C++. Users arrange the blocks into flowcharts, which run various processes in the flowchart order. A simple example of this would be a source block to receive data followed by an averaging block to shrink the data down to a more manageable size concluded by a block that displays the data or writes the data to a file. A module another team created that we use is the Osmocom source block, which allows the input from the Airspy to be read into a GNU Radio flow chart. I created two modules for signal processing, an averaging module to reduce noise and data consumption, and a block to write the data to an HDF5 file. I used Python to create the modules, using the modules NumPy, SciPy, and H5py. The code itself can be found on GitHub (https://github.com/adamanderson/QRT).
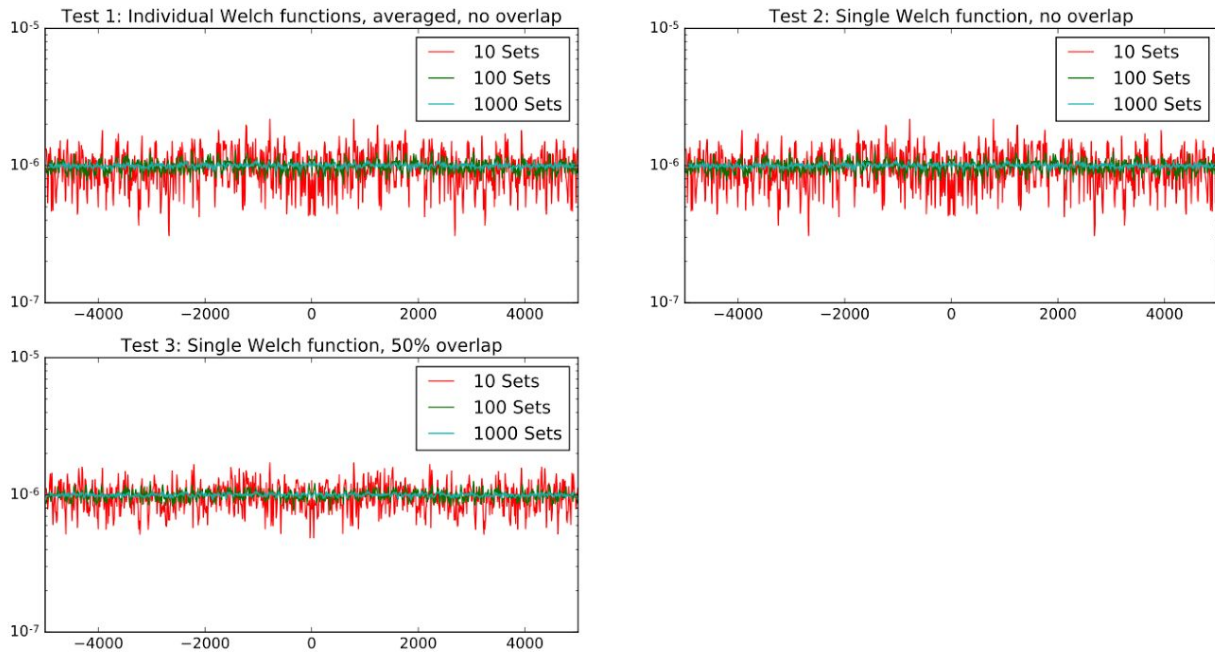
Data

Data is collected through the GNU Radio program. The Osmocom Source block uses the Airspy firmware to collect the data from the telescope. This data is then sent to the custom built

"welch" block. The welch block uses a signal processing method from the Scipy module of Python to take a chunk of data and use the Welch method to average it into one Fourier Transform. A Fourier Transform turns a function of time (a series of noisy waves) into a function of frequency (a graph of amplitude vs frequency). A Discrete Fourier Transform allows a non-infinite segment of a function of time to be transformed into a function of frequency. The Welch method takes a segmented average of multiple Discrete Fourier Transforms to allow for a better average than an arithmetic mean of the data (Heinzel 2002). This reduces more noise than just averaging, and allows for the signal to be seen. Noise is reduced as it balances itself out over a period of time, appearing as a flat line. After the data is averaged using the Welch method, it is again averaged in the welch block. This is because using the Welch method takes more CPU power on the Pi than simple averaging. If the Pi uses the Welch method on all the segments of data for a second's time, all available CPU power is used. The use of this block allows 10,000 samples per second to be reduced to a single Fourier Transform with as much precision as possible, reducing the data size by a factor of 10,000 and reducing noise.

After the data is processed using the Welch method, it is written to an HDF5 file. The Hierarchal Data Format 5 file format supports metadata, which allows important data to be written to the file in addition to the signal. The metadata stored in the HDF5 file includes the telescope's identifier, the time and date, the latitude and longitude, the right ascension and declination coordinates of the telescope's direction, the sampling rate of the Airspy, the local oscillating frequency of the Airspy, the scaling of the Fourier Transform created, the length of the Fourier Transform created, and the amount of simple averages. This metadata helps in the analysis of the signal processing, as one can know the where in the sky the telescope was pointing and also information pertinent to plotting the power spectra.

The data is then sent to a virtual server on the Fermilab network via Secure Shell. This allows users to access the data and interpret it. The Raspberry Pi is controlled remotely via Secure Shell and the Pyro function of Python. This allows the motors and data acquisition to work in tandem.

Analysis

To test the effectiveness of the Welch method as opposed to simple averaging, I created a python program that tested the Welch method with no overlap against simple averaging against

the Welch method with 50% overlap. The following figure shows that the same data appears the same when averaged and when the Welch method is used with no overlap. However, noise is significantly reduced when a 50% overlap is used.

After data is collected, it is less useful to view individual Fourier Transforms, so a waterfall plot is used to map intensity to frequency to time. An example of this is the next figure. To plot frequencies accurately, I removed the first and last bins of the Fourier Transform, as the amplitudes are significantly higher due to the noise of the local oscillator.
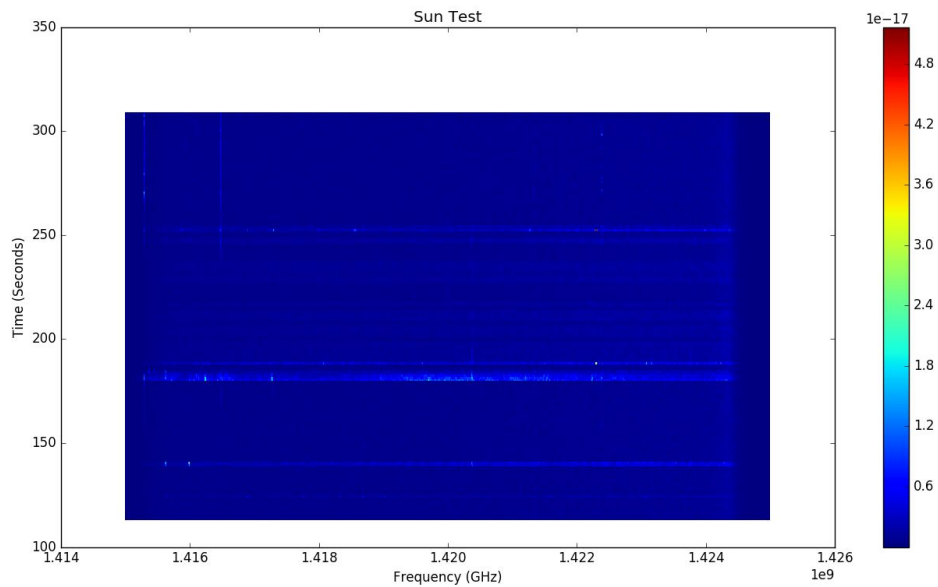
This waterfall chart shows the telescope scanning to and from the sun. From approximately second 180 to second 250 the overall noise level is higher, showing that the Sun was detected and affected noise levels.

Results

Data acquisition software is an integral part of a functioning radio telescope. The firmware to run the Airspy and record data is functioning, which will allow the QuarkNet Radio Telescope team to begin recording real data. The next step in the process is to combine the motor control system with the data acquisition system and to record at various known astronomical objects such as the moon and other planets to look at the strength of their signals.

Acknowledgements

Adam Anderson

Donna Kubik

Bennett Hardin

Lucas McDonald

Dave Carrano

Jonathon Groom

Danila Colosi-Carrano

Victor Pinks, II

Keith Mokry

Matt Grendzinski

Bryon Karger

Matt Hotham

References

1. Ewen, H. I., Purcell, E. M. (1951). Observation of a Line in the Galactic Radio Spectrum: Radiation from Galactic Hydrogen at 1,420 Mc./sec. *Nature* 168, 357–358.
2. Heinzel, G., Rüdiger, A., & Schilling, R. (2002, February 15). Spectrum and spectral density estimation by the Discrete Fourier Transform (DFT), including a comprehensive list of window functions and some new flat-top windows.
3. Miller, Diane F. (1998, April). Basics of Radio Astronomy for the Goldstone-Apple Valley Radio Telescope.