SAGI Summer School 2024: IC/DAQ + Electronics Project

Sam Condon, Logan Foote, and Hien Nguyen

July 31, 2024

Abstract

This document describes the instrument control and data acquisition (IC/DAQ) and electronics combined project for the 2024 edition of the SAGI Summer School at ICISE. In the project, we will build passive and active bandpass filter circuits, essential electronic components in all radio receivers, and we will use Python to perform automated measurements of the frequency response of the circuit. After measuring the frequency response, we will inject noise into a single frequency signal using an adder circuit to simulate a signal from a real antenna. We will measure the response of the circuit to this noisy signal to check the performance of our circuit. We will learn how to organize measurement automation and analysis code, and will use numpy's (Python package for matrix manipulation) Fast Fourier Transform algorithm extensively.

1 Introduction and Background

1.1 Motivation: Heterodyne Receiver for 21 cm Signal

Let's imagine we are building an instrument to observe the $\lambda = 21.106$ cm emission line produced by a spin-flip transition in neutral hydrogen. This line has been observed for decades and has revealed information about the structure and dynamics of the milky-way galaxy as well as information about how the universe was re-ionized. In vacuum, the 21 cm line corresponds to a frequency of:

$$f = \frac{c}{\lambda} = \frac{299792458\frac{\mathrm{m}}{\mathrm{s}}}{(21.106 \cdot 10^{-2})\mathrm{m}} = 1420.4 \text{ MHz} \tag{1}$$

Let's think about the propagation of the 21 cm signal at multiple stages of the observation. When the spin-flip first occurs, we get a pure signal at 21 cm. The corresponding signal in the time and frequency domain is shown in Figure 2. After propagating through space, arriving at the telescope, coupling to the transmission line, then stimulating the receiver electronics, we pick up noise from the sky and our electronics. The 21 cm signal with noise is shown in Figure 3.

The receiver electronics performs the job of turning the noisy analog signal from the sky into a digital signal which we can analyze on our computers. Recall that digital

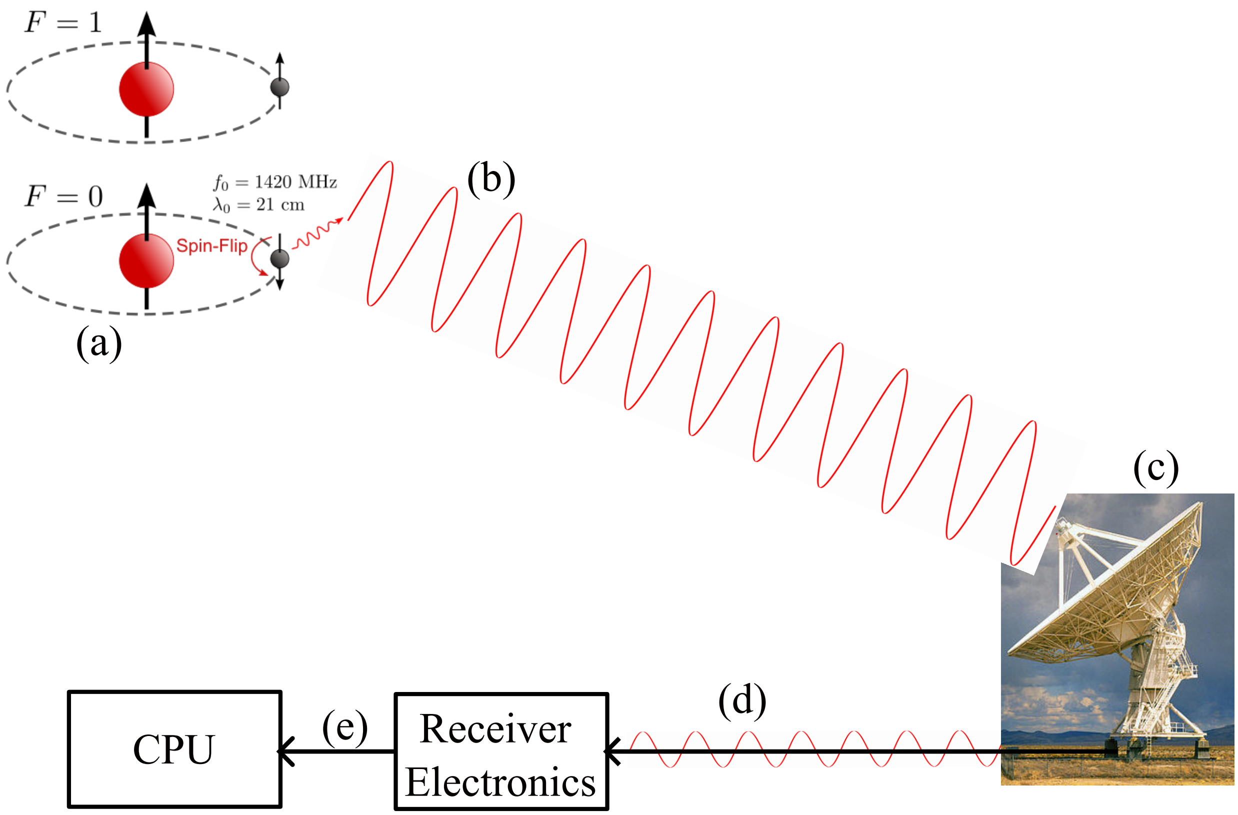

Figure 1: Simplified schematic of a 21 cm observation: (a) Spin-flip transition in neutral hydrogen in our galaxy produces electromagnetic radiation at a frequency of $f \approx 1420$ MHz. (b) The radiation travels to earth. (c) We observe the radiation with a dish or horn antenna. (d) The electromagnetic signal received by the dish or antenna is coupled to transmission lines into our receiver electronics. (e) The receiver electronics *demodulates* and *samples* the transmission line signal so that we can store it on our computer for scientific analysis.

signals consist of a discrete set of points rather than a continuum. To obtain this discrete set we must *sample* the original signal. From the lecture on sampling and aliasing (*sagi/lessons/instrument_control/lesson5/SamplingAliasingDft.pdf*), we know that in order to accurately represent a continuous signal from a set of samples, we must sample faster than the **Nyquist frequency** which is twice the maximum frequency in our signal of interest. Denoting our sampling frequency as f_s and the maximum frequency of interest as f_m, we must choose f_s to obey the following equation:

$$f_s > f_{\text{nyquist}} = 2f_m \tag{2}$$

For the 21 cm signal, we have $f_m = 1.42$ GHz so we must sample with:

$$f_s > 2.84 \text{ GHz}. \tag{3}$$

Historically, low noise and high resolution analog-to-digital converters that work up to 2.84 GHz either don't exist or are extremely expensive. We need a way to bring the 1.42 GHz

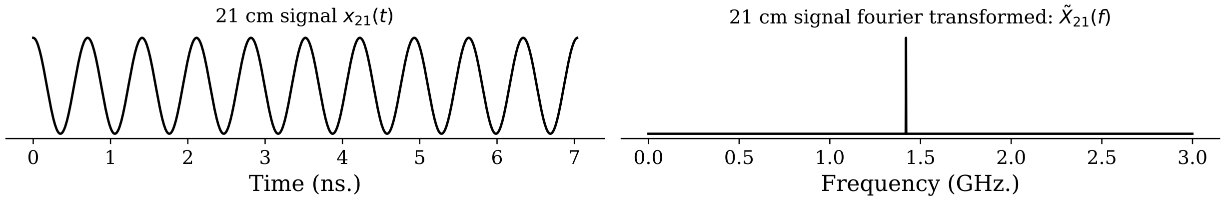

Figure 2: Pure 21 cm signal immediately after a spin-flip transition. On the left is the intensity of the electromagnetic field at a fixed position as a function of time $x_{21}(t)$. On the right is the Fourier Transform of this signal $\tilde{X}_{21}(f)$. The pure signal is a sinusoid at $f \approx 1.42$ GHz, which we observe as a single peak in the frequency spectrum.

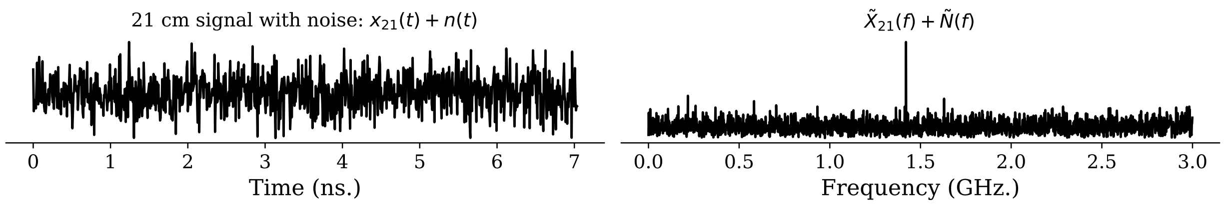

Figure 3: 21 cm signal with noise from the sky and instrument added. Although the original signal is tough to make out by eye in the timestream, we still see a strong peak at $f = 1.420$ GHz in the frequency spectrum.

signal to a lower frequency to allow nyquist sampling at a lower rate. We can achieve this in our receiver electronics with *mixing*. Consider the trigonometric identity for multiplication of two cosines at frequencies f_1 and f_2 [1]:

$$\cos(2\pi f_1 t) \cdot \cos(2\pi f_2 t) = \frac{1}{2}\left[\cos(2\pi(f_1 - f_2)t) + \cos(2\pi(f_1 + f_2)t)\right] \tag{4}$$

Notice that if we take $f_1 = f_2 = f_{21,\text{cm}}$ we obtain:

$$\cos(2\pi f_{21,\text{cm}} t) \cdot \cos(2\pi f_{21,\text{cm}} t) = \frac{1}{2}\left[\cos(2\pi(0)t) + \cos(2\pi(2f_{21,\text{cm}})t)\right] \tag{5}$$

$$= \frac{1}{2}\left[1 + \cos(2\pi(2f_{21,\text{cm}})t)\right] \tag{6}$$

Notably, we obtain two new frequencies at $f = 0$ Hz and $f = 2f_{21,\text{cm}}$. Let's multiply our noisy 21 cm signal with a sinusoid at the same frequency $f = 1420$ MHz. The result is shown in Figure 4.

Finally, we can extract the amplitude of the original 21 cm signal, which has been shifted to high and low frequencies, by *low-pass filtering* the signal. Our result should be a dc signal, with a bit of noise, that we can simply average at a much lower sample rate to get the strength of the 21 cm line emission! The frequency response of an example low-pass

[1] Can you show the corresponding identity in the frequency domain using the convolution and multiplication correspondence of the Fourier Transform?

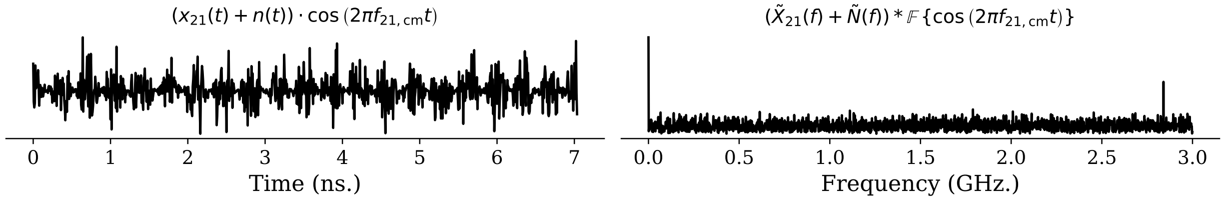

Figure 4: Noisy 21 cm signal multiplied with a sinusoid at $f = 1420$ MHz. As seen, we obtain peaks at 0 Hz and at $2 \cdot 1420 = 2840$ MHz, as predicted by equation 6

third-order butterworth filter is plotted over the multiplied signal in Figure 5. The signal in the time and frequency domain after low-pass filtering is shown in Figure 6. As seen, by mixing and low-pass filtering, we take a noisy signal at a high frequency and convert it to a low frequency signal that we can easily sample.

The takeaway here is that mixing and filtering are essential operations to perform measurements of astronomical signals! In this lab, we will gain experience with filter design, along with some of the basic measurement techniques used to construct receiver electronics to observe signals like the 21 cm line emission.

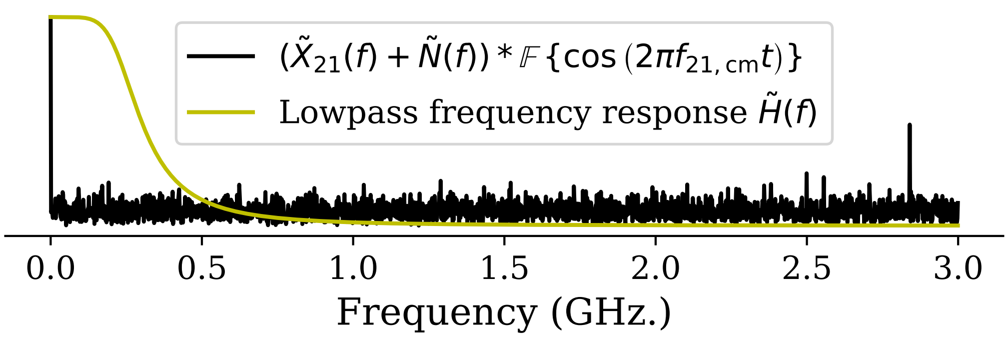

Figure 5: Frequency response of a 3rd order butterworth low-pass filter to get rid of high frequency components and find the 21cm signal magnitude. The frequency response is plotted against the mixed signal.

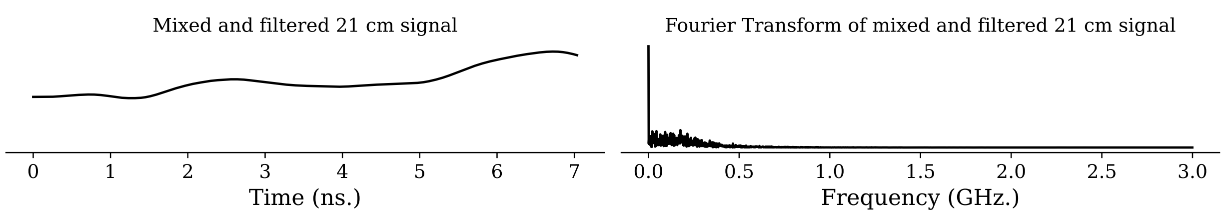

Figure 6: Filtered and mixed 21 cm signal.

1.2 Measurement Schematic

Sometimes, it is desirable to filter the sky signal before mixing. In this case we use a bandpass filter instead of a low pass filter because we have not yet mixed the signal to a lower frequency. We will build such a filter and measure the frequency response with the measurement schematic shown in Figure 7. We will send a series of fixed frequency signals with a function generator into the circuit and observe the output with an oscilloscope.

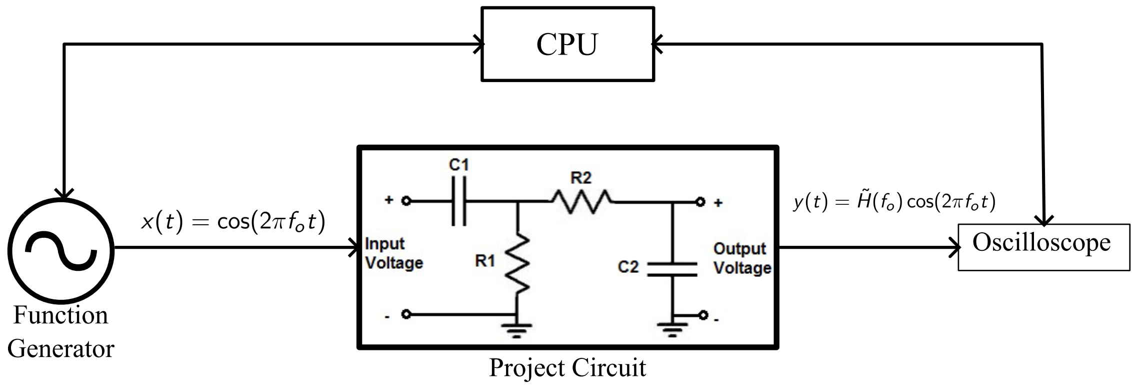

Figure 7: Filter frequency response $\tilde{H}(f)$ measurement schematic.

2 Instructions

We will be developing code to measure the frequency transmission through a circuit using a function generator and oscilloscope. First, we will write instrument control software to set the output waveform of the function generator and capture data using the oscilloscope. Then we will write the measurement procedure that uses the instrument control software to perform a sweep over frequencies and extract the transmission at each frequency. Finally, we will build a bandpass filter and use our measurement procedure to measure the transmission. Along the way, we will make sure to log important information so we have a record of the measurement setup.

2.1 Instrument control

First, we will write code to interface with the oscilloscope and the function generator. You can work through these notebooks in any order. For the oscilloscope, start with *siglent_sds1104xe_oscope.ipynb*. For the function generator, find the model number of the generator you are using and navigate to the corresponding notebook.

2.2 Frequency sweep measurement procedure

Next, we will set up a procedure to run the frequency sweep. Open *freq_sweep.ipynb* and follow the instructions.

2.3 Some notes

We have performed our measurement by saving the full oscilloscope timestream for each frequency, then later importing each of them and performing the FFT and peak extraction. Of course, the data would take up much less space if we first perform the FFT and peak extraction, and only save the transmission versus frequency data. Additionally, our code would run much faster, since the majority of the run time is taken up by saving and loading data. In practice, it is good to save all the data until you have performed the measurement several times, and you feel confident that you can throw away the timestream data. If you can afford the space and time, you might as well continue to store all the data in case you find an error or make an improvement to your analysis code down the line.

2.4 Bandpass filter measurement

Design and build a bandpass filter with a transmission in the 10-50 kHz range and rejection outside this range. If you would like to skip the design portion, check out the circuit model in *bandpass_filter.pdf*. If you would like to skip building the circuit, ask us for a pre-built circuit. Open *filter_measurement.ipynb* and follow the instructions to measure the transmission through your circuit.

2.5 Next steps

You can choose one or more of the following options for the remainder of the lab.

1. **Circuit modelling:** Calculate the equation for the frequency transmission through your bandpass filter, and compare the results to your measurement. Do the results agree? If not, can you explain what is different between your model and the circuit you built?

2. **Noisy signal:** We have been using a fairly large (5 V_{pp}) signal to measure our output with high signal to noise. What if our measurement has additional noise? Try injecting noise into your circuit, and see what the transmission curve looks like. How can we measure the transmission with higher signal-to-noise?

3. **Python classes:** So far, we have written a bunch of individual functions to control our instrument, which always take the instrument PyVisa resource as the first argument. If you are familiar with Python, you may recognize that a class object is a better way to organize this code. Go back to your function generator and oscilloscope python files and reformat the code into a single python class to control the instrument. Classes such as these are the standard method for writing instrument control software in Python.

3 Conclusion

We have now completed the full procedure you will follow in the lab to write Python code to automate measurements using Python. It took some time and effort to set up, but now you can perform frequency sweeps as many times as you want with little effort. This type of automation is enormously helpful in the lab, as it allows you to save time and mental energy each time you need to perform a measurement. In addition to our frequency sweep procedure, we can write other measurement procedures without having to rewrite the instrument control software for these instruments.

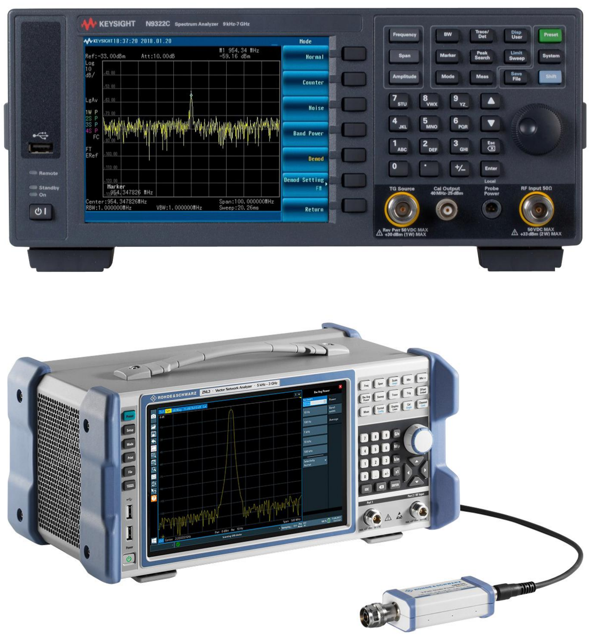

Figure 8: *Top*: Keysight spectrum analyzer. *Bottom*: Rhode & Schwarz vector network analyzer.

There are a few lab instruments you may encounter that perform similar functions to the code we have written. Examples of these instruments are shown in Fig. 8. The first instrument is the spectrum analyzer, which performs a similar function as the oscilloscope but with an FFT directly on the instrument to extract magnitude and phase. This instrument would allow us to skip the FFT step in Python, and some allow for peak amplitude extraction directly on the instrument. The second instrument is the Vector Network Analyzer (VNA), which has one port that outputs a signal and sweeps over frequency, and a second port at the other end of the circuit. Rather than performing an FFT and extracting the peak, the VNA directly compares the output signal to the input signal to extract the magnitude and phase relative to the input signal. Additionally, the VNA can measure the reflection back from the circuit into the output port, and switch the input/output to measure the backwards transmission and backwards reflection. If you have a VNA, you could complete this lab easily on the instrument without writing any of this code. However, both spectrum analyzers and VNAs are often more expensive than a function generator and oscilloscope, so you will not always have access to them. The code we have set up allows us to perform these measurements at a fraction of the cost.