

GNU Radio Conference 2019

Demonstration of GNU Radio Compatibility with a NASA Space Communications Network Modem

September 2019

Dave Miller
Dave.Todd.Miller@gmail.com

BACKGROUND

- ❑ **For GNU Radio Conference 2019, the Conference leadership encouraged the following activities and papers:**
 - GNU Radio Demonstrations rather than just GNU Radio simulations
 - Conference presentations emphasizing GNU Radio tutorials
- ❑ **Presenter Background:**
 - Currently employed as contractor at NASA Goddard Space Flight Center
 - However, note that all information in this presentation and associated paper is based on perspective and opinions of presenter, not NASA
 - Dave.Todd.Miller@gmail.com (Personal)
 - dmille60@peraton.com (Work)
 - About 1 year ago, began looking for cost-effective Digital Signal Processing (DSP) communications system simulation tool alternatives to Matlab/Simulink
 - Concluded that GNU Radio is good cost-effective alternative as communications simulation DSP tool and can also be used to develop an actual cost-effective Software Defined Radio (SDR) modem at data rates that NASA typically uses at S-Band
- ❑ **Therefore, developed GNU Radio SDR Receiver and then performed testing to demonstrate that GNU Radio SDR is compatible with an actual vendor modem (RT Logic) that has been incorporated into a NASA ground station**

Purpose and Scope of Presentation

□ **Purpose:**

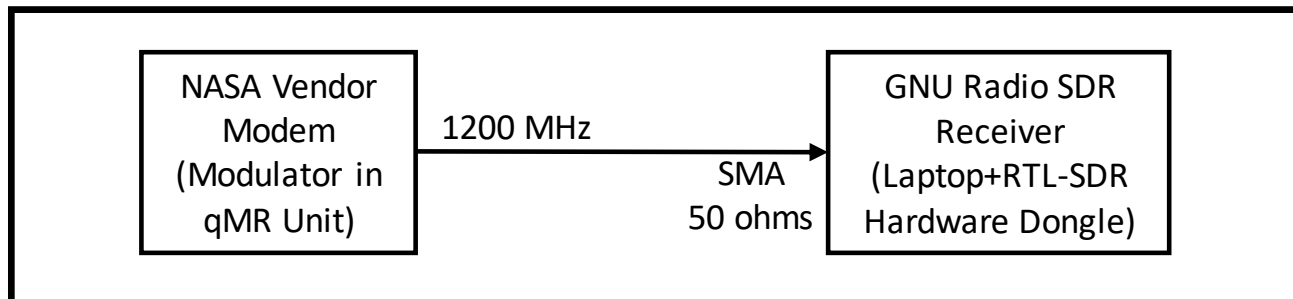
→ From tutorial perspective, this presentation will describe how the GNU Radio SDR receiver was developed and then tested with a typical NASA vendor modem

□ **Scope of Presentation:**

→ This presentation will emphasize describing modem development and testing activity from tutorial perspective in order to help others new to GNU Radio quickly implement a practical, cost-effective, and functional GNU Radio SDR Receiver with available GNU Radio Library Blocks

Scope of Demonstration Test Activity

- As Figure below depicts, this compatibility demonstration test activity used “back-to-back” test configuration:
 - 50 ohm coaxial cable between GNU Radio SDR receiver and NASA vendor modem modulator
 - 1.2 GHz Intermediate Frequency (IF)
 - Quantum Mission Receiver (qMR) unit manufactured by company RT Logic
 - Conducted tests with BPSK and QPSK
 - Typical NASA waveforms

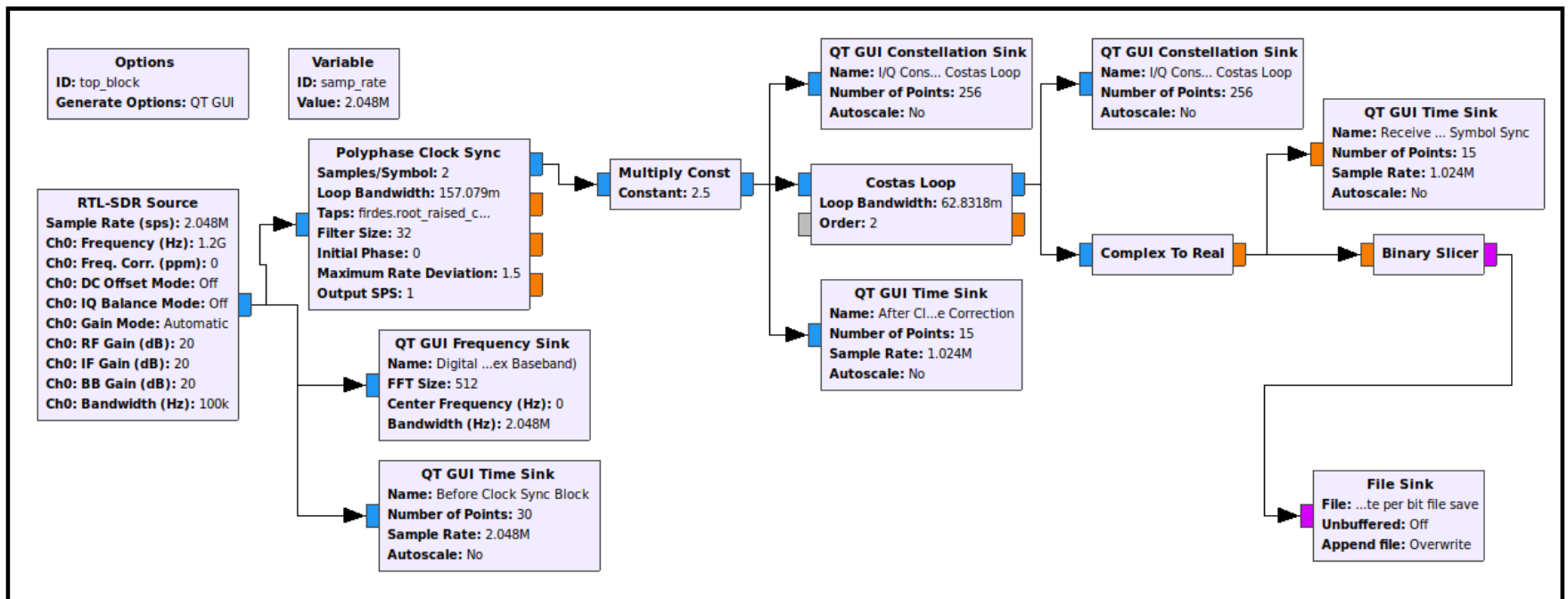


GNU Radio SDR Receiver Design

- **GNU Radio Receiver consisted of following parts emphasizing cost-effectiveness while obtaining performance up to 2.048 Mbps:**
 - Dell Laptop
 - Free open source Linux/Ubuntu operating system available on-line
 - Free open source GNU Radio software (version 3.7.11)
 - Inexpensive \$28.00 Commercial Off-The-Shelf (COTS) RTL-SDR hardware dongle (RTL2832U)
 - Universal Serial Bus (USB) interface on one side for connection to Dell laptop
 - On its other side, 50 ohm SubMiniature version A (SMA) interface for connection to IF signal
 - Refer to Section 1.6 of Stewart, R., 2017 for detailed description of RTL-SDR hardware dongle functions and design
 - Frequency Tuning and downconversion to digital IF (complex I/Q Baseband samples)
 - Bought via Amazon website

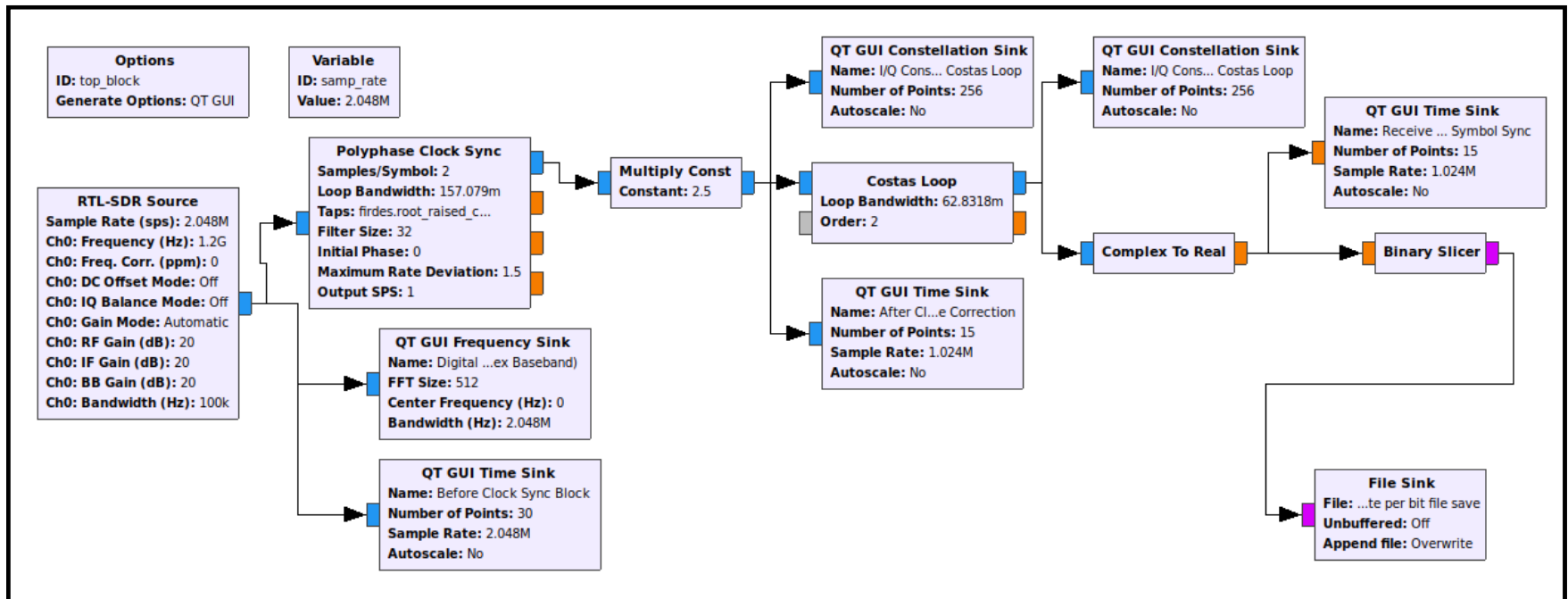
GNU Radio SDR Receiver Design (BPSK Test Case)

- ❑ Figure below depicts GNU Radio Companion Flowgraph Graphical User Interface (GUI) for BPSK test demonstration
- ❑ RTL-SDR Source block controls RTL-SDR hardware dongle
 - ➔ Tunes to IF frequency, downconverts IF signal to Digital IF (complex I/Q baseband), and sends digital IF complex sample stream to Dell laptop via USB interface
- ❑ Costas Loop block provides carrier synchronization
- ❑ Polyphase Clock Sync block provides symbol synchronization
 - ➔ Implementation based on algorithm described in Harris F., 2001



GNU Radio SDR Receiver Design (BPSK) (Real-Time Monitoring)

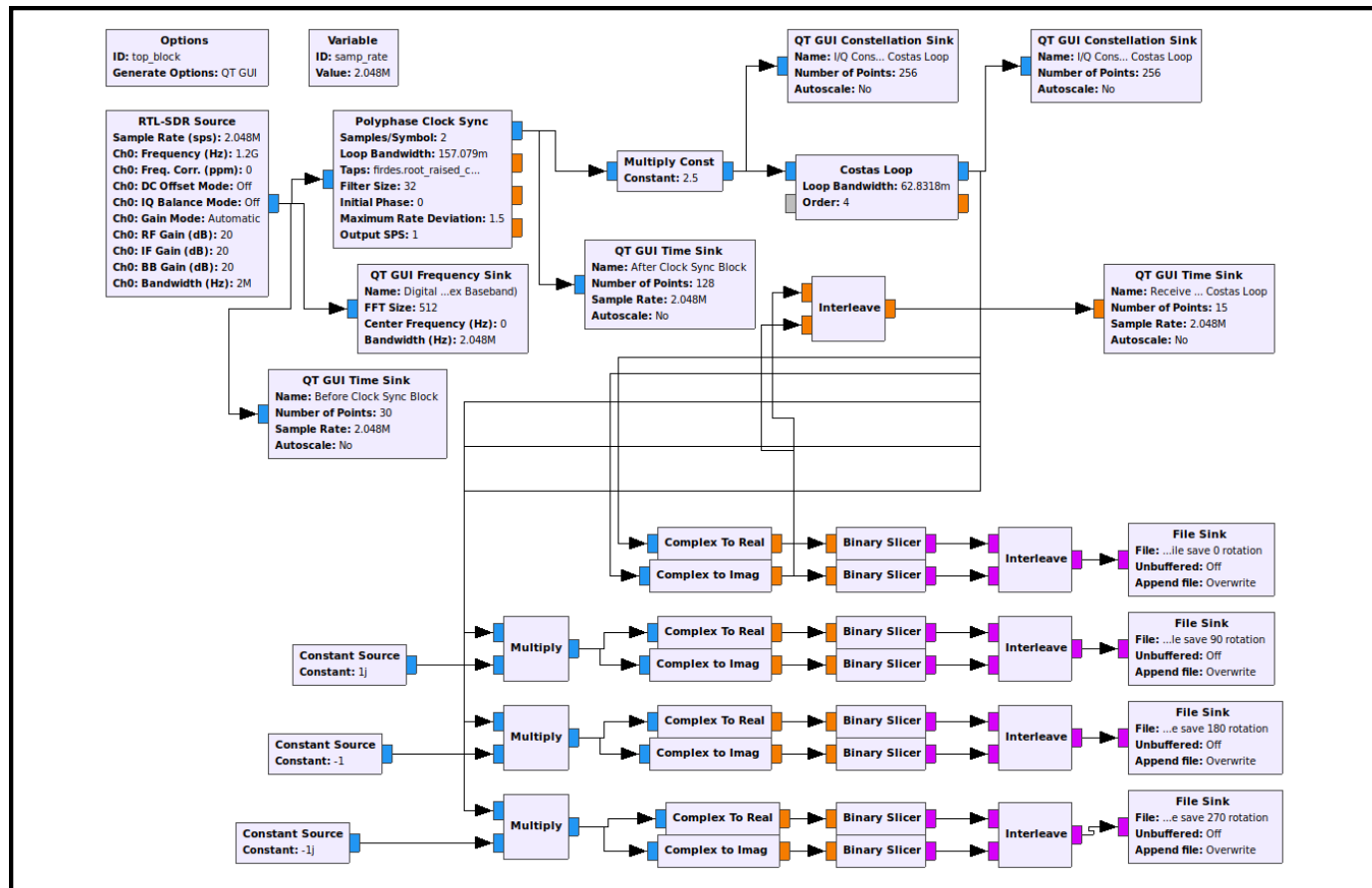
- ❑ **Blocks to monitor performance in real-time via displays and blocks for storage**
 - Time Sink blocks provide oscilloscope functions to monitor bit stream
 - Frequency Sink block provides Spectrum Analyzer function
 - I/Q Constellation Sink blocks to monitor carrier synchronization
 - File Sink block passes recovered bit stream to file for storage and post-test analysis



GNU Radio SDR Receiver Design (QPSK Test Case)

❑ QPSK flowgraph similar to BPSK, except:

- Complex multiply blocks added to provide convenient I/Q constellation rotation function to resolve carrier lock phase ambiguity (one of saved files will have correct transmitted bit pattern)
- Costas Loop Block order = 4 rather than 2



Main Design Challenges During Development from Tutorial Perspective

- ❑ During development, GNU Radio documentation and discussions boards provided a lot of useful information about GNU Radio blocks
- ❑ However following two charts provide details on following items that were most challenging to presenter to hopefully help new users:
 - Taps Parameter of Polyphase Clock Sync Block
 - Adding RTL-SDR Source Block for dongle control to GNU Radio Library

“Taps” Parameter of Polyphase Clock Sync Block

□ Details on setting for “taps” parameter in GUI for Polyphase Clock Sync Block:

- `firdes.root_raised_cosine(32, 32*Samples per symbol, 1.0, RRC rolloff, 2 * L * Number of filters in polyphase filter bank)`
 - For SDR testing during this activity:
 - Samples per symbol = 2
 - Root Raised Cosine (RRC) Roll-Off Factor = 0.5
 - L = length of “prototype” filter response in symbols = 11 for this demo activity
 - Number of Filters in Polyphase Filter Bank = 32 for this demo activity
 - Therefore, final setting for taps parameter in block:
 - `firdes.root_raised_cosine(32,64,1.0,0.5,704)`
- C++ code for blocks like symbol synchronizer and costas loop is on <https://github.com/gnuradio>
 - Code provides more insight into details of blocks

Addition of RTL-SDR Source Block

- ❑ **RTL-SDR Source Block was not available in original GNU Radio installation that was used**
- ❑ **Although better method may have been available, following command found on discussion boards was used in Ubuntu Terminal to install both GNU Radio (version 3.7.11) and RTL-SDR Dongle Source Block Simultaneously:**
 - `sudo apt install gr-osmosdr`

Development Process in Steps

- ❑ **Development of GNU Radio SDR occurred in stages at home before last stage of testing with vendor modem at another facility:**
 - First, Simulation: Developed Simulation Modulator with GNU Radio Blocks
 - Second, Procured RTL-SDR Dongle that included an Antenna: Tested with FM radio stations
 - Third, Procured an inexpensive new Signal Generator (\approx \$90) on Ebay: Signal Generator included BPSK function at 50 MHz IF with continuous 10101010 data pattern capability
 - FeelTech FY6600-60M
 - Fourth, Final Demonstration Testing: Conducted final BPSK and QPSK testing with actual vendor modem at remote facility

Test Approach

- **The following specific demonstration test cases were conducted:**
 - Test Case #1: BPSK at 1.024 Mbps
 - Test Case #2: QPSK at 2.048 Mbps
 - NASA vendor modem transmitted short repeating bit pattern of 15 bits in length, specifically, Pseudo Random Bit Stream (PRBS) 4 pattern

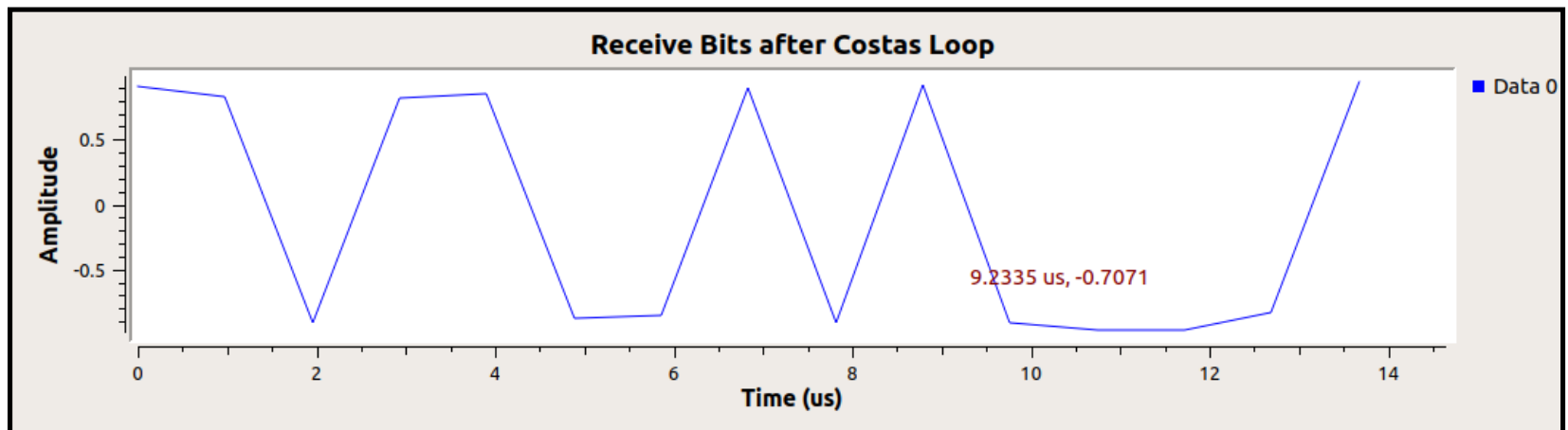
GNU Radio Companion Block	BPSK (1.024 Mbps) Test Case #1	QPSK (2.048 Mbps) Test Case #2
RTL-SDR Dongle Source Block:		
Center Frequency	1.2 GHz	1.2 GHz
Sample Rate	2.048 Msps	2.048 Msps
Costas Loop Block:		
Order	2	4
Symbol Synchronizer Block:		
Input samples per symbol	2	2
Output samples per symbol	1	1
RRC Roll-Off Factor	0.5	0.5

Demonstration Test Results

- ❑ **The following performance occurred during both test cases (BPSK and QPSK):**
 - GNU Radio SDR successfully recovered transmitted bit stream with correct repeating bit pattern (PRBS4)
 - GNU Radio displayed expected frequency spectrums for BPSK and QPSK at tested data rates
 - GNU Radio SDR successfully achieved Symbol Synchronizer Lock
 - GNU Radio SDR successfully achieved Carrier Synchronization Lock
- ❑ **Following charts provide more information and figures taken in real-time to show how above conclusions were made within scope of testing**

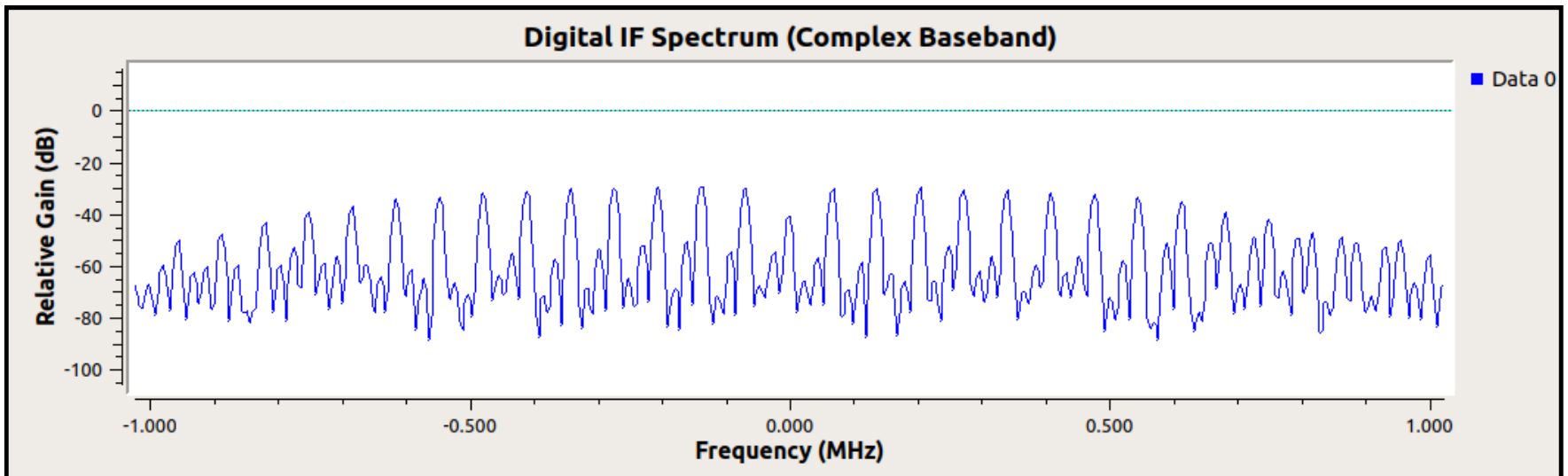
Demonstration Test Results (Bit Pattern Recovery)

- **PRBS 4 BER performance was analyzed via Time Sink Blocks in real-time and collected bit stream data in saved files as post-test activity**
 - Snapshot of BPSK Time Sink Block display during testing is shown below
 - Bits in snapshot are actually flipped version of PRBS4 pattern because of phase onto which Costas Loop locked during test case
 - This initial phase of testing was conducted without adding noise, therefore, as one would expect, perfect Bit Error Rate (BER) performance occurred



Demonstration Test Results (Frequency Spectrum)

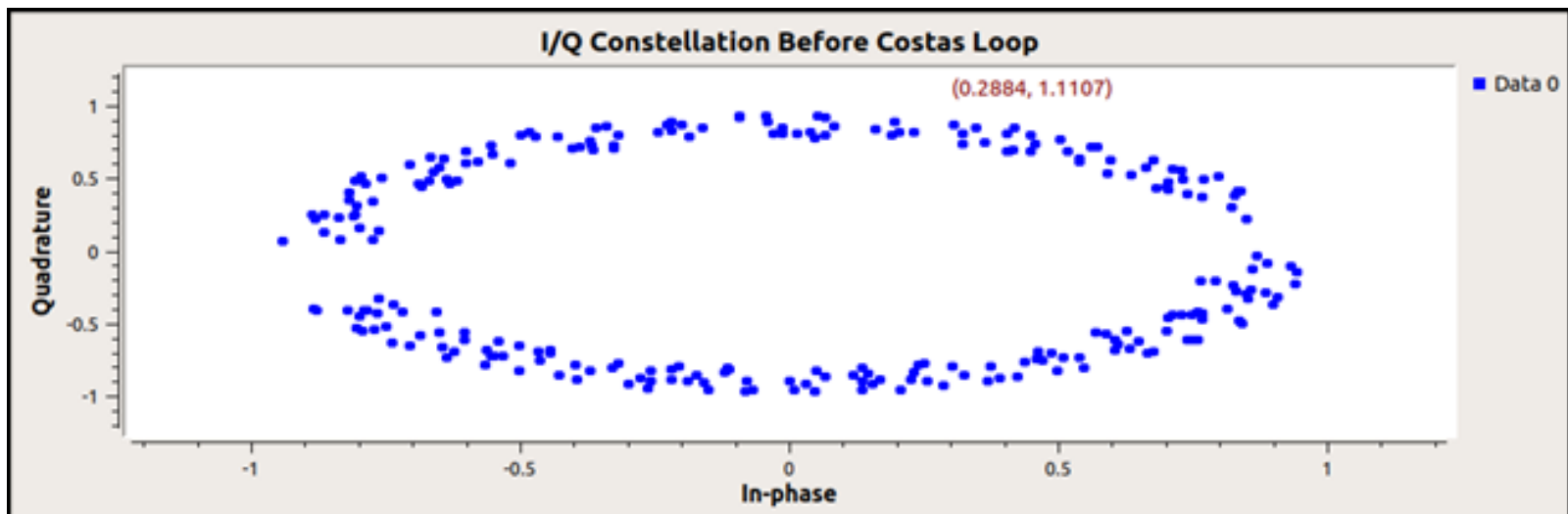
- Figure below depicts a snapshot of GNU Radio Frequency Sink block display during BPSK test case:
 - 1.024 Mbps data rate
 - Repeating 15 bit pattern



Demonstration Test Results

(I/Q Constellation “Before” Costas Loop Block)

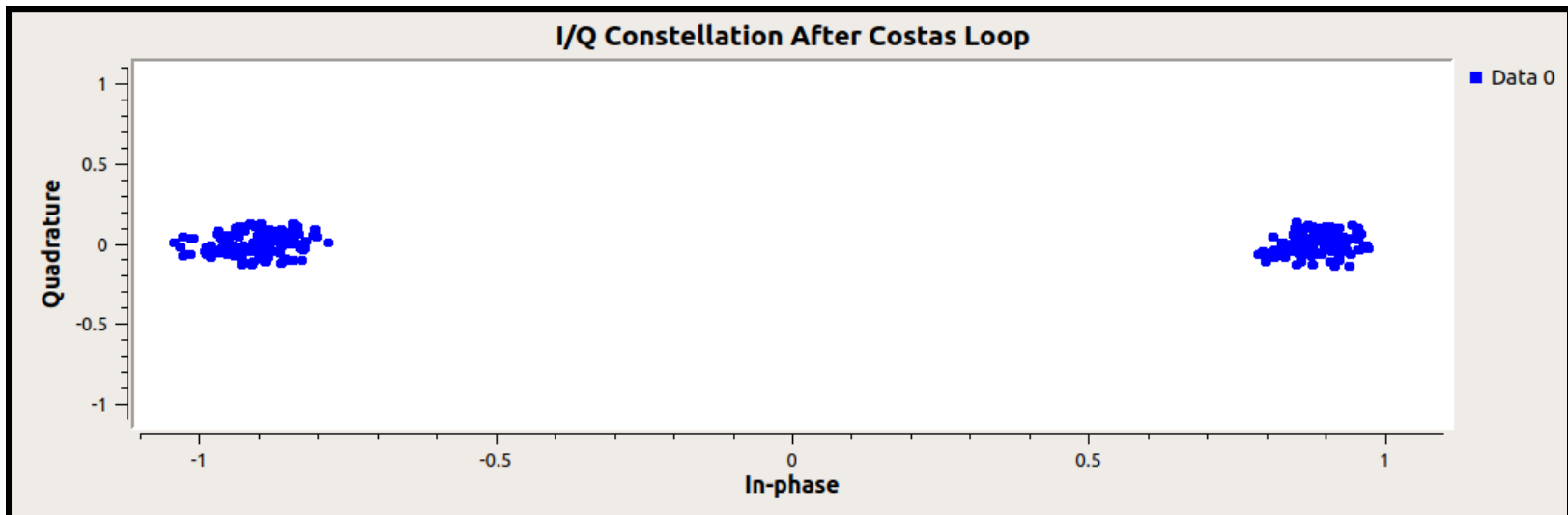
- Figure below depicts snapshot of display of I/Q Constellation Sink block located before Costas Loop block during Test Case #1 (BPSK)
 - Specifically, Figure depicts rotating I/Q Constellation that occurred before carrier synchronization
 - Before Costas Loop block removed residual frequency offset between Vendor Modem and RTL-SDR hardware dongle



Demonstration Test Results

(I/Q Constellation “After” Costas Loop Block)

- Figure below depicts snapshot of display of I/Q Constellation Sink block located after Costas Loop during Test Case #1 (BPSK)
 - Specifically, figure depicts I/Q Constellation without rotating I/Q Constellation that occurred after carrier synchronization
 - After Costas Loop block removed residual frequency offset between Vendor Modem and RTL-SDR hardware dongle



Demonstration Test Results (Residual Frequency Offset Removal)

- From more detailed tutorial perspective, equation (1) below depicts complex baseband signal with residual frequency rotation that RTL-SDR hardware dongle provided to GNU Radio and then Costas Loop Block Removed:

$$z_{IN}(t) = e^{j\theta t} \cdot e^{j[(2\pi f_{\Delta})t + \theta_{\Delta}]} \quad (1)$$

- For BPSK, $\theta = 0$ degrees or 180 degrees
- For QPSK, $\theta = 45$ degrees, 135 degrees, 225 degrees, or 315 degrees
- $z_{IN}(t)$ = Complex signal from RTL-SDR hardware dongle
- f_{Δ} = Residual frequency difference between IF signal from NASA vendor modem and RTL-SDR dongle hardware center frequency even when both were set to 1.2 GHz center frequency
- θ_{Δ} = Fixed residual phase difference between IF signal from NASA vendor modem and RTL-SDR dongle hardware center frequency at $t=0$ even when both were set to 1.2 GHz center frequency
- t = time
- Equation (2) depicts complex baseband signal with residual frequency and phase now removed (Costas Loop output with carrier synchronization lock)

$$z_{OUT}(t) = e^{j\theta t} \quad (2)$$

- $z_{OUT}(t)$ = Complex signal at output of Costas Loop after carrier synchronization lock

Demonstration Test Conclusions

- **Within scope of this initial demonstration testing phase, GNU Radio is compatible with some NASA modems and NASA waveforms like BPSK and QPSK at NASA S-band data rates**
 - However, additional testing phases and expanded functionality will be needed in order to fully characterize GNU Radio SDR performance with all NASA waveforms and modems that NASA employs for S-band communications services
 - For example, NASA Phase Modulation (PM) with Subcarrier waveform and also NASA coding should be part of forward GNU Radio SDR work

Possible Uses for GNU Radio in a Ground Station & Forward Work

- **Based on this initial GNU Radio SDR design, development, and testing activity, operations and engineering personnel of ground station could use GNU Radio SDR as follows:**
 - Employ immediately as inexpensive, compact, light, and mobile test tool with receiver, spectrum analyzer, and oscilloscope capabilities for conducting troubleshooting activities
 - Employ immediately as an inexpensive modem for engineering and operator team digital communications and communications theory training and education, including end-to-end link simulations
 - Recently provided GNU Radio demos to NASA space network communications engineers and their supporting contractor engineers
 - Employ immediately as inexpensive and easy to reconfigure modem in test bed, (including end-to-end testing) to explore new future network waveforms in cost-effective manner
 - Employ immediately as operational inexpensive digital IF recorder and baseband data recorder
 - Forward Work: After additional functions like PM demodulation, modulator, coding, equalizer, and Doppler coarse frequency offset tuning are added, GNU Radio SDR could be used by NASA or another organization as inexpensive and easy to configure spare modem for some operations to augment more expensive standard vendor modems