A four-channel voltage recording system for Radio Frequency Interference mitigation research

D. Anish Roshi[1,2], E. Armas[1,2], C. Wescott[1,2], W. Dellinger[1,2], N. Patel[1,2]

(1) Florida Space Institute, Orlando, USA, anish.damodaran@ucf.edu,

(2) University of Central Florida, Orlando, USA

CARSE internal report, October 23, 2024

Abstract

A multi-channel voltage recording system is essential to developing algorithms for mitigating radio frequency interference (RFI) in radio astronomy. One of the system's channels will be connected to the radio telescope, and other channels will be used to acquire data from reference antennas. We decided to use reconfigurable hardware to realize the system. We have created a system that includes radio frequency (RF) signal conditioning, analog-to-digital conversion (ADC), high-speed data transfer, and a data storage unit for storing time-domain voltage samples from four channels. This system is built using custom RF hardware, FPGA-based digital signal processing (DSP), and a server-based software solution capable of storing large volumes of data from the FPGA stream. This report outlines the details of the four-channel voltage recording system.

1.1 Hardware system

We needed a multi-channel voltage recording system to aid in our RFI mitigation work. To accomplish this, we decided to develop a system based on a Field-Programmable Gate Array (FPGA). Fig. 1 illustrates the block diagram of the complete system, which consists of three main components: 1) custom radio frequency (RF) hardware for signal conditioning, 2) an FPGA with integrated analog-to-digital converters (ADCs) that facilitates high-speed data transfer via 100 GbE, and 3) a server designed to store large volumes of data from the FPGA stream.

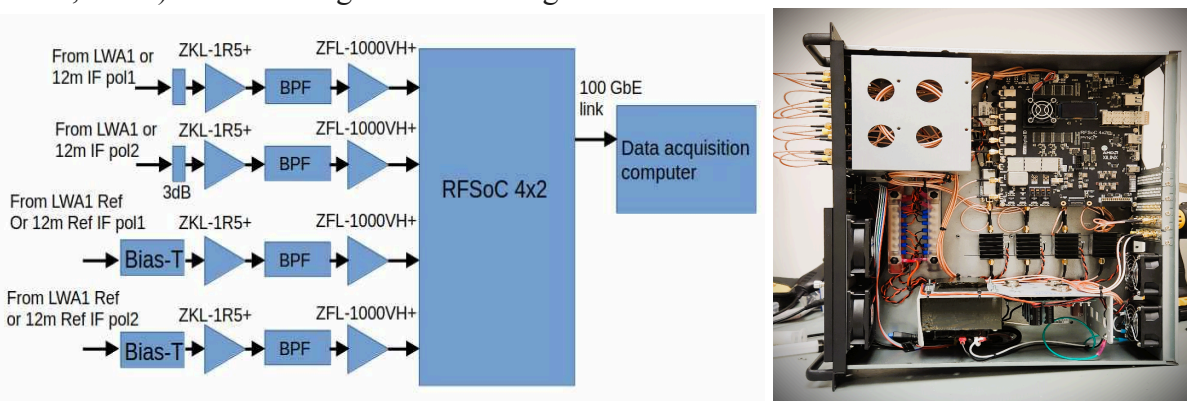

Figure 1 (left) illustrates the block diagram of the 4-channel voltage recording system. Two of the four channels include bias-T modules to power the reference antennas' low-noise amplifiers (LNAs). On the right, the custom RF hardware features an RFSoC 4x2 FPGA board, a power supply unit, and bias-T modules, all integrated into a single chassis made from a standard 19” server rack-mountable frame. The server itself is housed in a similar 19” chassis. Both units are housed together in a mobile equipment rack.

The RF signal conditioning unit is designed to achieve a high dynamic range (see [1]). Fig. 2 illustrates the circuit schematic of the unit. It consists of two off-the-shelf Minicircuits amplifiers per channel, which amplify the signal to the levels required for the ADCs. These amplifiers have 1 dB compression points greater than +15 dBm. For two of the four channels, Bias-Ts are included to power the LNAs of the reference antenna. We re-engineered the Bias-Ts to operate within a frequency range of 20 MHz to over 3 GHz. Custom 30 to 40 MHz bandpass filters are

currently integrated into the system to collect data from the DLITE [2] facility. Fig. 3 shows the response of the bandpass filter.

We use a RealDigital RFSOC-4x2 development board with an AMD Xilinx XCZU48DR System on Chip (SoC). This board provides four 14-bit RF ADCs accessible via SMA connectors. The FPGA firmware (see Section 1.2) was developed to buffer data from all four onboard ADCs, package it together, and transmit it to the server using a 100 Gbps Medium Access Control (MAC) and QSFP fiber link. The data is then recorded onto a set of four solid-state drives (SSDs) connected to a PCIe expansion card, providing a total memory capacity of 4TB across the SSDs.

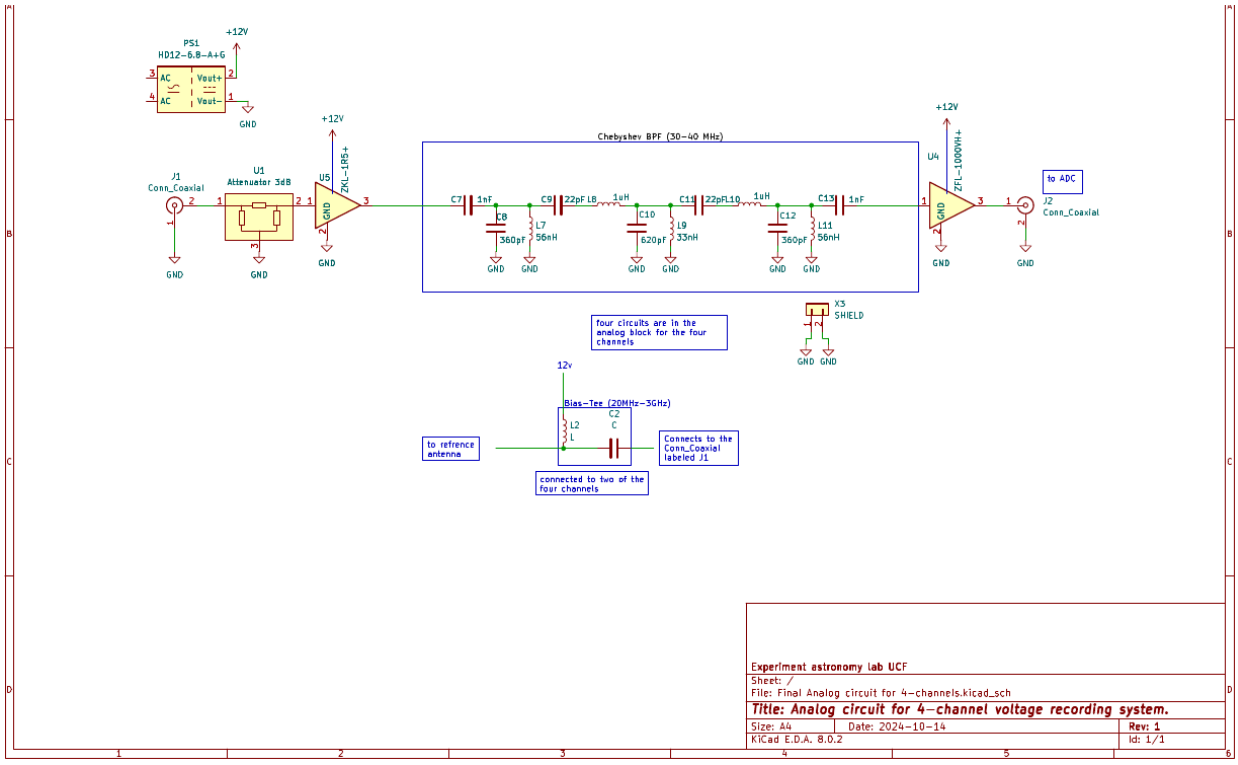

Figure 2: The circuit schematic of the signal conditioning unit of the 4-channel voltage recording system.

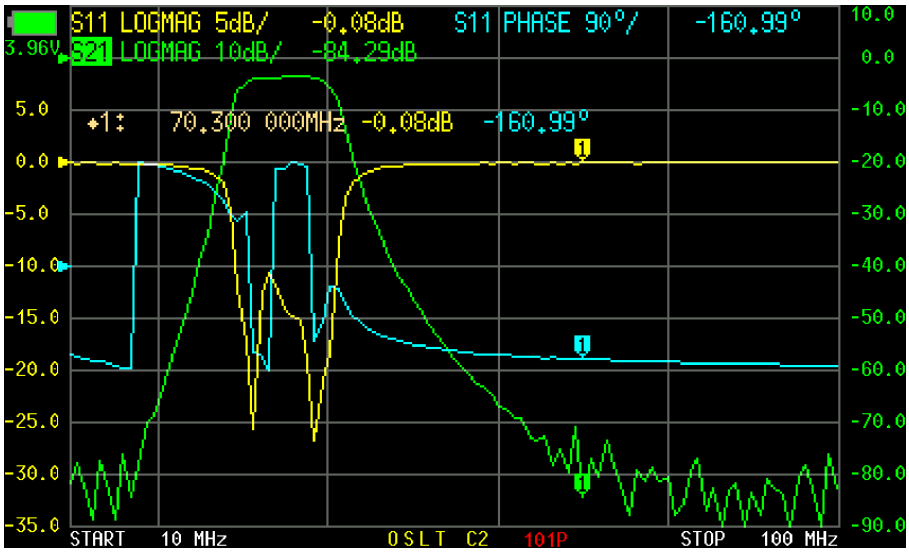

Figure 3: The scatter parameters S11 and S21 of the custom-designed 5th order Chebyshev-Type lumped element band-pass filter measured using NanoVNA. The filter was assembled on a standard FR-4 printed circuit board (PCB).

1.2 Firmware and Modes of Operation

Figs. 4 and 5 display the Simulink blocks used for the FPGA firmware. We based our design on an example provided by CASPER and made modifications to suit our needs. The CASPER blocks utilized are highlighted in yellow. We employed CASPER's default fabric clock frequency of 245.76 MHz for our design.

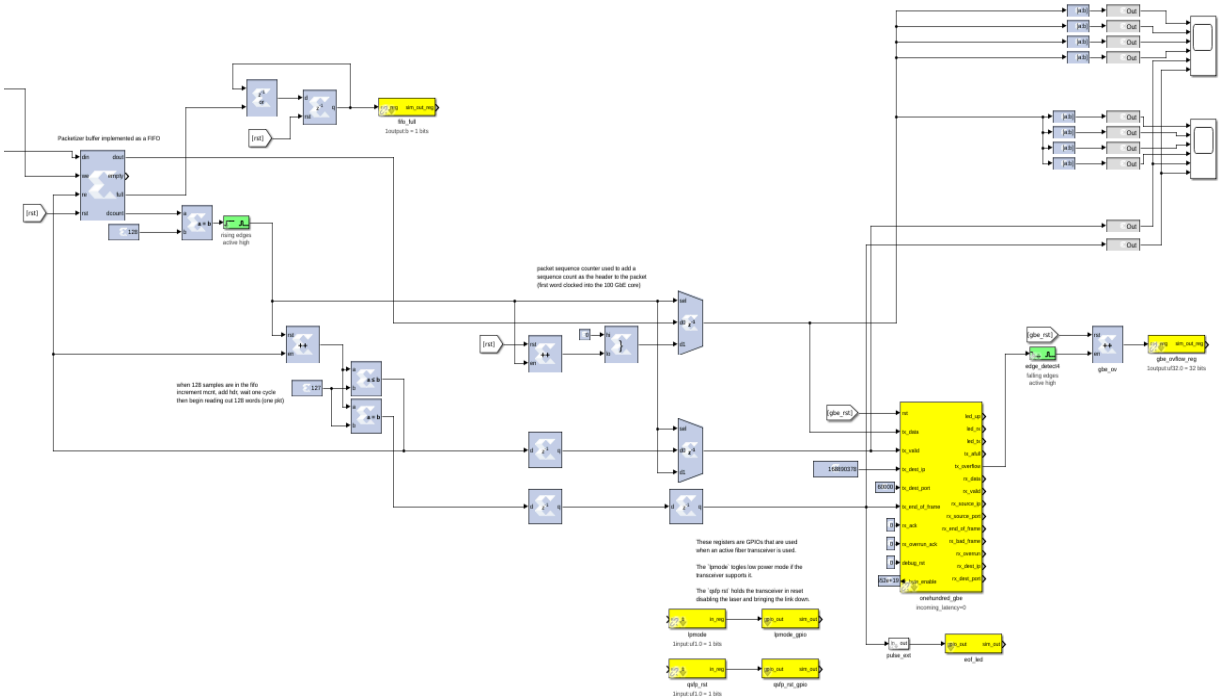

Figure 4: The integrated 100 GbE ethernet block and packetizer of the firmware. These blocks were taken from the CAPSER example design.

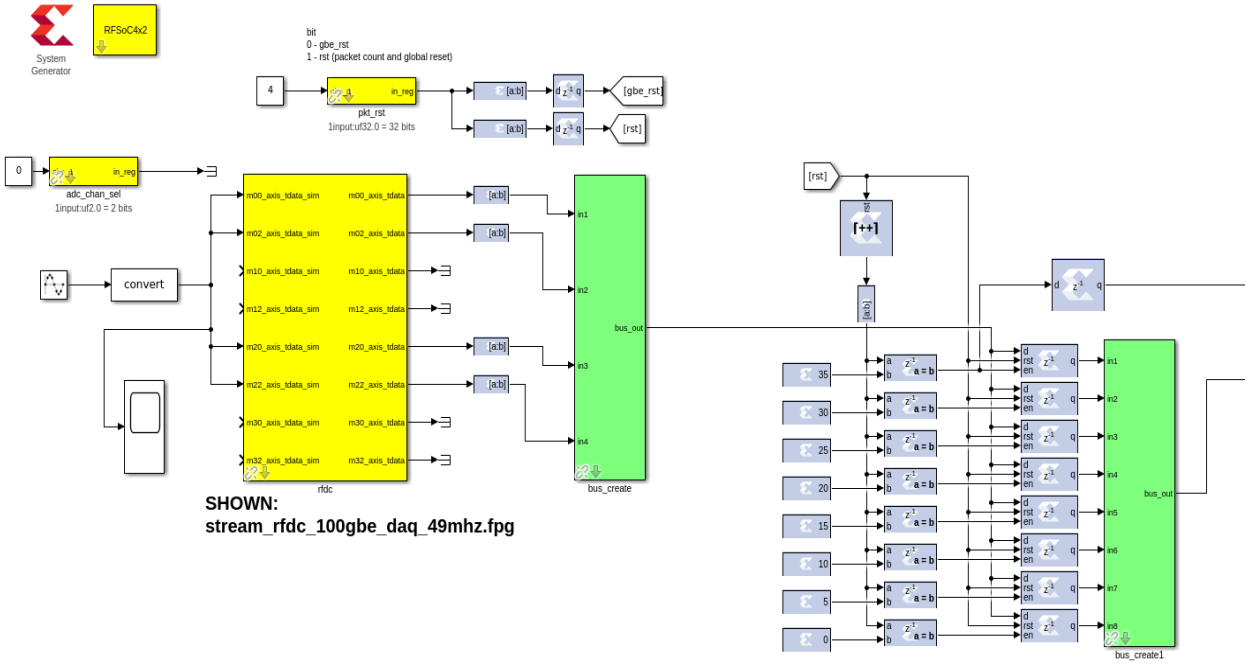

Figure 5: The RF Data Converter (RFDC) block and sampling frequency selector of the firmware. The ADCs are embedded in the RFDC block. The diagram shows the constants in the selector for 49.15 MHz sampling frequency.

The Ethernet block collects data from all four onboard ADCs, packages it into UDP packets, and transmits it to the server via a 100 Gbps Ethernet interface (see Fig. 4). Each UDP packet consists of a 64-byte header and 8192 bytes of data sourced from the four ADCs. The header includes a 64-bit packet counter, incremented by the hardware for each transmitted packet. The 14-bit outputs of the ADCs are extended to 16-bit values by adding zeros to the least significant bits. 1024 such 16-bit values from the four ADCs make up the 8192 bytes of data in a UDP packet. Including the header, the total size of each UDP packet is 8256 bytes.

The RF Data Converter (RFDC), an AMD logic core, configures the ADCs. The RFSoC board provides access to four ADCs: ADC_A, ADC_B, ADC_C, and ADC_D. ADC_A and ADC_B are in tile 226, while ADC_C and ADC_D are in tile 224. The CASPER block RFDC is configured to sample ADCs at a rate of 1966.08 Msps. The ADC data is first decimated by a factor of 2, and then four successive samples are streamed through the four outputs of the AXI interface. We select one sample per AXI output, resulting in an effective sampling rate of 245.76 Msps. Additionally, we use the sampling frequency selector logic to reduce the sampling rate further (see Fig. 5). For instance, Fig. 5 illustrates a decimation factor of 5 using the sampling frequency selector, yielding a sampling rate of 49.15 MHz.

We have developed three different FPGA firmware versions, each implementing a unique mode of operation.

1. The first mode samples at 49.15 MHz, providing a maximum bandwidth of about 25 MHz.
2. The second mode samples at 61.4 MHz, with a maximum bandwidth of about 30 MHz.
3. The third mode operates at 245.76 MHz, providing a maximum bandwidth of around 122 MHz.

The first and second modes continuously record voltage data to SSDs for approximately two hours. In these modes, the ADC data from four channels is transferred from the FPGA at a rate of about 3.7 Gbps. With a total SSD capacity of 4 TB, we can record data from the four channels for up to two hours. The third mode aimed to establish a real-time RFI mitigation system utilizing a Graphics Processing Unit (GPU). In this model, data could be captured non-contiguously and stored on disk for validation.

1.3 Data acquisition software

We developed a hashpipe-based data acquisition software to receive UDP packets from the FPGA board and record them onto SSDs. The software writes the 8256 bytes in each captured packet to a binary file. The program generates a sequence of output files, each with a maximum size of 128 GB. Additionally, a Python wrapper for the hashpipe program was developed, which will take the duration of data acquisition in units of seconds as input.

2.0 Data Processing Software

The offline data processing begins by examining the counter values in the header to identify any packet loss. We customized FFTW-based software developed for the Murchison Widefield Array project to compute offline self and cross products from the recorded voltages. The program reads the binary files, computes the Fast Fourier Transform (FFT) for a specified number of spectral channels, calculates ten products for four inputs across each spectral channel, averages these products over a specified number of FFTs, and records the self and cross products in two separate files for each 128 GB data file. All subsequent post-correlation operations are conducted using the data from these output files.

3.0 Data visualization

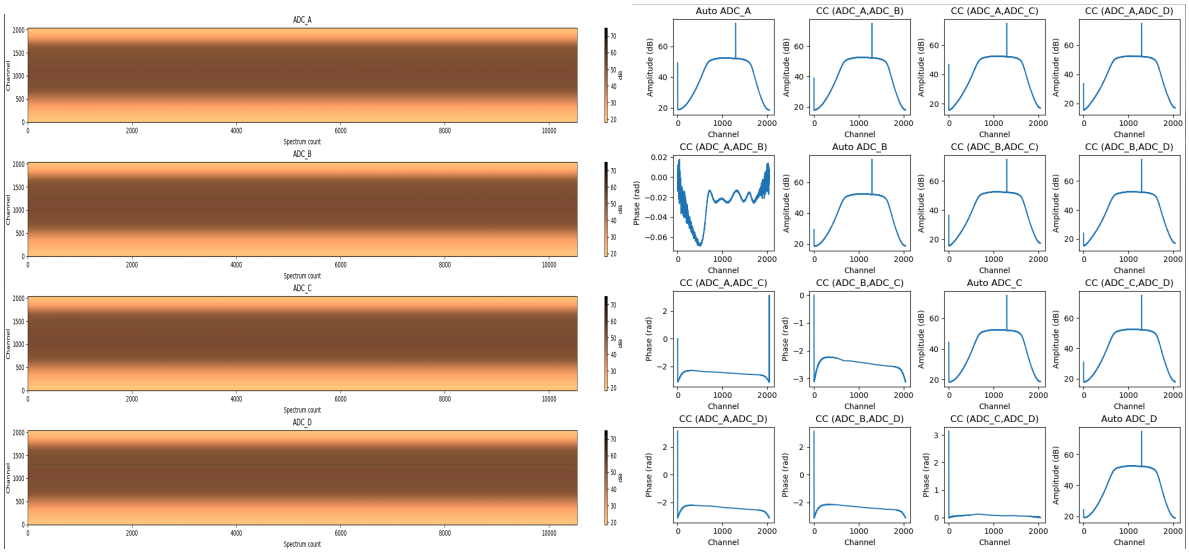

Figure 6: Outputs from the data visualization program examineCC.py. On the right, the self-correlation spectra (or power spectra) from four channels over time are displayed. The correlation matrix for a specific time point is shown on the left. The diagonal plots represent the self-correlation spectra, while the upper triangle and lower triangle plots illustrate the amplitude and corresponding phase of the cross spectra, respectively. The user can select the data for the correlation matrix plot from the display on the right using the cursor.

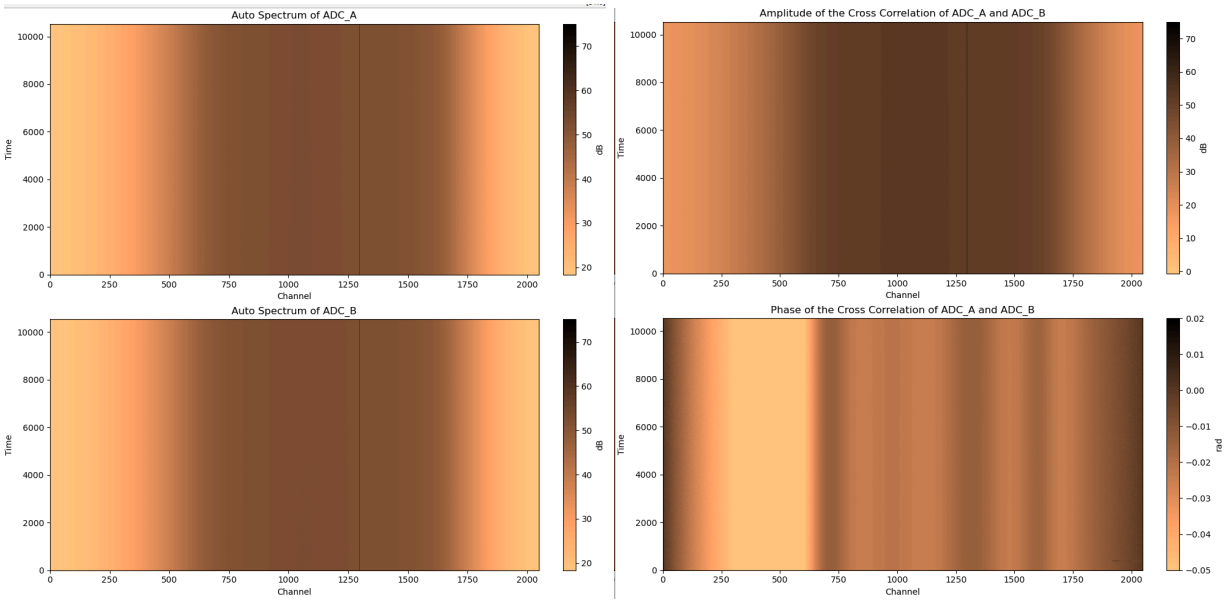

Figure 7: Outputs from the data visualization program graysCCandAuto.py. The program generates plots of the self-correlation spectra versus time (on the right) and the cross-spectral amplitude and phase versus time (on the left) for a specified ADC pair.

We developed data visualization programs that read self and cross-products, displaying selected data in different formats. In the first program, examineCC.py, users can select data from the self-correlation spectra versus time plots using a cursor. The amplitude and phase of all the cross products for the selected data are then plotted in a matrix format (see Fig. 6). In the second program, graysCCandAuto.py, users specify an ADC pair, and the self and cross spectra are plotted as a function of time (see Fig. 7).

4.0 Results from the lab test

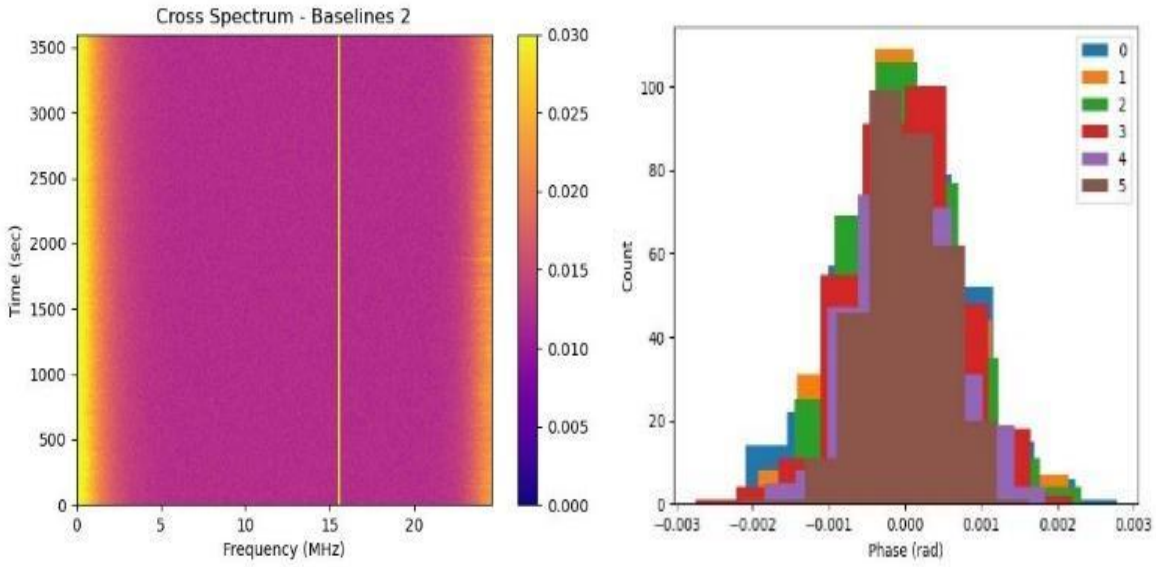

Figure 8: (Left) ADC_A -- ADC_B cross-correlation spectra over 1 hour (see text). (Right): The histogram of the phase of all cross-correlations for 1 hour.

Testing was conducted in the lab to evaluate the performance of three different operational modes, with the results presented in Fig. 8. In the 49.15 MHz sampling mode, data was collected for one hour using correlated noise and tone as the input. The cross-correlations among all four channels were computed offline, and the mean cross-spectral values were subtracted from each frequency channel. Fig. 8 (left) illustrates the ADC_A and ADC_B pair results. The increase in correlated power near 15.6 MHz, as seen in Fig. 8 (left), is due to the injected tone. A histogram displaying the phase of all the cross-products obtained during the one-hour data collection is presented in Fig. 8 (right). These plots demonstrate the stability of the system over this one hour. Similar testing was also conducted for the 61.4 MHz and 246.75 MHz modes of operation.

Acknowledgment

We would like to thank Mitch Burrett, Wei Liu, and Dan Werthimer from CASPER for their technical support, valuable suggestions, and advice during the development of the system. Additionally, we acknowledge the support of Prof. Rafael A. Rodríguez-Solís from the University of Puerto Rico, Mayagüez, throughout the project.

References

[1] D. A. Roshi, “Noise Figure Analysis of a 0 - 2.5 GHz directly digitized radio astronomy receiver system”, CARSE report, July 2024, https://carseuprm.org/resources/publication-rscs/

[2] J. F. Helmboldt, B. B. Markowski, D. J. Bonanno, T. E. Clarke, J. Dowell, B. C. Hicks, et al., 2021, The deployable low-band ionosphere and transient experiment. Radio Science, 56, e2021RS007298. https://doi.org/10.1029/2021RS007298