Russell Silva
AMRUPT, Summer ‘18
Empirical Tests No. 1

Empirical Tests No. 1

I. Overview

My original intention for the first round of empirical testing was to quantify the level of accuracy and real time precision of the developed subspace Angle of Arrival (AoA) system. Due to inconsistent and disoriented Angle of Arrival measurements from the first round of testing, and the number of technical issues and fixes which have occurred, it is clear that the system is in a debugging stage rather than an experimental one.

This document is split into two parts. The first entails the planning and setup of the testing environment and equipment, along with a brief description of the source code prepared for the tests linked on GitHub. The second entails the results of the tests, issues which have undermined the accuracy and real time precision of AoA measurements, and possible solutions whether implemented or not yet implemented. Please read the project proposal beforehand for better understanding of this document.

II. Testing Planning/Setup and Prepared Software

II. i. Equipment/Setup

The current list of equipment for the test includes the coherent receiver, usb hub, raspberry pi, ethernet cable, 8 male to female sma cables, 4 antennas, two laptops, one transmitter cc1310, a Back-UPS 450, an RF signal generator, and a testing rig for antenna array placement. The testing rig is shown in Figure 1. The top mast holding the antenna array is 6 feet above the ground and 5 feet wide. A platform was placed towards the bottom of the rig which held the coherent receiver, usb hub, raspberry pi, and ethernet cable connection. This platform was placed 2.5 feet above the ground and 3.5 feet away from the top mast. These metrics were chosen to eliminate RF interference from near field coupling [1]. To compensate for the low platform placement, two male to female sma cables had to be used for each antenna-receiver connection. The near field of the antenna array for 434 MHz would be one wavelength (~0.7 meters).

Figure 1: Testing Rig

One laptop was used to establish a VNC connection to a Raspberry Pi via ethernet so that on-site debugging could be performed when needed. Another laptop was used to control the CC1310 to easily set the frequency and Tx power of unmodulated transmissions with SmartRF Studio. Only one laptop is needed indoors if Wifi is present (the VNC connection to the Raspberry Pi can be made wirelessly).

The four RTL SDR radio receivers which were used for direction finding were held within the coherent receiver. The coherent receiver contained a noise card, supervisory RTL SDR, and an external connection to an RF Signal Generator to correct for timing and phase offsets between the RTL SDRs. Each RTL SDR was connected to input ports on a usb hub. The usb hub had one output port connected to the Raspberry Pi. The command “rtl_test” was entered in the terminal before testing, to ensure that no samples were lost from the 4 to 1 usb connection.

The Back-UPS 450 was used to power electronic devices in outdoor environments. It had a 1-hour battery life when the RF signal generator, laptop, Raspberry Pi, usb hub, and coherent receiver were connected.

The testing rig would be setup in an outdoor and indoor environment for initial testing. The outdoor environment was a turfed baseball field next to Phillips Hall, Cornell University. The indoor environment was in the Duffield Hall atrium, Cornell University. The testing rig was placed in the middle of the baseball field for the outdoor test and was centered in an area with few surrounding objects in the indoor atrium.

II. ii. Overview of Software

GNU Radio is a free, open source software that provides signal processing blocks, and can display a streamlined interaction between these blocks in a GUI called GNU Radio Companion. GNU Radio was primarily used because it implements the high-speed ring buffering protocols needed to maintain synchronicity between high data rate (2 Megasamples per second) streams [2]. The first flowchart was a two-antenna element program which would use the phase difference of an incoming radio signal at two antennas to compute an angle of arrival [4]. The second flowchart implemented root MUSIC, a subspace technique that estimates an angle of arrival based on the roots of a polynomial determined by the eigenvector analysis of a sensor array correlation matrix [5]. The flowcharts along with the Python and C++ programs associated with the blocks within the flowcharts have been pushed to the AMRUPT GitHub here. Please note that “alpha” and “normalized spacing” in the two-element flowchart and root MUSIC flowchart respectively have been corrected to 0.5 meters (each antenna in the array is spaced at half the wavelength for a 433.92 MHz signal).

The two-antenna element flowchart (based on Whiting et. al) is displayed in Figure 2. Two RTL-SDR sources are sent through virtual sinks which sends the raw I/Q samples to be used in cross correlation for timing offset corrections and phase difference analysis at the same time. The cross correlation is performed in the sample offset blocks on top which computes a delay between the data channels based on a 3 iteration FFT convolution in the frequency domain. The delay then applied by the delay blocks below. An important note is that the number of iterations specified in the sample offset blocks cannot be set too high, or the processing demands of this block will cause a loss of timing synchronicity especially when comparing more than 2 data streams.

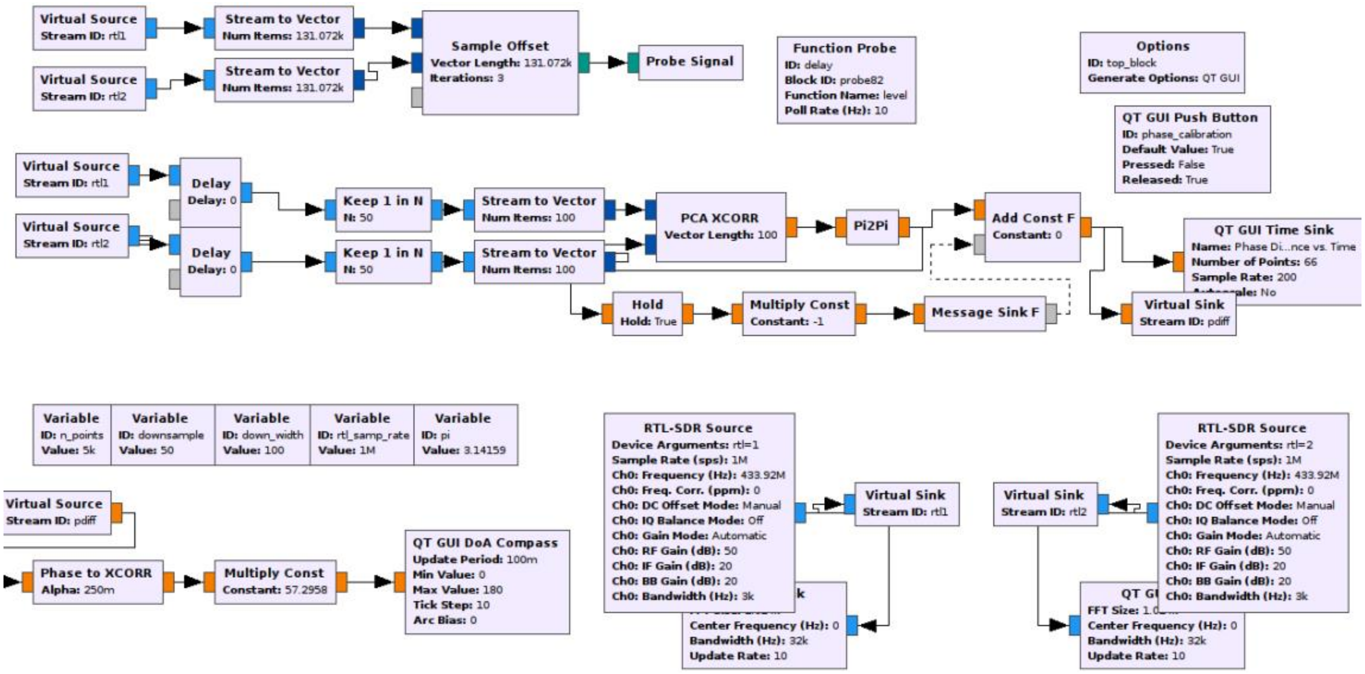

Figure 2. Two-antenna element AoA flowchart based on the phase difference between receivers

Once timing offsets are corrected for, the two RTL SDR datastreams are interpolated to include 1 out of every 50 samples for further DSP analysis. 100 samples are placed in a vector after this interpolation and sent to a block called "PCA XCORR." The "PCA XCORR" block code here uses Capon Beamforming to compute a phase difference between the signals received by two RTL SDRs. This block was found to produce a very stable phase difference between two RTL SDRs when a common sinusoidal signal was supplied to the receivers (less than 0.1 radian variation). The "Pi2Pi" block wraps the float result from "PCA XCORR" into a range from -pi to pi.

Once a phase difference is reliably computed, a phase offset can be corrected by pushing a button named "phase_calibration" on a GNU Radio GUI during runtime. The phase calibration button should be pressed when the common sinusoidal signal is being applied to both receivers. When the phase_calibration button is pressed, the phase difference registered by "PCA XCORR" will be sent in a message format into the "Add Const F" block, which will continuously subtract the offset before the phase difference data stream is sent to the Angle of Arrival computation. The Angle of Arrival is computed by the "Phase to XCORR" block using the calculation `DOA = arccos(phase/(2*pi*alpha))` where alpha is determined by the antenna spacing and the wavelength of the received signal. The flowchart has since been updated to include a float sink as well as a DoA Compass for displaying AoA values.

The root MUSIC flowchart is displayed in Figure 3. Timing offsets are computed using three sample offset blocks which send delay values respective to the RTL SDR combination (delay1, delay2, and delay3) variables listed as "Function Probe" hold these values.

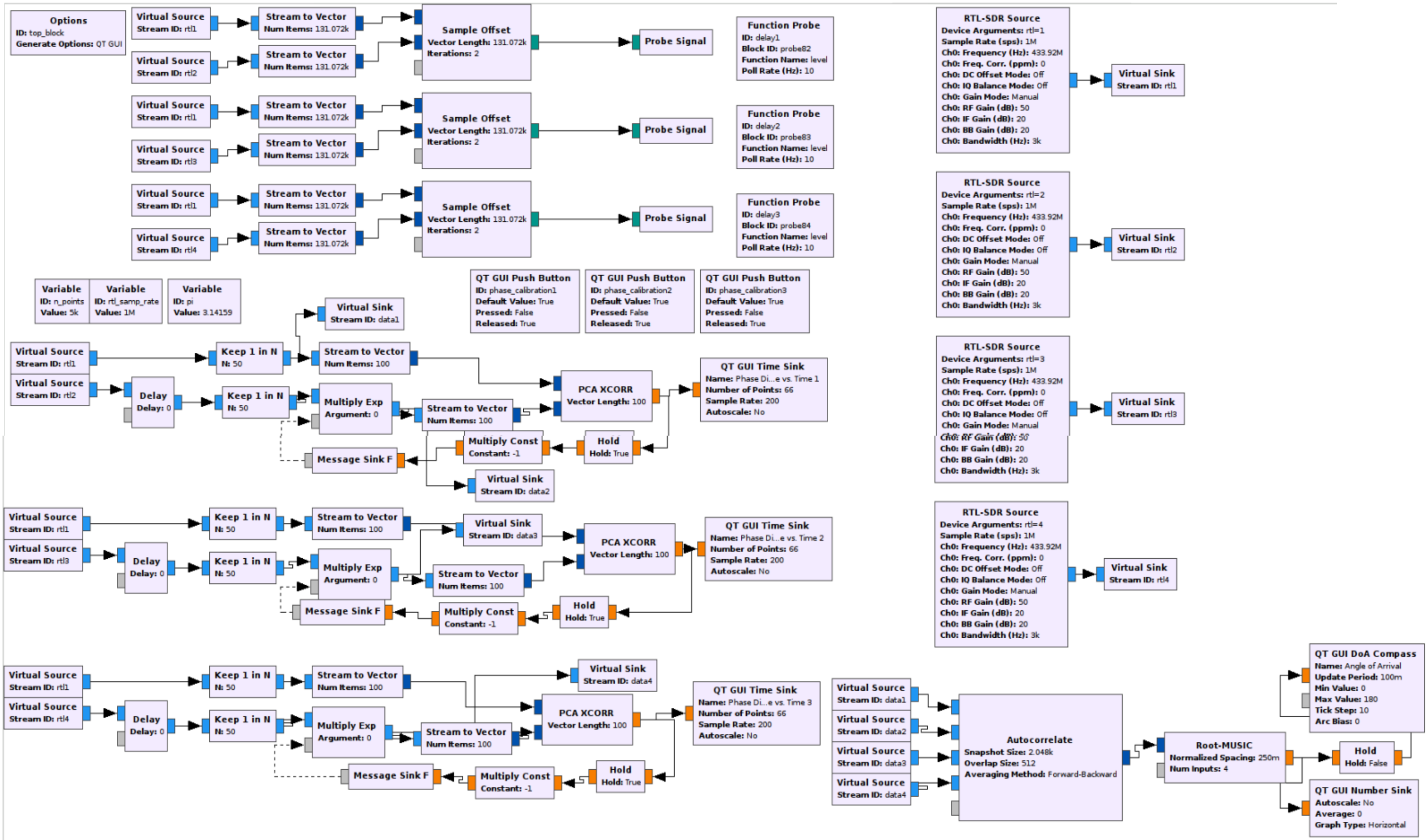

Figure 3. Root MUSIC AoA flowchart

After timing offsets are corrected between the RTL SDR channels, phase offsets are corrected using a similar procedure to the previous flowchart. Instead of directly altering the phase difference computed by PCA XCORR, the Multiply Exp block was used for phase offset correction was used so that the I/Q streams can be phase corrected instead of the phase difference values. This is done because the autocorrelation block takes in I/Q streams and not phase difference streams. The Multiply Exp block alters the phasors which changes the complex values of the signal directly by multiplying the phasor exponent by the phase offset. This method has been tested to yield stable phase difference results as the previous method.

During root MUSIC testing the following protocol would be performed on the user end for timing and phase offset correction. The RF signal generator was turned off and the noise source was turned on by a separate terminal before the start of the program. On startup, the program computes and applies timing delays between the channels, which would cause the phase difference display on the GUI with the noise source on to steady to a phase difference with +/- 0.2 radian jittering. Afterwards, the noise source was turned off, and a common sinusoidal signal source was fed into the first two receivers on the coherent receivers. The phase_calibration1 button was pressed to correct for a phase offset when this common sinusoid is sent to RTL SDR 1&2. The same is then done for RTL SDR 1&3 and RTL SDR 1&4. Finally, the RF signal generator is turned off and the four antennas are connected to each receiver accordingly. The GUI Compass/float sink can then be monitored to record AoA values.

III. Testing Results and Improvements

III. i. Testing Results

I performed all the tests at a 433.92 MHz center frequency, 3k receiving bandwidth, and 50 RF gain at each receiver. I conducted the first test outdoors, and the second test indoors. From the results of these tests, I have already made several improvements to the system (to be described in section III. ii.). After making these improvements, a third indoor test was conducted. All the tests were performed with a fellow Cornell Student who helped carry equipment and hold the transmitter/computer. To properly test the AoA accuracy and real time precision of this system, calculated AoAs were to be recorded at different set measured angles and distances from the receiver basestation for both flowchart programs.

The following chart was prepared to record AoA results at multiple angles, at different distances prior to testing:

Angle of Arrival (2 element AoA or root MUSIC)

AoA RPi3	AoA Measured (Compass)	Distance
	0	~20 feet
	30	~20 feet
	60	~20 feet
	90	~20 feet
	120	~20 feet
	150	~20 feet
	180	~20 feet
	60	~40 feet
	90	~40 feet
	60	~60 feet
	90	~60 feet

Unfortunately, the AoAs for the two element AoA program could not be recorded because the phase difference would fluctuate very rapidly when moving the slightest amount along an imaginary line determined by the measured angle. This effect was recorded here. Due, to this effect the AoA at a specific location was too unstable to record. Furthermore, impossible phase differences between the two antennas were registered for the test's specification. At 433.92 MHz, the maximum phase difference that can occur between two antennas spaced at 0.345 meters (quarter wavelength spacing) is a half radian. Phase differences from a half radian to a full radian were observed during testing.

The root MUSIC program generated NULL AoA values for 433.92 MHz (this was observed from the GUI compass and float sink). It is interesting to note that this did not happen at twice the frequency. A program based on the original MUSIC algorithm was briefly tested to see if it had the same effect, and it did not; although its results had the same instability issues as the two element AoA program. (This issue was recently fixed, see Solution in the Works 3 in the next section).

III. ii. Issues, Reconciliations, and Solutions

Issue 1: The GUI Compass may have an instability issue with reported issues here and here. Therefore, the following tests will be done using the console or a GUI float sink (displaying the float value result of MUSIC/root MUSIC on the GNU Radio GUI) in addition to the GUI Compass to see if there is disparity between the end AoA results.

Reconciliation 1: The float sink and GUI compass display the same values, so the compass is likely not a problem. The float sink will still be used for further validation.

Issue 2: Constant sinusoidal amplitude disparity between receiver channels (independent of transmitter location). This was observed when a common sinusoidal signal was fed into each receiver as well.

Solution 2: The powering of the bias tee on each receiver (from the RTL source block code here – bias tee powered on line 233) fixed this constant sinusoidal amplitude disparity. I made this improvement because it is a necessary step according the coherent receiver technical support instructions and because an antenna signal would be blocked off from the bias tee routing more power to the noise input channel (even when the noise source is off). An interesting result of the bias tee power-up is the reduced the phase offset between the noise phase difference and the signal phase difference between receivers during phase offset correction which can be seen in the testing video here.

Issue 3: Sinusoidal amplitude disparity dependent on transmitter position (amplitude disparity between the received waveforms displayed from each receiver when a 433.92 mHz tone was sent from the cc1310) and rapid phase difference fluctuation between any two antennas on the array when transmitter location is changed slightly. The phase difference between antenna elements should not change as the tag moves along the same angle-based line. Phase difference fluctuations were observed to go from -2pi to 2pi radians in just a couple of meters along the same angle-based line (angle-based line: an imaginary line corresponding to an angle of arrival). for all tests so far. Furthermore, impossible phase differences such as a 2pi phase difference is being registered for antennas spaced a quarter wavelength apart.

Solution in the Works 3: After speaking with Xiaonan Hui, a postdoctoral student in Dr. Kan's lab, we concluded that this issue is likely a multipath problem. This was initially difficult to pick up as the source of the issue, since the sinusoidal waves at receivers are periodic and very stable. However, this is the case for coherent reflected and line of sight signals from a single source, in which the reflected and line of sight phasors add to change the amplitude and phase of signal impinging on the antenna array, but not its period or waveform shape. Since the receiving antennas are omnidirectional, all reflected transmitted signals from all possible directions are included in this result.

The forward backwards smoothing in the ETTUS autocorrelation block does not directly mitigate multipath interference. It was my initial belief that the ETTUS autocorrelation block performed forward backwards spatial smoothing; however, it in fact performs forward backwards smoothing in time between samples. I recently programmed a forward spatial smoothing module here based off of [3] which averages the subarray matrices produced by several autocorrelation blocks.

Another way to minimize multipath is to reduce the gain from the transmitter, so that the LOS signal is more prominent among lower power reflected signals. The Tx gain can be set to lower dB values from smartRF studio during further testing.

During initial testing, the transmitter was near to a human (me) and a laptop, which could present a proximity multipath problem. The CC1310 has been placed at the end of a wooden plank (see Figure 4) and attached to a usb extension cable to minimize multipath interference from the transmitter. When the change was performed, root MUSIC did not produce anymore NULL AoA values. In future testing, a JSC to SMA connection will be made so that the orientation and polarization of the receiver and transmitter antennas are the same. The JSC to SMA connection will only be made after the CC1310 is opened with smartRF studio, because the CC1310 runs an example code at very high Tx power which overuses the power supply of the laptop usb port. This example code halts when the CC1310 is opened with smartRF studio.

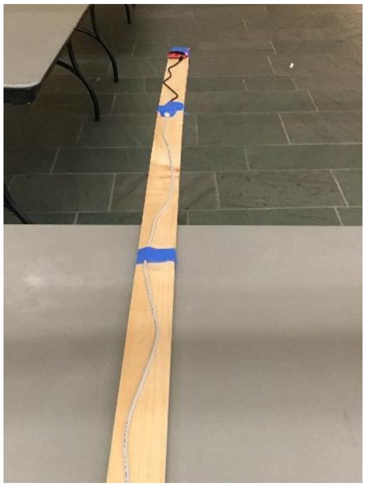

Figure 4: CC1310 Transmit Structure

I will check if our system produces reasonable AoA values using the changes mentioned in "Solution in the Works 3," and perform further debugging indoors before proceeding to an outdoor test. This is because moving all of the equipment outdoors will take considerable time and effort, and similar AoA systems have worked in similar indoor environments in close proximity to the antenna array (please see here and here). I want to make sure the AoA system developed is at its full potential without any technical issues before performing outdoor testing.

IV. References

[1] https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6476659
[2] https://www.gnuradio.org/blog/buffers
[3] https://github.com/jakapoor/AMRUPT/blob/master/Literature/General_radio_direction_finding/Angle%20of%20Arrival%20Methods/Direction%20of%20Arrival%20Estimation%20in%20a%20Multipath%20Environment%20an%20Overview%20and%20a%20New%20Contribution.pdf
[4] https://github.com/samwhiting/gnuradio-doa
[5] https://github.com/EttusResearch/gr-doa