Demodulating Meteor-M N2 Weather Satellite Data

Prava Dhulipalla, Minju Kang, Allison Basore

May 2019

1 Background

The Meteor-M N2 Russian Weather satellite was launched on July 8th, 2014 with the purpose of monitoring global weather. The Meteor-M N2 satellite shown in Figure 1 transmits digital LRPT protocol at a 137.1MHz frequency and a sample rate of 140000Hz. Using an RTL-SDR receiver, weather images can be transmitted by the satellite to a station on Earth while overhead. The satellite passes over Boston within a receiving radius approximately twice per day around 8 PM and 10 PM.

Figure 1: Meteor-M N2 Satellite

2 Motivation

Our primary motivation for this project was to learn more about QPSK demodulation and phase-locked looping, as those were ideas that we hadn't explicitly learned in our Analog and Digital Communications class.

3 Theory

As mentioned, we wanted to learn more about QPSK and Phase-Locked Looping. The following sections go into the details of the demodulating process.

3.1 LRPT

The Meteor M-2 transmits images back to earth in the LRPT, or Low Rate Picture Transmission, format. The LRPT system was created with the intent to be able to transmit and receive images from orbital weather satellites to uses with a VHF signal. When processed, the data-stream is processed using a convolutional error control code called Reed-Solomon, convolution encoded, processed with forward error correction, and then padded with unique synchronization words. This allows the system to validate and avoid unnecessary lost packages. When transmitting, the stream is transmitted as a 80 kB QPSK encoded signal, on a carrier, on the 137 MHZ band.

The LRPT has three digital channels. Each pixel of the resulting image representing approximately 0.62 mi of land. Another digital transmission system, APT (Automatic Picture Transmission), has a mere two image channels and an image resolution a whopping 4x smaller while also being high susceptible to errors.

For our project, we didn't have to do a super in-depth look into Low Rate Picture Transmission in order to help decode the data from the Meteor M-2 into an image - what we primarily focused on is the QPSK Modulation / Demodulation and the clock recovery (in terms of the phase-locked loop). We also learned about the error correction schema but chose not to implement given the timeline of this project.

3.2 QPSK

Quadrature Phase Shift Keying (QPSK) is a form of modulating data. Using two streams of data with two bits per symbol, it transmits 4 bits at once. One of four possible carrier phase shifts 0, 90, 180, or 270 degrees is used for each bit. The two data streams correspond to the first and second bit of data. For example, with the sequence 01 00 10 10 would mean the first stream transmitted 0,0,1,1 and the other stream transmitted 1,0,0,0. The two streams are the cosine sine parts of the signal and can be separated by imaginary and real parts.

In order to demodulate the bits, one stream is multiplied by a negative sine and the other by a cosine put there a lower pass filter, and potentially amplified.

Figure 2 shows a diagram of the demodulation.

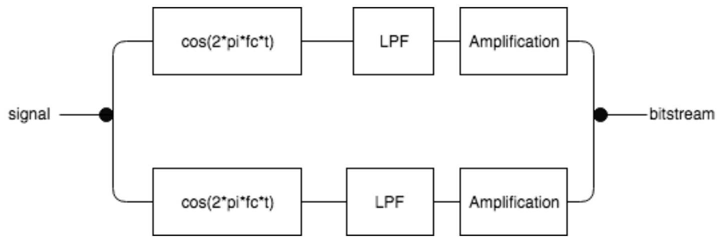

Figure 2: QPSK Diagram

3.3 Costas Loop

When you transmit QPSK model over the aerial, you would find yourself receiving deviated signal due to the instability of the transmitter or receiver as well as Doppler effect. The deviated signal will have phase and frequency offset, resulting the received signal such as figure 3 (Image credit to jmfriedt).

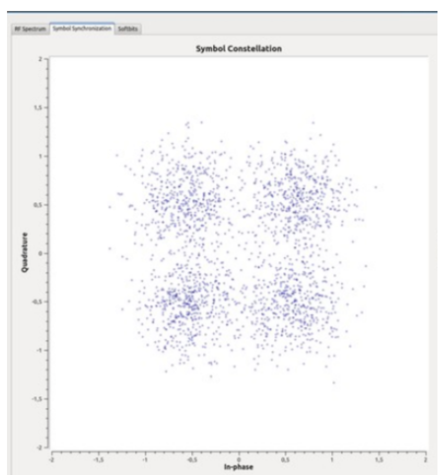

Figure 3: Example Raw Signal Received Through an Antenna

Since the raw signal is incorrect a Phase Lock Loop (PLL) is used for estimating the original transmitted signal. A Costas loop is a special kind of PLL mechanism and it enables the signal to have the frequency and phase offset corrected.

$$V_{RF} = I(t)sin(\omega t + \theta) + Q(t)cos(\omega t + \theta) \tag{1}$$

If the signal is QPSK modulated as equation 1 where *I(t)* and *Q(t)* can be either +1 or -1, the filtered signals will be equation 2 and equation 3.

$$Z_I(t) = I(t)cos(\theta - \theta') - Q(t)sin(\theta - \theta') \tag{2}$$

$$Z_Q(t) = I(t)sin(\theta - \theta') - Q(t)cos(\theta - \theta') \tag{3}$$

If we multiply the equation 2 and 3 with *I(t)* and *Q(t)*, we will get the following equation.

$$\begin{gathered} \text{e (t)} = \text{I}^2(t)sin(\theta - \theta') - Q(t)I(t)cos(\theta - \theta') \\ \text{- I(t) Q(t) cos}(\theta - \theta') + Q^2(t)sin(\theta - \theta') \\ = 2\ \text{sin}(\theta - \theta') \\ \approx 2(\theta - \theta') \end{gathered}$$

(4)

Then, once you calculate the phase offset, you can reproduce the the signal that's originally transmitted.

For our own implementation, we used different implementation for the demodulation. Our received signal can be represented as equation 5 where h represents the real numbers, f_δ represents the frequency offset and n[k] represents the noise.

$$y[k] = hx[k]e^{jf_\delta k+\theta}n[k] \tag{5}$$

The figure 4 plots the Fourier transform of the signal. From the figure, we could see that the clearest carrier frequency signal when the signal is quadrupled.

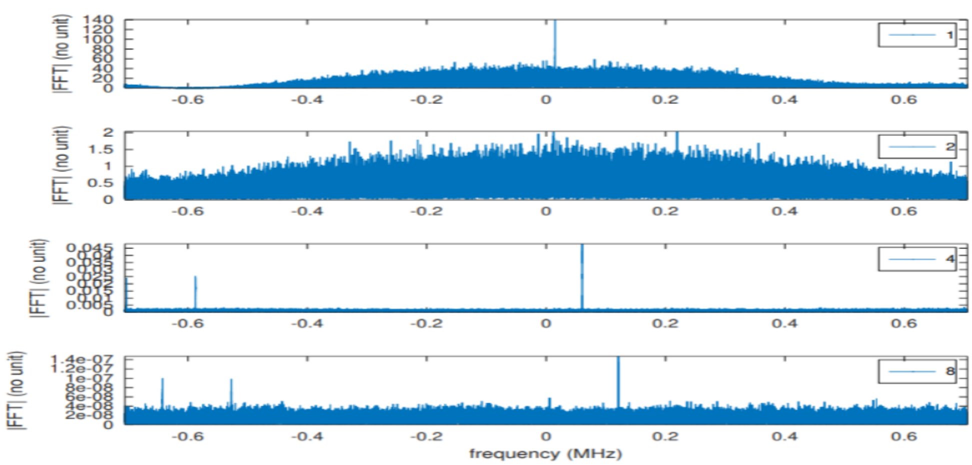

Figure 4: FFT plots of powered signal

Therefore, we quadrupled the signal and extracted the following signal.

$$(x[k])^4 = -he^{j(4f_\delta+4\theta)} \tag{6}$$

By looking at the fft of quadrupled signal, we could estimate the f_δ and θ. From such, we attempted to regenerate the original signal.

4 Methodology

Using references of other similar projects we determined the hardware and software components of our project. For the hardware, we needed an antenna that broadcasts over the range of 139.9KHz. This connects through a standard coaxial cable to a USB RTL-SDR which plugs directly into our laptops. For software, we used MATLAB to load and process the data .wav files with QPSK and PLL code. Finally we perform error correction for mis-transmitted bits and converted the bits to an image using online software. The sections below give more details about each part of our methodology.

4.1 Data Collection

For data collection we attempted to use two antenna. The first was the SUPERSONIC Motorized Amplified TV Antenna with a reception rage of 120 Miles, frequency range of 47-860 MHz and antenna gain range of 20-28dB. The second antenna we used was the Mohu Leaf Fifty Amplified Indoor HDTV Antenna with a 60-mile range, frequency range of 49-952MHz and a 15 dB gain. Figure 5 and 6.

Figure 5: SUPERSONIC Antenna

We checked whenever the satellite passes over our location via *Orbitron*. We connected the antenna to the power box and collected the data with proper amplification values. Figure 8 also shows a map image of satellite's path flying close to our location. The Meteor M2 satellite data has a sample Rate of 140,000 and each packet is 1024 byte long.

Unfortunately, we were not able to collect any data with these antennas. After many nights of attempting to collect data and seeing no spikes, we concluded that both of our antennas were not strong enough and required better detection. The first antenna, while it had a wider range, was highly directional and was could not consistently find the satellite. The second antenna was not powerful enough to detect the full signal. Alternatives would be to build an antenna or to buy an antenna with high amplification and further range. Additionally, we

Figure 6: Mohu Antenna

collected data only when Meteor-M2 was overhead. Another approach would be to mount the antenna and be taking data constantly. While we were not able to collect data ourselves, we were able to access an alternative raw data-set that we could use to process with our code.

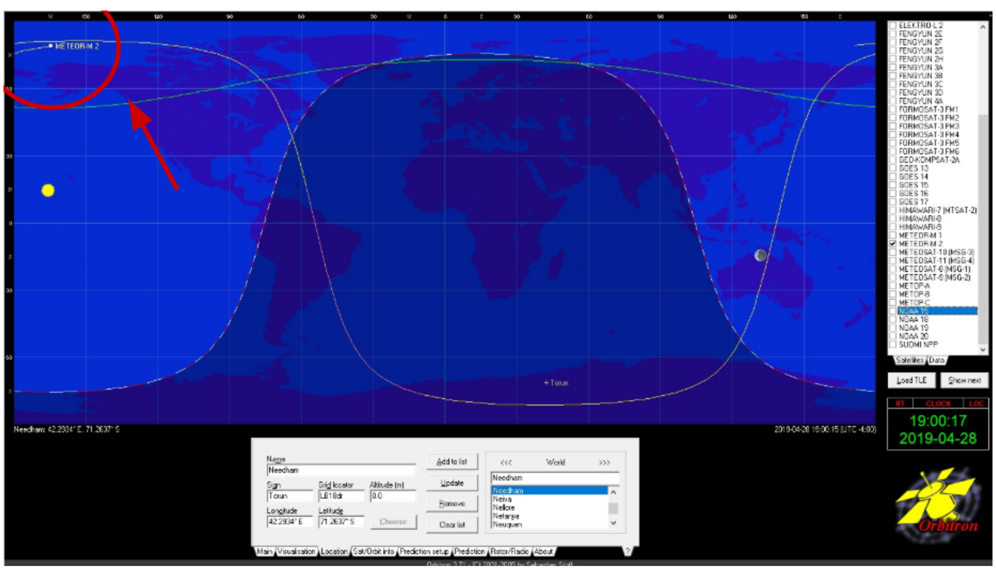

Figure 7: Screenshot of Orbitron

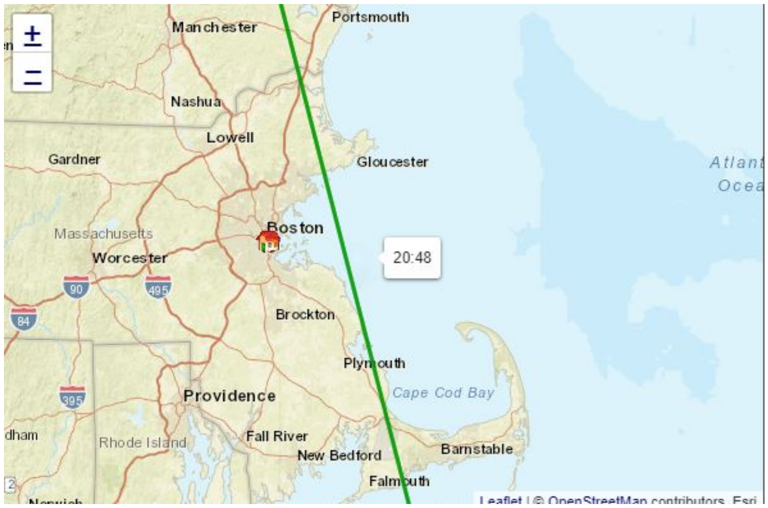

Figure 8: Map view of Meteor M2 path

4.2 Processing

Our Github Repo has the latest versions of our QPSK and PLL code.

After re-sampling to match the symbol rate we were able to see the QPSK constellation. Figure 9 shows this constellation for a portion of our image data.

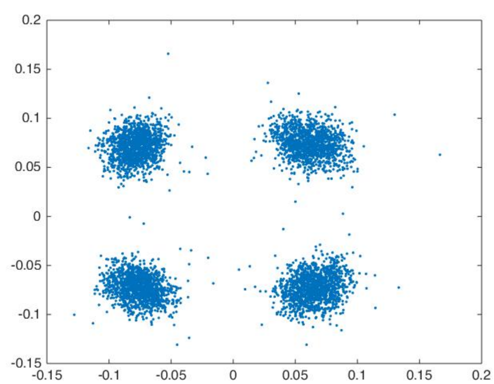

Figure 9: QPSK Constellation

Once this data was put through a PLL loop for phase correction, we were able to convert it to 8 bit signed soft QPSK. An LRPT analyzer then converts the bytes to an image.

5 Results and Takeaways

Using an LRPTRx application, we were able to test our data for images. This LRPTOffline decoder can be found as part of this suite of LRPT tools: https://www.dropbox.com/s/qq1fjyitpa3j14o/software.zip

This application runs a control loop to correct for phase shift and performs error correction. Using this program to create the 8 bit signed soft QPSK we were able to obtain the satellite images. Figure 10 and 11 shows the images collected with this method.

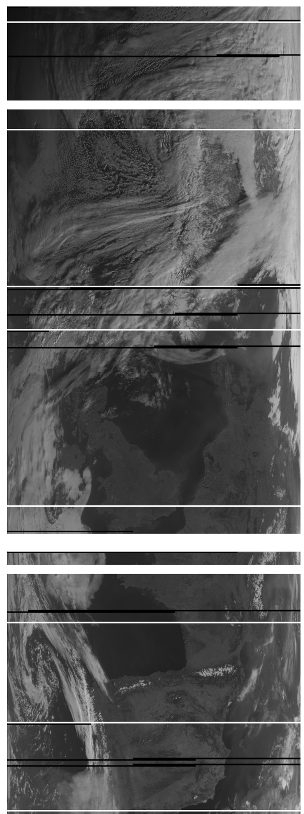

Figure 10: Result Image 1

After the time spent collecting the data, we were not able to complete the phase shifting correction code and bit creation phase with our own code. While we were not able to get the results we wanted, we feel that we learned many good takeaways from this project. We are now familiar with the complexities that come with handling real data. We learned about the specifics of collecting data with an antenna and used SDR for collection. We also observed an learned several techniques for phase shift correction along with QPSK processing.

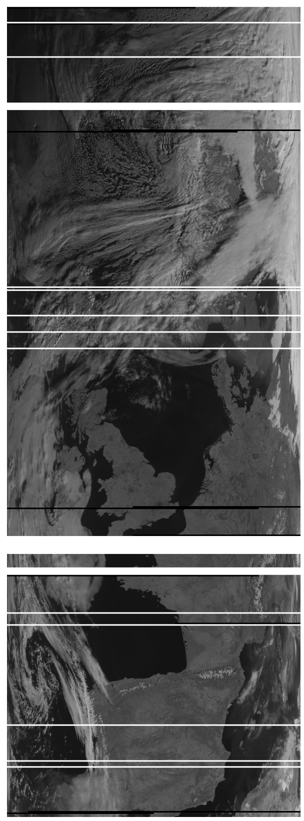

Figure 11: Result Image 1