Google Summer of Code - 2018

Passive radar capability to gr-radar toolbox

Suraj Hanchinal

Introduction

Radar toolbox which was developed by Stefan Wunsch in GSoc 2014 was a great success and included a variety of tools such as target simulator as well as OFDM signal support. This toolbox was a great addition to the GNU Radio OOT modules. The toolbox mainly provided support for active radar.

Even though the initial toolbox is impressive, it is lacking functionality such as passive radar support and support for devices other than USRPs. Adding support for devices like RTL-SDR and WiFi routers and other SDRs might increase the functionality of the toolbox.
Passive radars are generally easier to setup and use as you do not need a powerful transmitter or a permit and you just need a device to receive the radio waves and process them. This ease of use can translate into better usability and better applications.

My proposal consists of implementing passive radar capabilities along with their simulation to the gr-radar toolbox and adding support for devices other than USRP for passive radar.

Section 1 describes the overall structure of the project and the flow of the passive radar. Section 2 describes the passive radar algorithms going to be implemented in the toolbox. Section 3 describes the deliverables to be expected at the end of GSoc '18. Section 4 contains the timeline for the entire project and Section 5 lays out my qualifications.

Structure of the project

Signal Acquisition: This is the first step of the flow of passive radar. The signal source can be anything from a virtual signal source to a RTL-SDR to a proper software defined radio. The signal may be single antenna or multiple depending on the type of algorithm being used.

Processing Blocks: These blocks take input in form of messages containing the acquired signals above and process them. The algorithms and methods used for processing are described in the next section.

Filter for Object tracking: Although particle filter as well as Kalman filter have been implemented, there has not been a filter which extracts the position of a moving object with all the pre-processing required i.e. remove all the clutter etc. This will be added as a optional filter for the application part of the toolbox.

Visualisation: The data processed by the processing blocks has to be presented in a understandable way. One of the necessary visual tools for radar is doppler diagram which has been already added to the gr-toolbox. This will be modified to work with the passive radar. Also GUI for the object path will be added using the filter described above.

Simulation: A very important feature of the original toolbox was the target simulator block which used the point-target radar model to simulate the RX signals. A simulator will be built for the passive radar with features like multiple receiver antennas and multiple TX "illuminators". For example, you could simulate a triangulation using 3 fm stations with a receiver antenna using the simulator.

Passive Radar Algorithms

Cross-Correlation Algorithm

Passive radar unlike an active radar deals with a lot of incomplete information like where is the transmission source? how far is it? What modulation is it using ? To counter these problems, there are usually two antennas, one of which is pointed towards the transmitter. The received signal on this antenna is called the **reference signal**. The other antenna is pointed towards the object of interest. The received signal from this antenna contains a lot of clutter from large objects such as mountains as well as a strong direct path signal from the transmitter itself. This clutter can be removed using various methods such as linear least squared estimator. Once the clutter is removed the signal is cross-correlated with the reference signal to obtain the doppler diagram. This is the conventional algorithm for passive radar. This algorithm can be used even with WiFi routers [1] . Using the gr-802.111n module of GNU-Radio, we can use WiFi routers as “illuminators” and the SDR as a receiver and localise and locate objects. This will be the first algorithm for passive radar. As mentioned above this will be modular and can be used with different devices such RTL-SDRs and WiFi routers etc.

OFDM signal reconstruction Algorithm

The second major algorithm this project will focus on OFDM modulated waves in specific. This will mainly follow [2], a research paper describing a system of passive radar using wifi routers as the “illuminators” and a single SDR antenna as the receiver. Instead of having a reference signal, the original signal is reconstructed. The neat thing about the research paper is it also provides a simulation model which can be added as a simulation for the toolbox. This algorithm is a more application based and uses already existing architecture such as WiFi routers in buildings. This algorithm will be easier to use as compared to say active radar since most of the architecture is already setup i.e. the routers. This is not specific to Wifi routers though and can also be used with DAB modulation scheme which also uses OFDM.

Deliverables during GSoc 2018

The main objective will be implementing passive radar methods to the gr-radar toolbox. At the end of the coding period the deliverables will be:

1. Processing Blocks for the two algorithms described above.
2. A clutter reduction block which removes clutter and gives information about the object of interest.
3. A visualisation tool for viewing the results from the processed data. This includes Doppler Diagram, Object path graph and more.
4. A simulator block which lets you add multiple illuminators as well as antennas to simulate passive radar as an extension to the current target simulator.
5. Documentation of the entire coding process as well as all of the tools.

The deliverables may be extended to add more algorithms or other blocks after further discussion with the mentor.

Timeline

The following is the timeline planned for the project and I will try very hard to follow all the set deadlines here. Documentation will be written alongside the development.

April 23rd to May 14th: I will familiarise myself with the GNU-Radio community as well as discuss the minute details of the project with my mentor. Since I have already gone through the code. I will also spend this time setting up a testbench for experiments since some of the features implemented need to be tested in real life.

May 14th to May 21st: The first week will consist of setting up a code structure and starting to work on the cross correlation based algorithm block.

May 22st to May 29th: Continuation of work on cross correlation algorithm and finishing the basic correlation block i.e. 2 antennas approach. Also the work on clutter removal algorithm is started. This includes reading up on literature on the various clutter removal algorithms.

May 30th to June 5th: A proper clutter removal algorithm is chosen and the coding for this commences. The major priority of this week will be at least to get rid of the strong direct signal interference from the transmitter.

June 6th to June 13th: Completion of work on the clutter removal algorithm and its integration with the cross-relation algorithm. This will lead to completion of the first part of the passive radar algorithm.

June 14th to June 21st: Work is started on the simulator. Firstly the feature adding multiple “illuminators” will be added. Also other simulation models are explored for further improvement.

June 22nd to June 29th: Work on multiple “illuminator” simulation is completed. Work will begin on the simulator involving multiple antennas.

June 30th to July 6th: Work on multiple antenna simulation is completed and work begins on the second major passive radar algorithm - OFDM based passive radar algorithm. Since this involves a lot of math, The paper is studied and the basic code is built in this week.

July 7th to July 14th: Work is continued on the OFDM algorithm. Work is done on the symbol reconstruction algorithm described in the paper.

July 15th to July 22nd: The OFDM radar algorithm is completed and is also tested with the simulator. Work is started on the Visualisation tool by modifying the already existing doppler diagram GUI.

July 22nd to July 29th: The doppler diagram GUI is perfected and the object path GUI is completed.

July 30th to August 5th: The code is formatted properly and finishing touches are given to the entire codebase. The project is complete!!

Personal Information and Qualifications

I am a 2nd year undergraduate Electronics and Communication engineering student studying in **Indian Institute of Technology, Kanpur.** I am **not** getting credit for the GSoC project. I am proficient in 4 languages including English. I will have internet access over the entire course of the GSoC period.

I am proficient in C++, Python, VHDL, C# and javascript. I have read up and studied the **entire codebase** of gr-radar toolbox.

Below are some of the projects I have worked on:

1.I have extensively worked on Kalman filters and real-time data processing in a project on Localization using Gait Analysis using Inertial Measuring Units(IMUs)[3]. This project involved converting a **research paper** [4] into code since the code was not made available by the author This project is written in Python.

2. I have also worked on RTL-SDRs for the project “Encrypted Wireless Video transmission” . Video link is here [5] . I used Raspberry Pi to transmit DVB-S modulated video to an RTL-SDR and demodulated it using leandvb. This project is written in Python. I cannot share the code as this was done as a University Project.

3. I am also working on a **open-source** project “Openage”, which is a free and improved remake of the Age of Empires game engine where I am currently working on the OpenGL renderer. This project is based on C++ as well as GLSL. The renderer is a core part of the engine which is being reworked, so it still hasn’t been merged to the main branch. Here [6] is my fork on which I am working.

My coursework includes signal processing and I have read up on various research papers on passive radar to familiarize myself with it. I also believe that "**cyberspectrum is the best spectrum**".

License

The code written for the project will be open-source and GPLV3 licensed.

Acknowledgements

In the end, I promise to adhere to all the guidelines set by GNU-Radio Organisation as well GSoC . I accept the three-strikes rule and the other rules in the rules and conducts page. I would also like to thank GNU-Radio for providing me this opportunity to contribute to the great open-source community. I would also like to thank **Martin Braun** and **Marcus Muller** for providing valuable guidance while developing this proposal.

Contact Details

Name: Suraj Hanchinal
Email: surajhanchinal@gmail.com
College email: surajvh@iitk.ac.in
Github: https://www.github.com/surajhanchinal

References

[1]:https://pdfs.semanticscholar.org/0a3c/9b13412fc3929f679217d1d4f11497a50e53.pdf

[2]: http://ieeexplore.ieee.org/document/7833233/

[3]: https://www.github.com/surajhanchinal/imu_localisation

[4]:https://github.com/surajhanchinal/imu_localisation/blob/master/IEEE_Trans-on-HMS-2016-1.pdf

[5]: https://photos.app.goo.gl/cR6rCZAerIIDGe963

[6]: www.github.com/surajhanchinal/openage/tree/new-renderer