Energy Detection Spectral Sensing With GNU Radio Companion

4503104

Bjarki Johannsson

b.johannsson@student.tudelft.nl

ET4394 Wireless Networking

Instructor: Przemyslaw Pawelczak

April 30, 2016

Contents

1 Introduction

Interest in cognitive radio has been growing rapidly in recent years. The increased demand for wireless services calls for better utilization of the spectrum available to today's technology, while research also looks at increasing the spectrum by going to higher frequencies.
One way to improve the spectral utilization is to identify and exploit empty spots in channels that have been assigned for use. Although a given frequency band has been allocated, it does not necessarily mean that the band is being used at all times. Using spectral sensing techniques one can scan for empty bandwidth which can be used for transmission. Care must be taken not to interfere with the band's primary user.
The most straight forward way to implement a spectral scanner is to detect the energy level of a given frequency. The noise floor can be estimated by measuring the energy where no signal is being transmitted. If a signal is transmitted, it results in a energy level above the noise level at the carrier's frequency which can be detected by using an energy threshold.
In this project such an energy based spectral sensor will be implemented using GNU Radio's GUI application called *GNU Radio Companion* [1]. A low priced RTL2832U DVB-T USB dongle with an accompanying antenna are used to detect the transmission from local DVB-T transmitters.

1.1 Local DVB-T Transmitters

Table 1 lists the location and transmitting frequencies of nearby DVB-T broadcasting towers [2].

Location	**Frequencies [MHz]**
Delft	498, 722, 762, 818
Den Haag	498, 722, 762, 818
Vlaardingen	474, 498, 530, 762, 818
Zoetermeer	498, 722, 762, 818
Rotterdam	474, 498, 522, 762, 818

Table 1: Location and transmitting frequencies of nearby DVT-B transmitters.

Table 1 indicates six transmitting frequencies of interest; 474, 498, 522, 530, 722, 762 and 818 MHz.

2 Setup

2.1 Software

GNU Radio Companion provides a very user friendly environment for implementing software versions of a wide variety of devices. The user can drag-and-drop ready made signal processing blocks to form a flow diagram. Block parameters can be set according to needs. In this project GNU Radio Companion version 3.7.2.1 was used on a 64-bit Ubuntu 14.04 machine. Additional package was installed for the RTL dongle.
Figure 1 shows the layout of the spectral sensor.

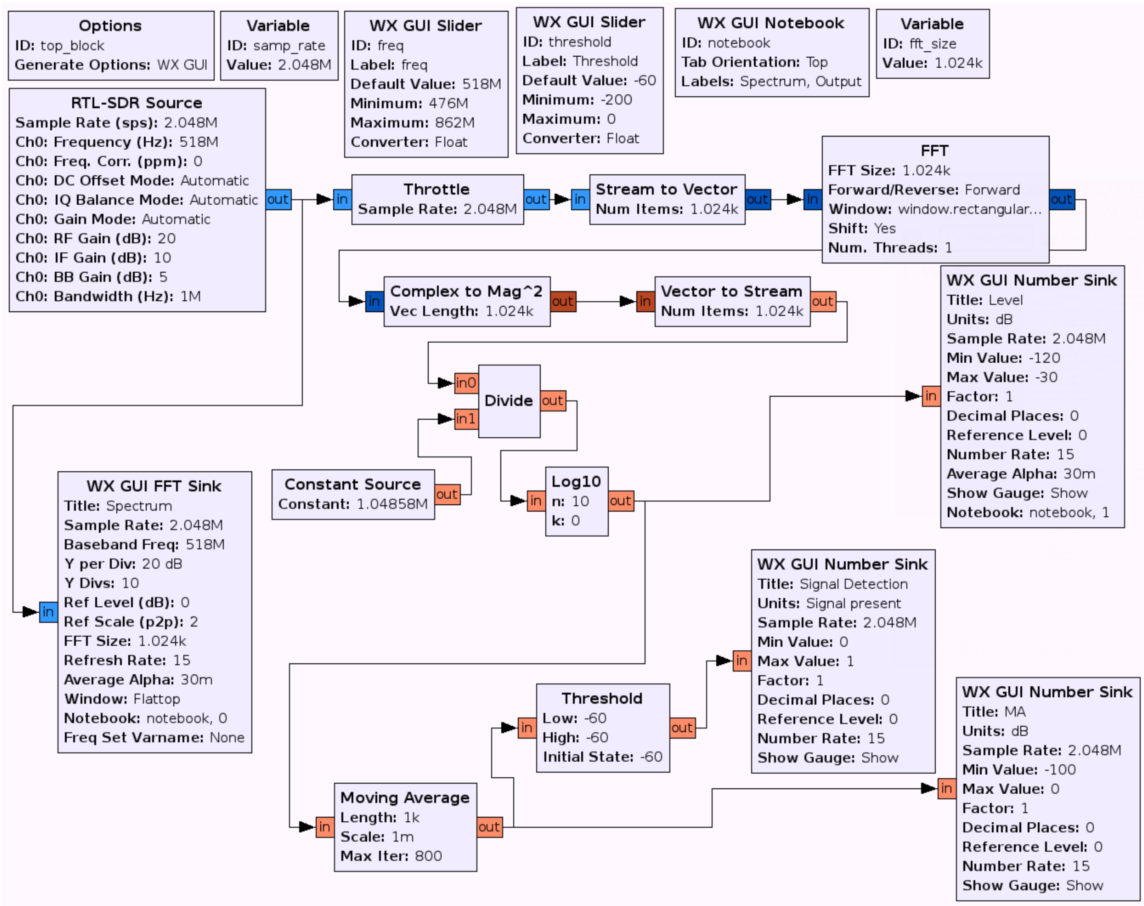

Figure 1: The GNURadio-Companion block diagram for the spectral energy detector.

Brief descriptions of the blocks are provided in Table 2. Full description of the blocks' signal processing functionality is outside the scope of this project.

Block	Function
RTL-SDR Source	Gathers the signal
WX GUI FFT Sink	Displays a FFT plot of the input
Throttle	Safety valve for sample throughput
Stream to Vector	Gather samples into vector for FFT
FFT	FFT transformation
Complex to Mag	Squares to get magnitude
Vector to Stream	Convert back to stream
Divide	Divides stream by Constant Source value
Log10	Convert values to dB
WX GUI Number Sink	Displays value
Moving Average	Takes the moving average of stream
Threshold	Outputs 1 if input exceeds threshold, 0 otherwise

Table 2: Description of the GNU Radio Companion blocks.

2.2 Hardware

After installing the drivers for the dongle[1], it is ready to use with GNU Radio.

3 Measurements

Measurements were taken in three location around Delft; in Poptahof, the TUD library and in Icarusweg. Noise measurements were performed in arbitrary empty frequencies. Tables 3 a, b and c show the measured values for each location. An unknown signal was detected by chance at 698 MHz, and was added to the frequencies mentioned in section 1.1.

[1]The driver installation is platform specific and instructions can easily be found by searching the internet.

Table 3: Signal and noise measurements.

(a) Location Poptahof.

Signal		
Frequency [MHz]	**Level [dB]**	**Detected**
474	-66.6	X
498	-58.6	✓
522	-53.1	✓
530	-66.8	X
698	-60.5	✓
722	-63.1	X
762	-64.1	X
818	-66.7	X

Noise	
Frequency [MHz]	**Level [dB]**
487	-66.8
506	-66.6
533	-66.9
592	-66.7
670	-67.0
740	-67.5
830	-67.1
862	-67.9

(b) Location TUD library.

Signal		
Frequency [MHz]	**Level [dB]**	**Detected**
474	-64.7	X
498	-52.7	✓
522	-53.8	✓
530	-66.7	X
698	-56.8	✓
722	-59.5	✓
762	-60.4	✓
818	-63.1	X

Noise	
Frequency [MHz]	**Level [dB]**
487	-66.4
506	-66.9
533	-67.1
592	-66.6
670	-67.2
740	-67.3
830	-67.3
862	-67.0

(c) Location Icarusweg.

Signal		
Frequency [MHz]	**Level [dB]**	**Detected**
474	-66.4	X
498	-52.6	✓
522	-52.9	✓
530	-67.0	X
698	-50.8	✓
722	-61.8	✓
762	-64.4	X
818	-66.0	X

Noise	
Frequency [MHz]	**Level [dB]**
487	-66.9
506	-66.3
533	-67.0
592	-66.5
670	-67.0
740	-68.1
830	-67.4
862	-67.2

4 Results

Figure 2 shows the measured values at each location for easy comparison.

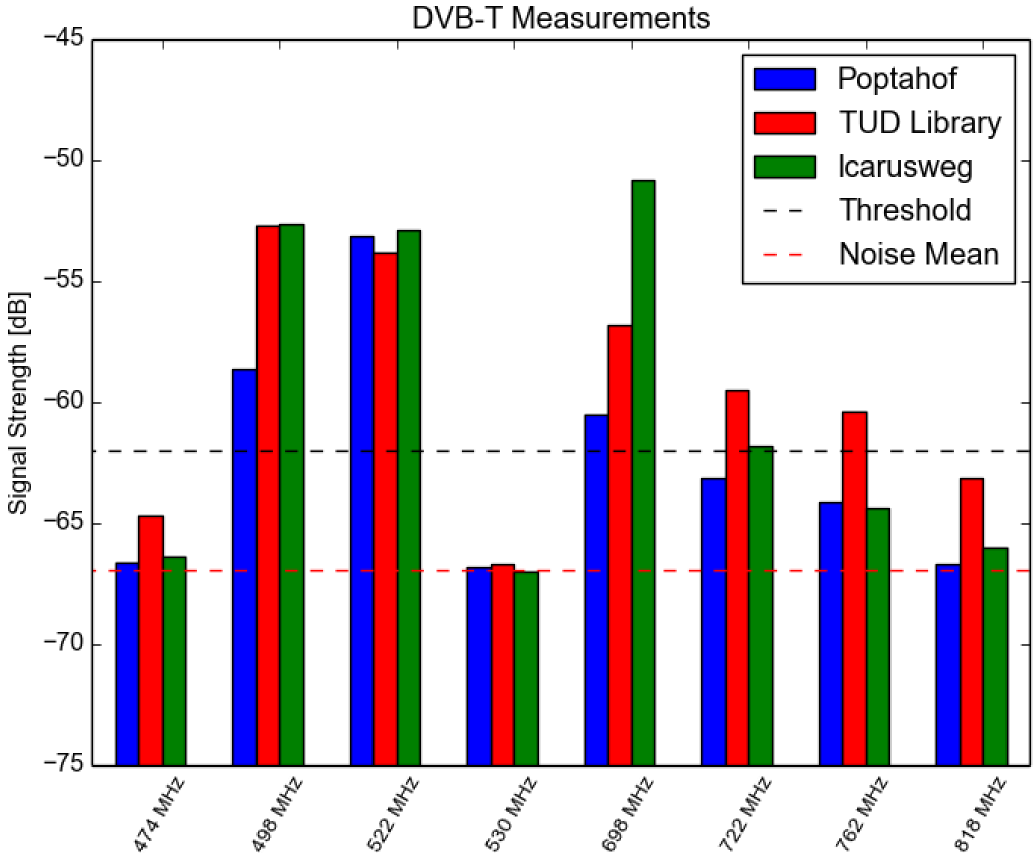

Figure 2: Measured signal strength.

It appears that reception is better in the library than the other two locations. This could perhaps be explained by the fact that the apartments used for measuring in Poptahof and Icarusweg are in residential neighborhoods, surrounded by tall buildings, while the immediate surroundings of the library are relatively clear. The tall buildings can cause shadow fading of the signal, when interfering with the line of sight from the transmitter.
Since there is no indication of a signal at 530 Mhz, these measurements were left out in the proceeding calculations.

Figure 3 shows the PDFs for the signal and noise levels.

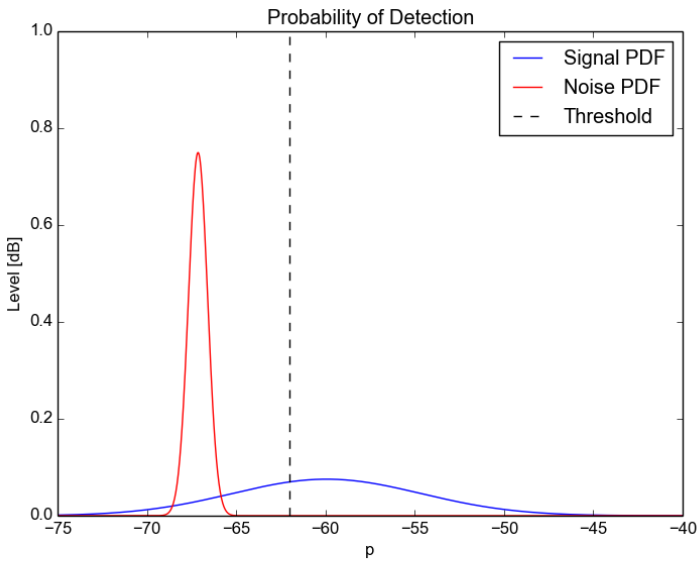

Figure 3: Probability distribution functions for the signal and noise levels.

The low distribution of the noise level came as somewhat of surprise. It is clear that the threshold can be lowered by a few decibels without significantly increasing the probability of a false alarm. At the same time this would increase the probability of detecting the transmitting signal, lowering the chance of interfering with primary users.
Figure 4 shows the ROC curve for the receiver.

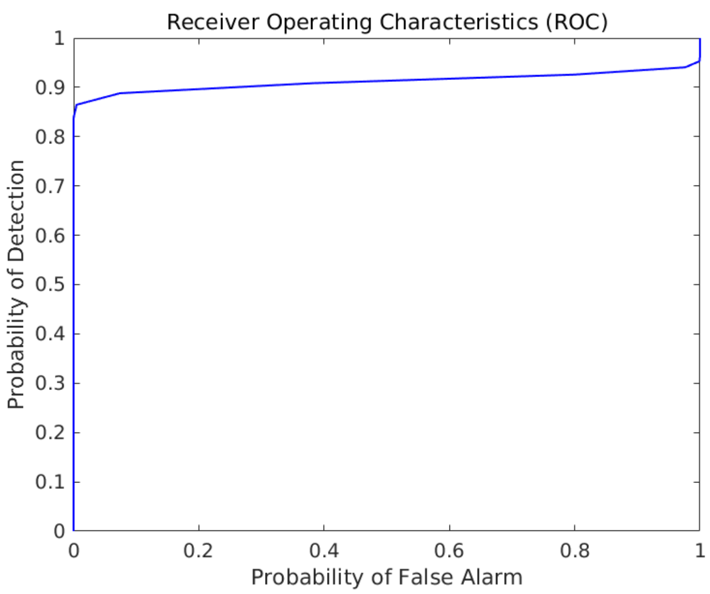

Figure 4: The ROC curve shows that there is quite some room for improvement.

Calculating the probability of detection yielded a somewhat low value, or $P_{detection} = 0.779$. The probability of a false alarm however was calculated as $P_{false\,alarm} = 1.75 * 10^{-9}$. For a real-life personal cognitive radio it is the author's opinion that the probability of detection should be increased at the cost of higher probability of false alarm. This would result in less probability of interfering with primary users while also decreasing the opportunities to use an empty channel.

5 Conclusions

Perhaps the threshold should be lowered to allow for higher probability of detection. A value of roughly -65 dB would have yielded four more detections among the three locations, but consequently this also increases the probability of a false alarm. This tradeoff needs to be evaluated when using cognitive radio. Having too low probability of detection causes interference with other users of the channel, possibly resulting in an unusable channel for both primary and secondary users.

This project has provided valuable insight into spectrum sensing for cognitive radio. The GUI is relatively easy to use although the knowledge must be there to use it properly. Some effort was spent trying to add new blocks, but to little avail since the steps provided in the tutorial did not work and the author's knowledge of linking libraries and modules are limited.

References

[1] GNU Radio web page, http://gnuradio.org/, accessed on 26.04.2016

[2] SATBroadcasts web page, http://www.satbroadcasts.com/DVB-T_Radar_results.html, accessed on 26.04.2016