Russell Silva
4/17/2018
AMRUPT, Spring '18

Weekly Report #6 – Coherent Receiver Hardware/Software Setup

Goals

Finish GNU Radio and RTL SDR Driver software installation and setup on the Raspberry Pi. Once this is complete, use the rtl-sdr IQ recorder utility to capture raw data. Finally, implement cross correlation and subsampling technique developed by Sam Whiting, et al. in GNU Radio, to obtain reliable AOA calculations from synchronized samples.

Problems

Not one single RTL SDR on the Coherent Receiver device can be used unless all four RTL SDRs and the supervisory/single channel receiver are powered via usb connection. Because of this, work on the coherent receiver was delayed until Monday (waiting for a shipment of usb extenders).

General Approach

Since programming could not be done directly on RTL SDRs for the majority of this week, work was more focused on planning out the next stages of the hardware/software setup. Relevant computer software such as GNU Radio and SDR# (RTL SDR driver files) was installed and used briefly for understanding the functionality of these programs. Primarily, the RTL SDR source feature on GNU Radio was explored, such as adjusting the frequency and RF gain of the RTL SDR before retrieving samples from it.

We plan on installing the RTL SDR drivers and testing out the basic functionality of each radio receiver according the quick start tutorials shown in [1] and [2]. The Multi-channel RTL-SDR support page for direction finding on the coherent receiver in [4] states recommends installing [3] for more improved and up to date RTL SDR drivers than provided in SDR#. Basic I/Q samples will be collected and written into a file using the rtl_sdr command as gone over in [4] and [5].

The tentative strategy as of now (based on the management plan) is to complete the fundamental hardware/software setup, and setup testing (to verify connections, noise to RTL SDR source switching, etc.) by next week once basic I/Q extraction has been achieved. Following this, the completion of the GNU Radio flowchart and custom radio blocks is planned to take week and a half. We hope that this plan will be followed at an appropriate level so that high-level debugging, code optimization, and testing under multiple environments can be performed in the last weeks of this semester.

Code-level problems and solutions, and empirical testing

As shown in [4] and [5], the following code will be used for preliminary I/Q data extraction:

```
rtl_sdr -d0 -f 1125000000 -g 35 -s 2500000 -n 50000000 -N FMcapture0-2.dat &
rtl_sdr -d1 -f 1125000000 -g 35 -s 2500000 -n 50000000 -N FMcapture1-2.dat &
rtl_sdr -d2 -f 1125000000 -g 35 -s 2500000 -n 50000000 -N FMcapture2-2.dat &
rtl_sdr -d2 -f 1125000000 -g 35 -s 2500000 -n 50000000 -N FMcapture2-2.dat &
```

Relevant Links

[1] https://www.rtl-sdr.com/rtl-sdr-quick-start-guide/
[2] https://learn.adafruit.com/getting-started-with-rtl-sdr-and-sdr-sharp/download-and-install-software
[3] https://www.rtl-sdr.com/testing-keenerds-rtl-sdr-drivers/
[4] https://coherent-receiver.com/getting-started
[5] http://aaronscher.com/wireless_com_SDR/RTL_SDR_AM_spectrum_demod.html