GNU Radio: Signal Intelligence Toolbox

GSoC 2016 proposal

Sebastian Müller
Karlsruhe Institute of Technology
`gsenpo@gmail.com`

March 25, 2016

1 Introduction

Signal intelligence describes the gathering of information out of intercepted radio signals with unknown origin and unknown parameters.

It is typically used by intelligence agencies to get information about various radio sources where normally a receiver has knowledge about the sender's signal parameters like symbol rate, modulation and frequency band. The purpose of signal intelligence is extracting the transmitted information and to enable receiving of the source signal without any of this knowledge.

GNU Radio has already the ability to do signal intelligence, but it is only possible through some detours. A typical signal intelligence workflow could at this time be recording a broadband spectrum and afterwards searching for signals in the spectrum plot. Then, one signal can be mixed to DC and low-pass filtered. After that, one is able to begin analyzing the time/spectrum/waterfall/constellation plot to make a guess for the modulation. As the signal is recorded, it can be manually synchronized by estimating the symbol rate while examining the time signal. Samples can be moved or thrown away to realize manual synchronization or a synchronization block from GNU Radio can be used. Finally, one can choose the correct demodulation block for the guessed modulation scheme to get the transmitted bits. Real-time signal intelligence is hardly possible now. The tool `gqrx` has some of this ability (real-time AM/FM demodulation), but there is no final solution for this problem at this time. The target of this project is to develop an easily accessible and extendible solution for this workflow.

2 About me

As a graduate student of electrical engineering with focus on communication engineering in my 10th semester and a B.Sc. degree, I have advanced knowledge of signal processing and radio technology. I have chosen various lectures so far that will become useful during this project and have also done multiple projects involving Matlab, C++ and Python during my studies and internships.

During my bachelor's thesis at the Communications Engineering Lab (CEL) of the Karlsruhe Institute of Technology I worked together with Stefan Wunsch at the `gr-radar` OOT module and implemented a radar cross section estimation block [1].

Additionally, I won the 1st price of the IEEE Signal Intelligence Challenge, hosted at the CEL, in 2014 and 2015. During these competitions, I gathered much practial and theoretical knowledge about signal intelligence and signal processing. I used GNU Radio to solve several challenges during these competitions. As a student of the CEL I enjoy access to a big hardware pool in addition to my own Ettus USRP B210.

I have been watching the GNU Radio community for some time now and want to take GSoC as the opportunity to begin with my participation in the project. GNU Radio fits perfectly to my study focus and I am confident that I can contribute valuably to that project.

I have acknowledged and understood the rules of the GNU Radio community.

3 Benefits

The GNU Radio project can benefit from my work in the following ways.

The idea behind software defined radio is a multi purpose solution which can be easily adapted to different radio standards by changing the software. This process can be made significantly easier by automated signal intelligence algorithms. In the ideal case, the user does not have to know any properties of a signal to receive it and extract information from it. The signal intelligence toolkit would make a huge step in that direction.

GNU Radio is predestined to be used by practical beginners in the field of radio engineering. Everyone has access due to the free software status of the project and hardware can easily be bought (e.g. DVB-T sticks). The understanding of radio technology can be improved significantly by examining existing radio signals and finding out **why** and **how** they work. With `gr-sigint`, any beginner can examine radio signals without needing expert knowledge in this field.

4 Deliverables

The explicit components that I plan to develop during my GSoC program are listed below. Methods will be prototyped in Python and finally implemented in C++ afterwards. Since signal intelligence is a vast topic, the most important deliverables where chosen to be implemented during GSoC. In addition, many further deliverables come to my mind for the post-GSoC period, that I outlined in an outlook.

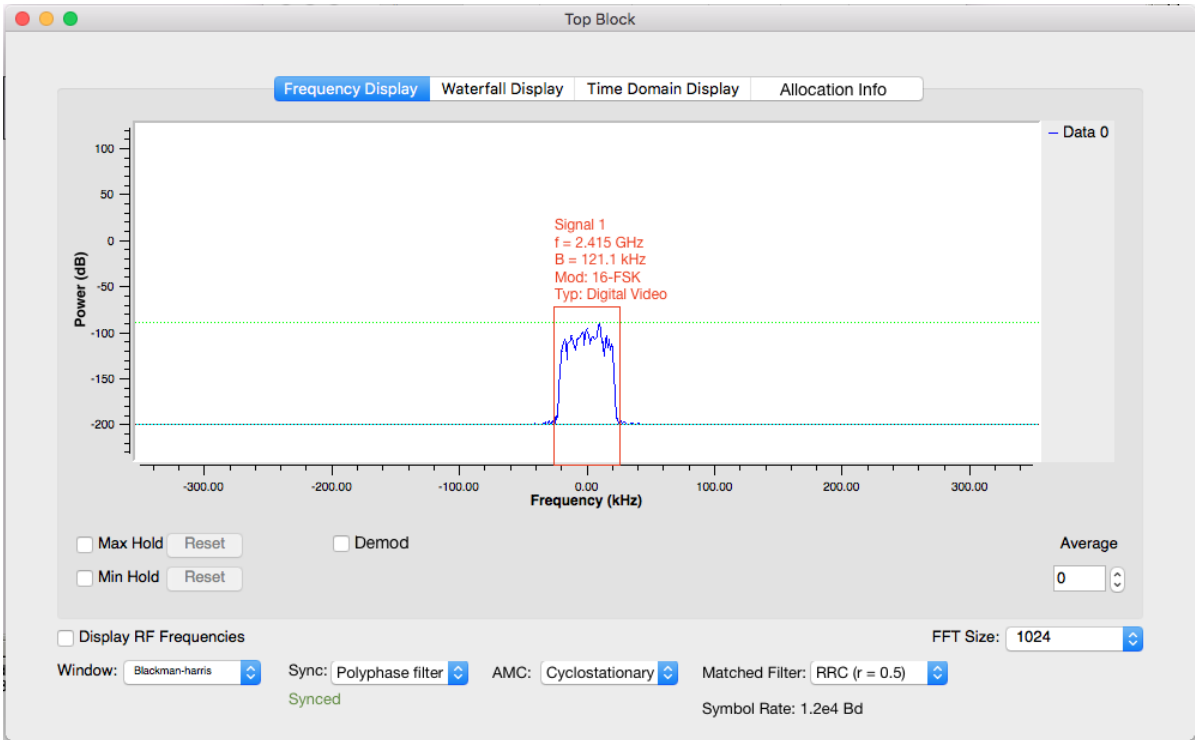
4.1 GSoC deliverables

Out-of-tree module `gr-sigint` Create new OOT module `gr-sigint` with `gr_modtool`. All of my work shall be part of this module. This module will be in the center of my participation in GNU Radio after the GSoC program. It shall be easily extendible and comprehensible.

Signal detection block When receiving a wideband signal, it is useful to automatically extract possible signals from the broadband spectrum. A block capable of detecting signals in the spectrum shall be developed. The implementation can be done by a simple energy detection followed by a RF map of the spectrum as described in [2]. Another approach is using a filter bank to split the input spectrum in sub channels. The sub channels can be examined with an energy detection and afterwards, the sub channels which contain a signal can be reconstructed with the work of [3].

Modulation classification block A seperate block shall be developed, capable of automatic modulation classification (AMC). Image comparison algorithms [4] can be applied to the spectrum, time signal, waterfall and constellation plots or analysis of cyclostationary features can be used for modulation detection. The results of [5] and [6] can be used and adapted here. As an additional target, machine or deep learning frameworks like *scikit-learn* or *Tensorflow* can be implemented [7]. The problem here for the typical user would be to generate enough training data for deep learning. Deploying a pre-trained version can be a solution. Approaches on this where made in [2]. The block shall be able to recognize FM, AM analog modulations as well as lower PSK, FSK and QAM. An easy extension of the modulation schemes will be possible.

Visualization Implementing a GUI will make the module very user friendly and easy accessible. The received spectrum and the detected signal shall be displayed. Also, information about the detected modulation will be visible. The tool `gqrx` [8] can serve as an example for this purpose. The GUI will be implemented in QT4/QT5 with `pyqtgraph`. A simple concept can be seen in Figure 1.



Average: 0

□ Display RF Frequencies

FFT Size: 1024

Window: Blackman-harris

Sync: Polyphase filter

Synced

AMC: Cyclostationary

Matched Filter: RRC (r = 0.5)

Symbol Rate: 1.2e4 Bd

Figure 1: Concept of GUI information (derivated from QT GUI Sink)

Documentation For an optimal collaboration, a good documentation of my work will be necessary. This will be realized by various source code comments and a distinct documentation in doxygen. Also, my work progress will be published on a blog with weekly updates. When necessary, video examples will be provided there. In addition, example flowgraphs with typical application cases will be integrated in the project.

4.2 Outlook

The mentioned deliverables only cover a part of the full signal intelligence workflow mentioned in the intoduction. In the future, solutions for further parts of this workflow can be developed.

A blind synchonization block would be needed in order to demodulate a given signal. This can be done using frequency synchonization with PLLs or Costas Loops and timing recovery with help of derivative matched filters or polyphase filter banks. Also, common preambles could be used to synchonize the signal.

An automatic demodulation block could be used afterwards. It can be implemented as kind of a demux component to select an existing demodulator from GNU Radio. The decision can be based on a message sent by the AMC block.

Further, a radio service database with known radio allocation data would be helpful.

With this, common signal properties can be guessed by using the allocation information of a signal. Also, it will give the user a nice overview of the spectrum around a signal. A basic block diagram of a signal intelligence workflow derivated from [2] can be seen in figure 2.

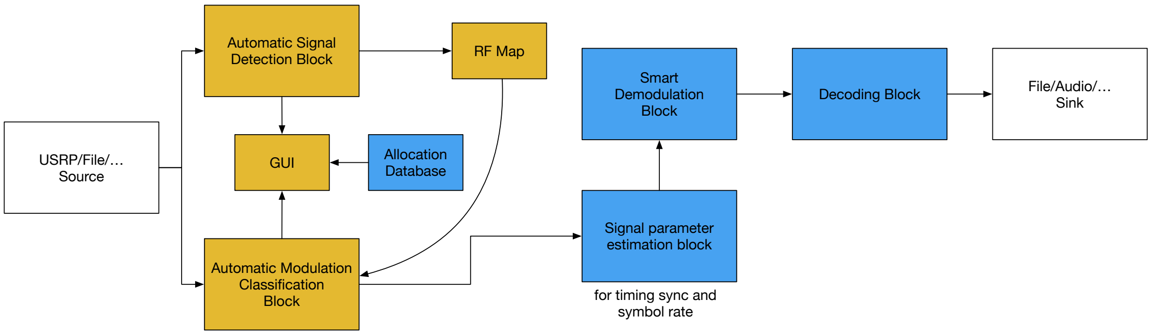

Automatic Signal Detection Block
RF Map
Smart Demodulation Block
Decoding Block
File/Audio/... Sink
USRP/File/... Source
GUI
Allocation Database
Signal parameter estimation block
Automatic Modulation Classification Block
for timing sync and symbol rate

Figure 2: Block diagram of signal intelligence workflow. The yellow blocks are planned to be examined/developed during the GSoC program, the blue blocks are a draft for the future development of this module.

5 Timeline

The timeline is orientated at the official GSoC timeline. As I have completed most of my studies with only one exam left, I can focus perfectly on my GSoC task. I have a side job, where I could take leave during the GSoC period. Be assured that the GSoC project will be my top priority during this summer.

Apr 22 - May 22 Get in touch with the GNU Radio community. Read the docs, install the software and play around to get used to the current version. Set up working environment and set up blog homepage. Do some literature research on my topic and begin to write first test code to evaluate possible approaches to the problem.

May 23 - Jun 02 Begin coding by creating new OOT module. Begin prototyping of signal detection block in Python. Try to detect signals in self-generated over-the-air samples and file sources. Access to hardware is provided, so over-the-air testing can be done.

Jun 03 - Jun 08 Implementation of previously evaluated signal detection algorithm in C++. Wrap algorithm in new GNU Radio block and write corresponding documentation.

Jun 09 - Jun 18 Start implementing GUI. Write own block to display results of signal detection block with use of `pyqtgraph`. Graphical feedback of signal detection will be implemented. Working GUI shall be availiable until midterms.

Jun 20 - Jul 16 Prototyping of modulation detection algorithms. Use different approaches: cyclostationary features, image comparison and deep learning algorithms. Test approaches with over-the air signals and train machine and deep learning algorithms.

Jul 17 - Jul 23 Implement results of modulation detection research in C++. Create a new GNU Radio block for modulation detection and write documentation.

Jul 24 - Jul 30 Expand GUI to display results of signal modulation detection block.

Aug 01 - Aug 04 Finish code documentation of created blocks and polish corresponding doxygen pages. Create example flowgraphs of typical applications with this module and create PyBOMBS recipe. Also use this time as buffer for remaining problems before pencils down deadline.

Aug 05 - Aug 14 Buffer zone. This time period allows to extend any previous tasks that needed longer than expected.

Aug 15 - Aug 23 Pencils down, provide version 1.0 of package via CGRAN. Submit the code samples to Google, revise documentation and cleanup code.

6 Conclusion

I think GNU Radio is the perfect project for communication engineers and people who want to get involved in radio technology to play around and build great applications. Signal intelligence is a very promising topic and there are endless possibilities that come to my mind for the post-GSoC future.

Participating in the GNU Radio project with the `gr-sigint` module would be a thrilling experience for me. I hope I successfully provided interesting ideas for the community and show why I think I am predestined to realize them. If you have any questions, please do not hesitate to contact me. And of course, Polar Codes are the coolest codes.

References

[1] https://github.com/kit-cel/gr-radar Visited March 5, 2016.

[2] O'shea, Timothy J., T. Charles Clancy, and Hani J. Ebeid."Practical signal detection and classification in gnu radio" SDR Forum Technical Conference (SDR). 2007.

[3] F. Harris, E. Venosa, Xiofei Chen and C. Dick,"Interleaving different bandwidth narrowband channels in perfect reconstruction cascade polyphase filter banks for efficient flexible variable bandwidth filters in wideband digital transceivers", Digital Signal Processing (DSP), 2015 IEEE International Conference on, Singapore, 2015, pp. 1111-1116.

[4] http://www.sigidwiki.com/wiki/Database#List_of_Known_Signals_in_Database Visited March 9, 2016.

[5] Mühlhaus, Michael Sebastian, "Automatische Modulationsartenerkennung in MIMO-Systemen", Dissetation, Karlsruhe, Karlsruher Institute of Technologie (KIT), 2014.

[6] K. Kim, I. A. Akbar, K. K. Bae, J. S. Um, C. M. Spooner and J. H. Reed, "Cyclostationary Approaches to Signal Detection and Classification in Cognitive Radio" New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on, Dublin, 2007, pp. 212-215.

[7] O'Shea, Timothy J., and Johnathan Corgan. "Convolutional Radio Modulation Recognition Networks" arXiv preprint arXiv:1602.04105 (2016).

[8] https://github.com/csete/gqrx Visited March 7, 2016.