Lab Manual: Data Science in Radio Astronomy II - Detecting Voyager 1

Introduction

Overview of the Voyager 1 Spacecraft

Voyager 1, launched in 1977, is the farthest human-made object from Earth, currently over 24 billion kilometers away. Despite its immense distance, it continues to transmit a weak carrier signal that can be detected by large radio telescopes like the Green Bank Telescope (GBT). Originally part of NASA's planetary exploration program, Voyager 1 provided groundbreaking data on Jupiter and Saturn before continuing into interstellar space. Today, its signal serves as a remarkable example of deep-space communication, demonstrating the challenges of detecting and analyzing extremely weak radio transmissions.

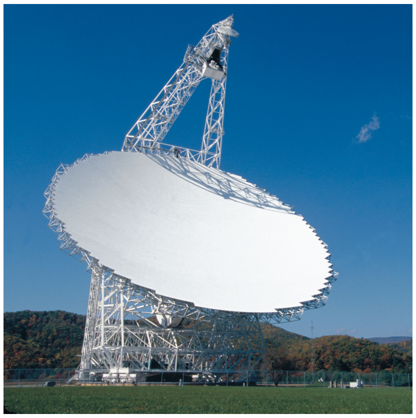


Figure 1: The 100-meter Green Bank Telescope (GBT), used to observe the signal from Voyager 1.

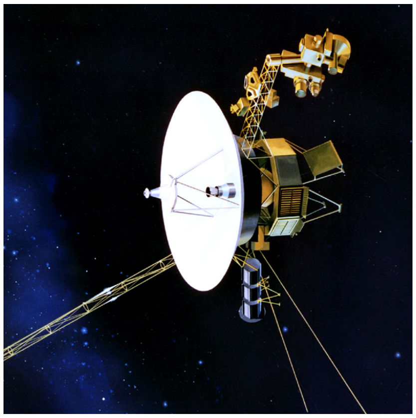


Figure 2: The Voyager 1 Spacecraft.

Why Study the Voyager 1 Signal?

Studying the Voyager 1 signal provides valuable insights into spacecraft tracking, radio astronomy, and signal processing. Detecting weak signals buried in noise is not only essential for monitoring space missions but also mirrors techniques used in SETI (Search for Extraterrestrial Intelligence). Spacecraft signals, like those from Voyager 1, are narrowband and can exhibit Doppler shifts due to relative motion, similar to how an extraterrestrial transmission might appear. By analyzing these signals, we refine methods for distinguishing artificial transmissions from background noise, a critical skill in modern radio astronomy.

How is the Signal Detected?

The signal recorded by the GBT primarily consists of a narrowband carrier transmitted at 8.4 GHz (X-band), used for communication with NASA's Deep Space Network (DSN). While Voyager 1 also transmits modulated telemetry data, this lab will focus on detecting and analyzing the carrier signal rather than fully demodulating its contents. The strength of this signal is incredibly low due to the vast distance, requiring sensitive instrumentation and careful data processing to extract useful information. Identifying the signal's spectral characteristics allows us to understand its structure, compare it against expected properties, and apply techniques for enhancing its detectability.

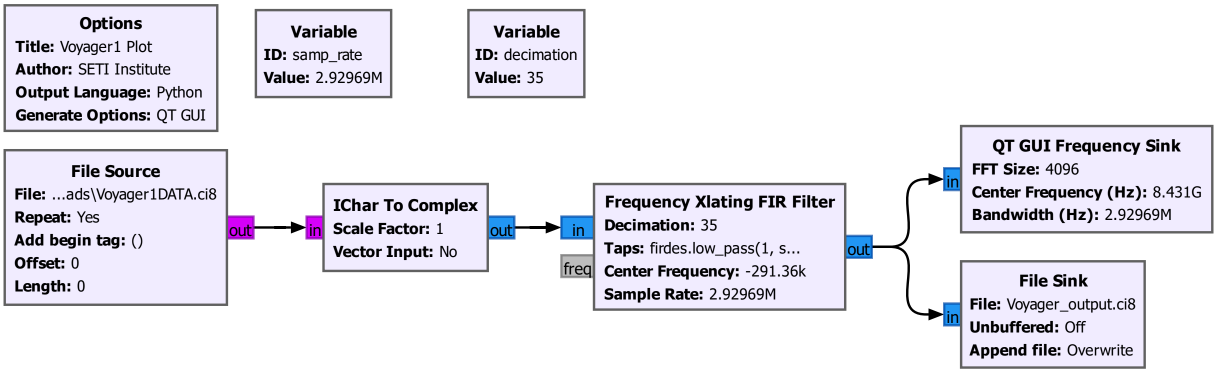
Overview of Lab Activities

In this lab, you will first use GNU Radio to visualize the recorded Voyager 1 signal, generating spectral plots and waterfall displays. After saving the processed data, you will transition to Python for detailed analysis. Automated peak detection techniques will help confirm the carrier frequency, and you will calculate the signal-to-noise ratio (SNR) to quantify its strength. Finally, you will explore radio frequency interference (RFI) mitigation by applying filtering techniques to improve signal clarity. Through this process, you will gain hands-on experience in data science topics used in radio signal analysis, reinforcing key concepts in digital signal processing, radio astronomy, and spacecraft communications.

Part 1: Signal Visualization in GNU Radio

Building the GNU Radio Companion (GRC) Flowgraph

1. Download the data file `Voyager1DATA.ci8`
2. Open GNU Radio Companion and create a new flowgraph.
3. Add the following blocks into your workspace and apply the modifications listed below. The color of the in/out ports reflect the data type being used by the block, left-click on the block and use your keyboard up/down arrows to change to data type, and connect the blocks to match Figure 3.



Options
Title: Voyager1 Plot
Author: SETI Institute
Output Language: Python
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 2.92969M

Variable
ID: decimation
Value: 35

File Source
File: ...ads\Voyager1DATA.ci8
Repeat: Yes
Add begin tag: ()
Offset: 0
Length: 0

IChar To Complex
Scale Factor: 1
Vector Input: No

Frequency Xlating FIR Filter
Decimation: 35
Taps: firdes.low_pass(1, s...
Center Frequency: -291.36k
Sample Rate: 2.92969M

QT GUI Frequency Sink
FFT Size: 4096
Center Frequency (Hz): 8.431G
Bandwidth (Hz): 2.92969M

File Sink
File: Voyager_output.ci8
Unbuffered: Off
Append file: Overwrite

Figure 3: GNU Radio Companion flowgraph to visualize the Voyager 1 signal.

- **Sample Rate** (Change the existing samp_rate variable to 2929690)
  - Sets the sampling frequency to match the observation parameters.
- **Variable** (ID: decimation, Value: 35)
  - A value that adjusts the sample rate when using the Frequency Translating FIR Filter block.
- **File Source** (File: your file path to `Voyager1DATA.ci8`, Output Type: byte)

  - Reads data from the file and passes it along for further processing.

- **IChar To Complex**
  - Converts the data type into the complex format.

- **Frequency Xlating FIR Filter** (Decimation: decimation, Taps: firdes.low_pass(1, samp_rate/decimation, 800, 50), Center Frequency: -291360.)
  - Frequency Translating Finite Impulse Response Filter - Pulls out the narrowband portion of the signal we're interested in.

- **QT GUI Frequency Sink** (FFT Size: 4096, Center Frequency: 8431e6, Y label: Relative Gain, Y units: dB, Config → Control Panel: Yes)
  - Performs a fast Fourier transform on the data and plots frequency vs. intensity.

- **File Sink** (File: Voyager_output.ci8)
  - Writes the input to a file for further processing in Python.

- Run the flowgraph using the play button on the top menu bar, you should see something similar to Figure 4.

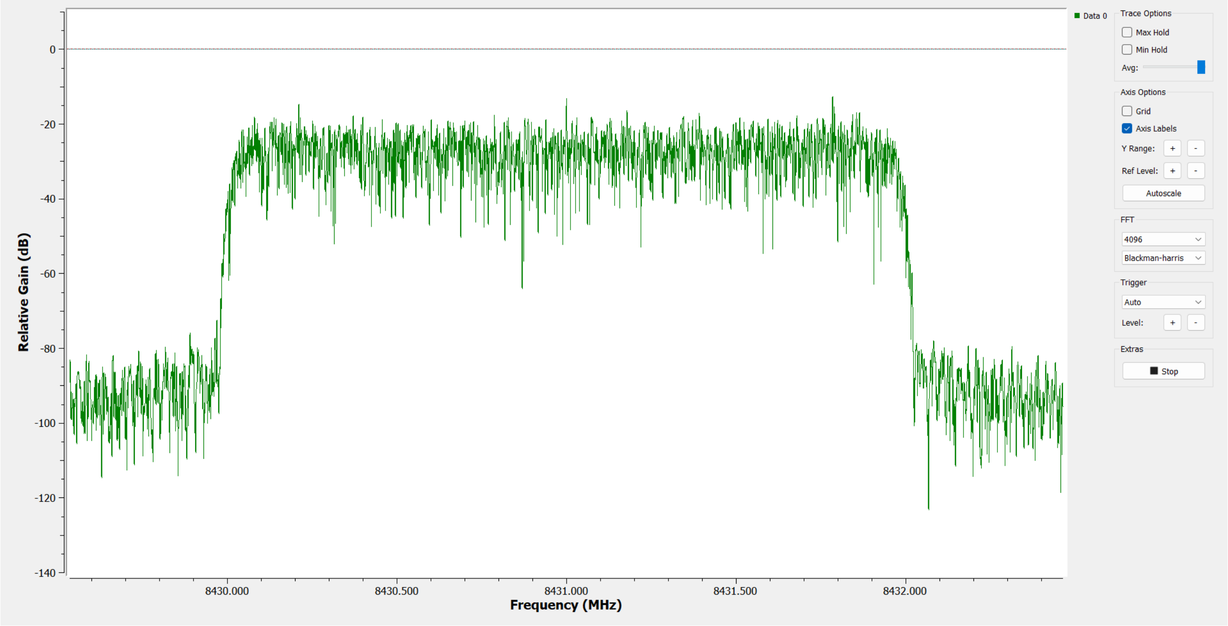


Figure 4: Initial visualization of the Voyager 1 Signal.

Interpreting and Enhancing the Voyager 1 Signal

When the frequency spectrum is first plotted, the signal will appear noisy, making it difficult to identify Voyager 1's transmission. The noise floor is relatively high due to the large distance and weak power of the spacecraft's transmission, causing the signal to be buried within random fluctuations. The shape of the noise is due to the Frequency Translating FIR filter, which attenuates the signal outside the region of interest containing the signal.

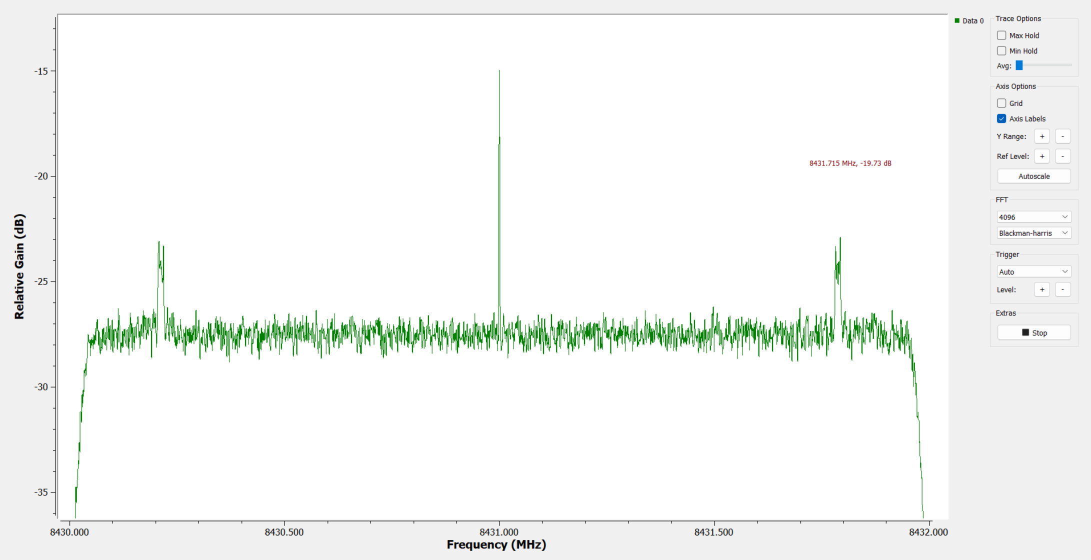
Questions:

1. What does the raw frequency spectrum look like?

2. Can you spot the central carrier and modulated sidebands among the noise?

3. What can be done to make the signal easier to see against the noise?

To improve visibility, **averaging** can be applied to reduce noise and highlight stable signal components. Noise fluctuates randomly over time, while the spacecraft signal remains consistent at a fixed frequency. By averaging multiple FFT frames together, random noise cancels out, while the stable signal becomes more pronounced. **In the control panel on the right, slowly adjust the averaging slider until the signal becomes visible.** Averaging smooths the background noise, making it easier to see the carrier signal, however excessive averaging can cause slow responsiveness and blur short-term variations in the data. **Isolate the region of interest (ROI) containing the central carrier and the modulated sidebands of the signal by left-clicking and dragging a box around the desired area.**



☐ Max Hold
☐ Min Hold
Avg:
☐ Grid
☑ Axis Labels
FFT: 4096, Blackman-harris
Trigger: Auto

Figure 5: Averaged region of interest containing the carrier wave and modulated sidebands of the Voyager 1 transmission.

Questions:

1. Why does the signal contain both a central carrier and sidebands?

2. How can we tell if the sidebands contain modulated data?

The main carrier wave appears at 8341 MHz which can be used to precisely track the Doppler velocity of the spacecraft. Modulated sidebands can be seen on either side, which contain the scientific and engineering data being transmitted.

Part 2: Signal Analysis in Python

In Part 1, you visualized the signal in GNU Radio and saved the data file. In this part, you will analyze that saved file using a Jupyter Notebook. Follow the step-by-step instructions below to perform spectral analysis, peak detection, and SNR computation on the saved signal.

Steps:

- **Open the Jupyter Notebook:**
  - Launch your Jupyter Notebook environment.
  - Open the provided notebook file: `DataScienceIILab.ipynb`
- **Install the Required Packages:**
  - Numpy
  - Matplotlib
  - SciPy
  - If these are not yet installed, open a terminal or Anaconda Prompt and run: `pip install numpy matplotlib scipy`
- **Step 3: Run Each Cell in Order**
  - **(a) Data Loading (Cells 1 & 2):**
    - Read the markdown cell titled *"Activity 1: Data Loading"* to understand the task.
    - Click on the code cell that loads the data file and press `Shift+Enter` to run it.
    - Check the output in the cell's output area to verify the correct number of samples is loaded.
  - **Frequency Axis Computation (Cells 3 & 4):**
    - Read the markdown cell titled *"Activity 2: Frequency Axis Computation"*.
    - Run the corresponding code cell by selecting it and pressing `Shift+Enter`.
    - Confirm that the printed frequency range (in MHz) is as expected and note the frequency span.
  - **Plot Entire Spectrum (Cells 5 & 6):**
    - Read the markdown instructions for *"Activity 3: Plot the Entire Power Spectrum"*.
    - Run the code cell that computes the FFT, averages the power spectrum, and plots the full spectrum.

    - Examine the resulting plot and document any significant features (e.g., dominant peaks, noise level) in a separate markdown cell.

- **Zoom into the Region of Interest (ROI) (Cells 7 & 8):**
    - Read the markdown cell titled *"Activity 4: Zoom into the Region of Interest (ROI)"*.
    - Run the code cell to create a frequency mask for the ROI, zoom in on the desired frequency range, and plot the zoomed spectrum.
    - In a separate markdown cell, note your observations about the features visible in the ROI.
- **Peak Detection on the ROI (Cells 9 & 10):**
    - Read the markdown cell titled *"Activity 5: Peak Detection"*.
    - Run the code cell that performs peak detection on the zoomed spectrum.
    - Verify that the detected peaks are clearly marked on the plot.
    - Adjust the peak detection parameters (if necessary) and document in a markdown cell how these parameters affect the detection.
- **SNR Computation (Cells 11 & 12):**
    - Read the markdown cell titled *"Activity 6: SNR Computation"*.
    - Run the code cell that defines the quiet regions, computes the noise floor, and calculates the SNR.
    - Note the printed SNR value and review the plot with the quiet regions highlighted.
    - In a separate markdown cell, discuss your observations about the SNR and consider alternative noise estimation methods (e.g., using the median).