GNU Radio Project Report

ET4394 Wireless Networking

Andri Rahmadhani - 4505611

Technische Universiteit Delft

Delft University of Technology

Challenge the future

Contents

1
Introduction

1.1. Cognitive Radio and Signal Detector

Cognitive radio is a radio that can be programmed and configured dynamically to run on an empty channel by automatically detecting available channels or white space in a wireless frequency spectrum. Dynamic Spectrum Access (DSA) is one of the most popular application of cognitive radio where the radio can detect the presence of primary user's frequency range and avoid interfering with it. In DSA, the radio monitors the activity of radio-frequency spectrum and use the available channels before start transmitting [1]. Therefore, the effectiveness of cognitive radio relies on signal detector component.

In this project, the signal detector is build using a portable Software Defined Radio (SDR) device that can be easily configured and programmed using GNURadio. Inside GNURadio, we can design functional block diagrams along with their parameters that can be interconnected each other to build a fully working signal detector. We introduce threshold level of signal detection as a parameter to evaluate the correctness of the detector. The detector has four possible outputs that are so-called Receiver Operating Characteristic (ROC), namely: True Positive (TP) or correct detecting existing signal, False Positive (FP) or false detection, False Negative (FN) or misdetection, and True Negative (TN) or correct detecting non-existing signal. The effective detector should have a high probability of either TP or TN and a low probability of FP and FN.

1.2. DVB-T Signal

Digital Video Broadcasting - Terrestrial (DVB-T) is a standard used in television broadcasting system mostly in Europe to transmit digital video [2]. It uses Orthogonal Frequency Division Multiplex (OFDM) transmission schemes to transmit several video and audio broadcast, which are multiplexed into a single carrier signal. DVB-T works on frequency range from 478 MHz to 862 MHz and employs several modulation systems, such as QPSK, 16QAM, and 64QAM [3]. In Netherlands, especially in Delft, there are several DVB-T Multiplexer (MUX) operators that are currently operating as seen in Table 1.1. Note that all operators in Table 1.1 have the same 8 MHz bandwidth [4].

Table 1.1: Frequency and Channel of DVB-T MUX operators in Delft

DVB-T MUX Operator	UHF Channel	Center Frequency (MHz)
RTS Bouquet 1	52	722
NTS1 Bouquet 2	49	698
NTS2 Bouquet 3	57	762
NTS3 Bouquet 4	24	498
NTS4 Bouquet 5	27	522

In this project, the signal in Table 1.1 should be detected by our implemented signal detector. If all the signals are detected correctly, it indicates that the system run as secondary user can safely work on unused frequencies within DVB-T spectrum range. Note that not all of the DVB-T channels are occupied the entire DVB-T spectrum range. Most of them are unused in certain area. This becomes the reason why DVB-T spectrum range is used frequently in cognitive radio applications.

2
PROJECT DESCRIPTION

2.1. OBJECTIVE

The objective of this project is to design and implement a simple, yet effective digital television (DVB-T) signal detector using GNURadio. The detector output can be used to obtain empty channels available within DVB-T frequency band. In order to get correct and reliable results, the detector is tested and characterized under varying channel condition and threshold, e.g. measure on different places with different noise level. From this, we can evaluate whether the detector is able to capture the correct signal or just throwing a false detection. Therefore, we have to improve the mechanism of the detector to increase the performance of detection.

2.2. HYPOTHESIS

The decision whether the signal is detected or not depends on the threshold value. The probability of false detection in which the detector gives positive detection when there is no real signal appeared will be decreased if we increase the threshold value. However, the probability of misdetection will increase as the low signal, which has a signal level around noise or reference level, will not be detected. On the contrary, if we decrease the threshold value, the probability of false detection will be increase and the probability of misdetection will be decreased as the noise slightly above threshold value could also be detected as a signal.

3
Design and Implementation

3.1. Design Approach

There are several methods that can be used to detect a signal as listed in [3]. One of the methods is energy detection, which is the most popular method due to its simplicity. Energy detection works by considering the power level of the signal captured by receive. If the power level drops below a certain threshold value, the detector will not detect the signal. Otherwise, the signal will be categorized as detected signal. The power level of the signal is obtained from FFT, which then squared and averaged to get the magnitude. This value will be compared with the threshold value. Note that in order to get a stable signal, the detector is equipped with moving average filter after getting the magnitude. In addition, band-pass filter is used right before FFT calculation to reduce noise. The block diagram of the proposed detector can be seen in Figure 3.1

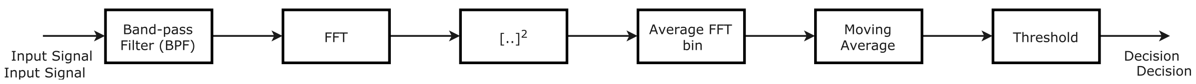


Figure 3.1: Proposed energy detector block diagram

From [4], a DVB-T carrier frequency has 8 MHz bandwidth, which can contain up to 12 channels depending on the modulation and also the quality of the channels. As we do not consider modulation and demodulation method for this project, the decision process works by only considering the bandwidth. For instance, if the detector detects a signal beyond ± 8 MHz, it will not be categorized as detected DVB-T signal. Using such approach gives a trade-off between misdetection and false alarm. Most likely, it reduces the probability of false alarm but may lead to the increased probability of misdetection. However, the main advantage is that the detector supposedly can distinguish between DVB-T signal and other signals.

3.2. Implementation in GNU Radio

In this project, the detector is built using SDR dongle, Realtek RTL2832U, which is used as a receiver. The dongle is capable of capturing FM, DVB-T, and DAB signal. It has a built-in band-pass filter to eliminate undesired frequencies. The signal that has been filtered will be converted to digital data using A/D converter and ready to be processed afterwards. We use GNU Radio Companion to process the captured signal. GNU Radio Companion allows us to easily build functional block diagrams but lacks of features for testing or building automated test scenario. Luckily, GNU Radio Companion generates a Python script, which can be run through GNU Radio Companion GUI or terminal just like running ordinary Python script. The script can be modified or imported to a new python script in which the automation functions reside. We prefer to create a new python script, called 'main' function because the generated script will be replaced once the block diagram is modified [5–7]. The block diagram of the proposed detector created using GNU Radio Companion can be seen in Figure 3.2 along with its description in Table 3.1.

The block diagram is built based on the workflow of the detector as seen in Figure 3.3. First, the main python script is executed. The script has a feature to let user inputs command parameters. Then, the GUI generated by GNU Radio Companion will appear and the automated measurement start running. The script

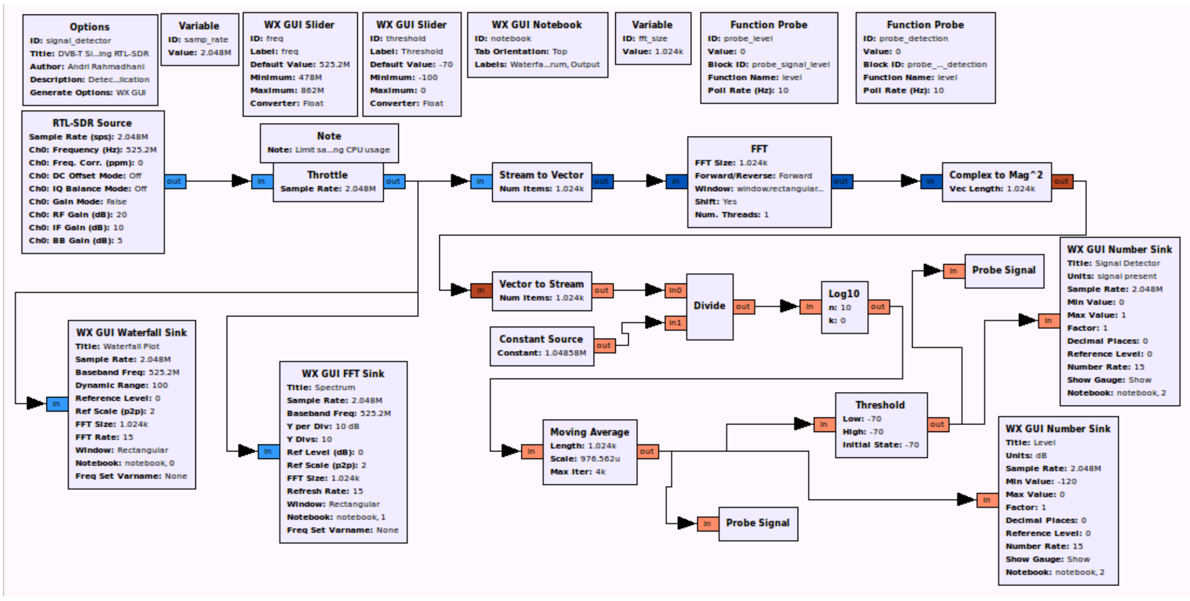


Figure 3.2: Diagram block of proposed detector in GNU Radio Companion

will run automatic noise level finder to obtain and set appropriate threshold if it is specified in input parameters. Subsequently, the script will incrementally scan frequencies in the given range, or the given list read from a file depending on the input parameters. In the scanning process, if the power level is higher than the threshold, the detection status is set to true (1) otherwise is set to false (0). The results are saved to files for further processing.

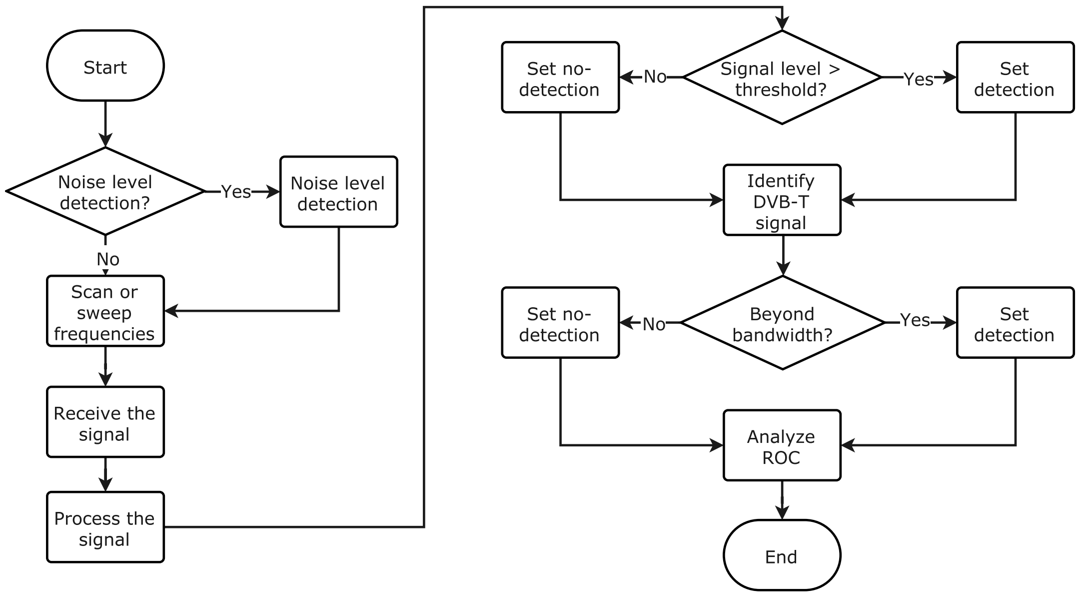


Figure 3.3: Workflow of proposed detector

In order to improve the detection results, we introduce DVB-T signal identifier to distinguish between DVB-T and other signals based on bandwidth. It counts the number of consecutive detections in a certain range of frequency and tries to find the center frequency by taking the median. If the detection is far beyond the bandwidth, the detector decides not to raise the detection flag. Otherwise, the detector set the detection flag to true (1), indicating the presence of DVB-T signal. The results will be used to analyze Receiver Operating Characteristic (ROC), which can be done automatically using the Python script [8–10].

Table 3.1: GNU Radio block diagram description

Block Name	Description	Parameter
RTL-SDR Source	Getting the input of signal from RTL2838 dongle	Sample Rate : 2.048 M Samples/s RF Gain : 20 dB IF Gain : 10 dB Baseband Gain : 5 dB BW : 1 MHz
Throttle	Limiting the sample per sec (sampling rate)	Sample Rate : 2.048 M Samples/s
Stream to Vector	Converting stream data into vector in order to be processed in FFT	Num Items : fft_size (1024)
Complex to Mag^2	Calculating magnitude squared value of FFT sample output	
Vector to Stream	Converting Vector to Stream after FFT process. (The opposite of Stream to Vector block)	Num Items : fft_size (1024)
Constant Source	Generating (FFT size) 2 value as a divisor	(fft_size) 2
Divide	Getting the average bins of FFT Result (Complex to Mag^2) by (FFT size) 2 in order to get the power	
Log10	Converting the power level into dB P (dB) = 10*log(P [Watt])	
WX GUI FFT Sink (Spectrum)	Displaying FFT results of spectrum	Sample Rate : 2.048 M Samples/s Baseband freq : freq FFT Size : fft_size (1024) Window : Rectangular
WX GUI Waterfall Sink	Displaying waterfall spectrogram	Sample Rate : 2.048 M Samples/s Baseband freq : freq FFT Size : fft_size (1024) Window : Rectangular
WX GUI Number Sink (Level)	Displaying the level value in dB	Sample Rate : 2.048 M Samples/s Min value : -120 Max value : 0 Average Alpha : 0.03
Threshold	Setting the threshold with upper limit -60 dB	threshold
WX GUI Number Sink (Signal Detection)	Displaying the signal detection based on threshold set. "1" if there is a signal present and "0" if there is no signal present	Sample Rate : 2.048 M Samples/s Min value : 0 Max value : 1
Variable Sampling Rate	Define the sampling rate	samp_rate : 2.048 M Sample/s
Variable FFT Size	Define the FFT Size	fft_size : 1024
WX GUI Slider Frequency	Define the frequency	freq freq. min : 478 MHz req. max : 862 MHz
WX GUI Slider Threshold	Define the threshold. It will give "1" if the signal level is above -60 dB and will give "0" if the signal level is below -60 dB.	threshold default = -60 dB min : -100 dB max : 0 dB
Probe Signal	Probe output of a block	ID : probe_signal_level
Probe Signal	Probe output of a block	ID : probe_signal_detection
Function Probe	Allow a probe signal to be accessible in Python script. It can be used to get output value of a block in which the probe signal is attached	ID : probe_level Block ID : probe_signal_level Function Name : level
Function Probe	Allow a probe signal to be accessible in Python script. It can be used to get output value of a block in which the probe signal is attached	ID : probe_detection Block ID : probe_signal_detection Function Name : level
Moving Average	Applying Moving Average to stabilized data	Length : fft_size Scale: 1.0/fft_size

4

Results & Analysis

4.1. Measurement Results

In order to obtain variation of channel conditions, measurement of DVB-T signal was taken in two different places, namely TU Delft Library and Poptahof Noord. Some parameters like power level, spectrum, and decision were taken during the measurement. For a starting point, we let the detector automatically choose appropriate threshold value above the average noise level, which is be done by quick scanning a given range of frequencies. Then, the signal scanning process begins from the minimum frequency until the maximum frequency with step value 1 MHz. As the number of measurements is large enough, only several results are shown in Table 4.1.

Table 4.1: Signal measurement results

User	**Center Frequency (MHz)**	**TU Delft Library (6th floor) (52.002947, 4.375374)**				**Poptahof Noord (4th floor) (51.999881, 4.351265)**			
		Level (dB)	**Detected (Th = -72.071 dB)**	**Average (dB)**	**Std. Deviation**	**Level (dB)**	**Detected (Th = -73.5985)**	**Average (dB)**	**Std. Deviation**
NTS3 Bouquet 4	498	-52.220	Yes			-48.940	Yes		
NTS4 Bouquet 5	522	-50.124	Yes			-43.887	Yes		
NTS1 Bouquet 2	698	-55.013	Yes	-52.700	4.038	-54.064	Yes	-51.733	4.789
RTS Bouquet 1	722	-52.007	Yes			-54.250	Yes		
NTS2 Bouquet 3	762	-61.558	Yes			-57.526	Yes		
Empty	586	-77.237	No			-78.107	No		
Empty	607	-77.594	No			-78.204	No		
Empty	621	-76.593	No			-78.247	No		
Empty	633	-78.085	No			-78.529	No		
Empty	642	-77.497	No			-78.427	No		
Empty	673	-77.820	No			-77.587	No		
Empty	730	-77.924	No	-75.109	6.671	-78.401	No	-76.218	6.679
Empty	782	-77.372	No			-78.470	No		
Empty	792	-47.862	No			-55.602	No		
Empty	813	-56.287	No			-57.373	No		
Empty	841	-77.951	No			-78.536	No		
...	...	...	...			...	...		

From Table 4.1, there are 5 registered DVB-T frequencies captured by the detector [4]. The rest are categorized as empty channels though they could be occupied by other radio systems. We obtain the mean and standard deviation of the level from both detected signals and empty channels to analyze receiver performance (ROC). Note that we set 5 seconds as the duration of measurement per frequency to ensure the stability of the detector, reducing error of measurement.

4.2. Signal Detection

A signal can be detected when its power level is above a threshold value as seen in Figure 4.1. In this project, we expand the mechanism of the signal detection by comparing the number of detection in a certain range of frequency with DVB-T bandwidth. In this case, the bandwidth is 8 MHz. Figure 4.2 depicts the edge detection of a DVB-T spectrum that has center frequency of 498 MHz (NTS3 Bouquet 4). One of the edges lies on 494 MHz and another one on 502 MHz, which means that the bandwidth of the detected signal is about 8 MHz and thus can be categorized as DVB-T signal.

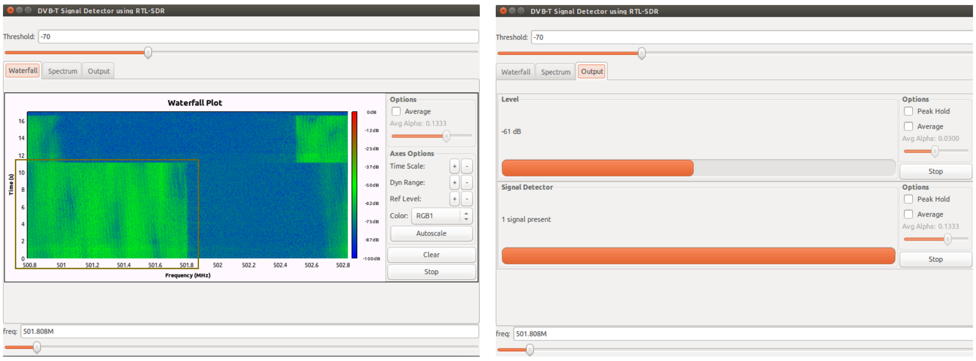


Figure 4.1: Waterfall plot and signal detection status

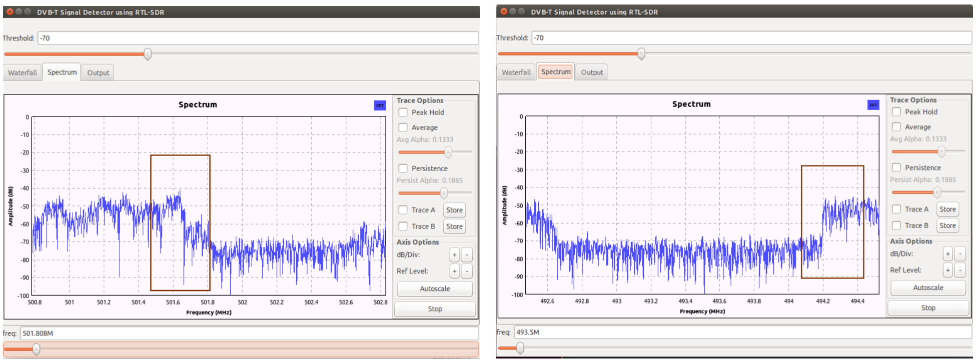


Figure 4.2: Edge detection of detected DVB-T signal which has center frequency of 498 MHz

By using automated measurement tool that is built with Python script, we are able to plot the measurement of signal level over the whole range of DVB-T frequency spectrum. It can also identify and recognize DVB-T signal based on bandwidth. Figure 4.3 shows the spectrum of the signals measured on the given range of frequency, and the recognized DVB-T center frequency with threshold -60 dB captured at TU Delft Library. There are some of signals that could not be detected as they below the threshold value, e.g. the signal at frequency 480 MHz, 550 MHz, and 760 MHz. Therefore, it may lead to the increased probability of misdetection of a signal.

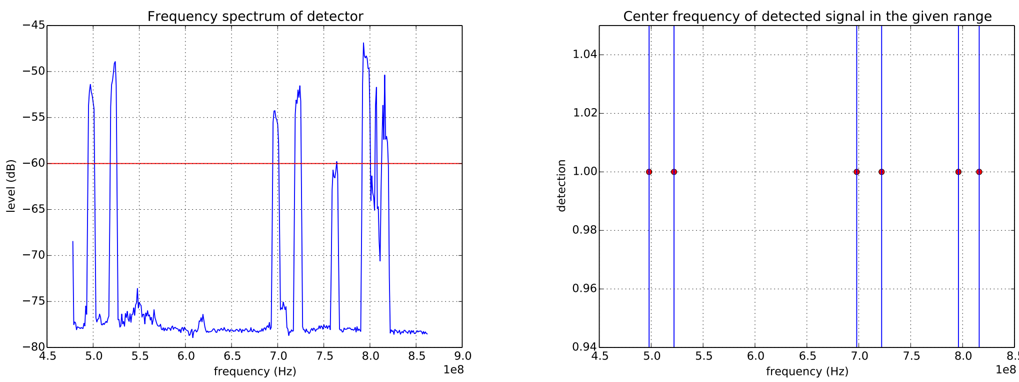


Figure 4.3: Frequency spectrum of signal with threshold -60 dB captured at TU Delft Library

One way to reduce the probability of misdetection is by lowering the threshold value. Figure 4.4 shows the frequency spectrum and the detected center frequency of DVB-T signal when the threshold value is lowered to -72.0716 dB by using automatic threshold finder. The signal at 480 MHz is still not detected as DVB-T signal as the bandwidth is not around 8 MHz. This also happens to the signal captured at the range of 790 - 820 MHz in which the detected bandwidth is larger than 8 MHz. Therefore, the detector decides not to include such kind of signals in its detection list.

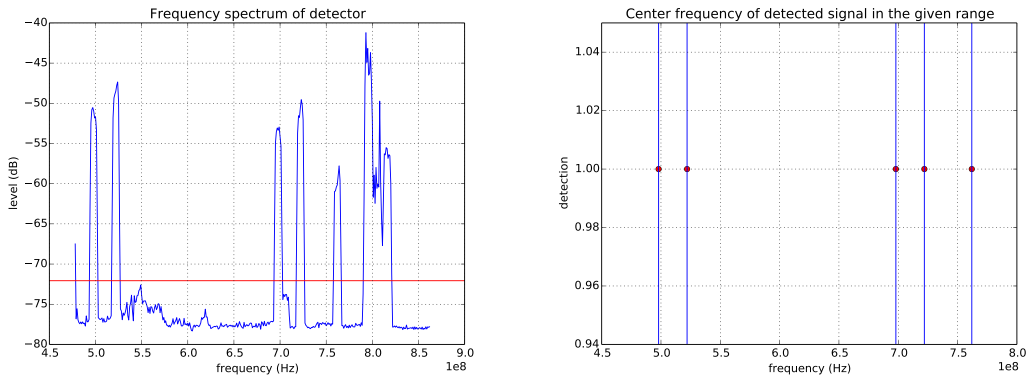


Figure 4.4: Frequency spectrum of signal with automactic threshold finder (-72.0716 dB) captured at TU Delft Library

The 480 MHz and 600 MHz signals have relatively high receive level with a bandwidth around 200 kHz. From the Nederland Frequency Database [11], those frequencies supposed to be empty. It could be caused by unregistered signal or probably the intermodulation product of other signals. Another interesting point is that the signal in the range of 790 - 820 MHz now become undetected though the level of the signal is sufficiently large. According to the same reference [11], this range is used for Wideband Frequency Modulation (WFM) radio. However, given the fact that the detector could detect these signals with sufficient threshold value, the final decision made by the detector will not be changed, again, because the bandwidth is far beyond 8 MHz. Moreover, in this project we are only interested in finding DVB-T signal spectrum. Thus, it should not be a problem to have misdetection on non-DVB-T signals.

The frequency spectrum of the second measurement done on different place, i.e. in Poptahof Noord, with threshold -60 dB is depicted in Figure 4.5. The signal spectrum is rather distinct from the frequency spectrum measured at TU Delft Library. Signal around 720 MHz has a stronger level but is still not considered as a DVB-T signal. Signal around 790 - 820 Mhz is detected as a single signal tough it seems like two signal appeared. However, this signal should not be considered as a DVB-T signal because it is a WFM radio signal. This indicates the false alarm generated by the detector.

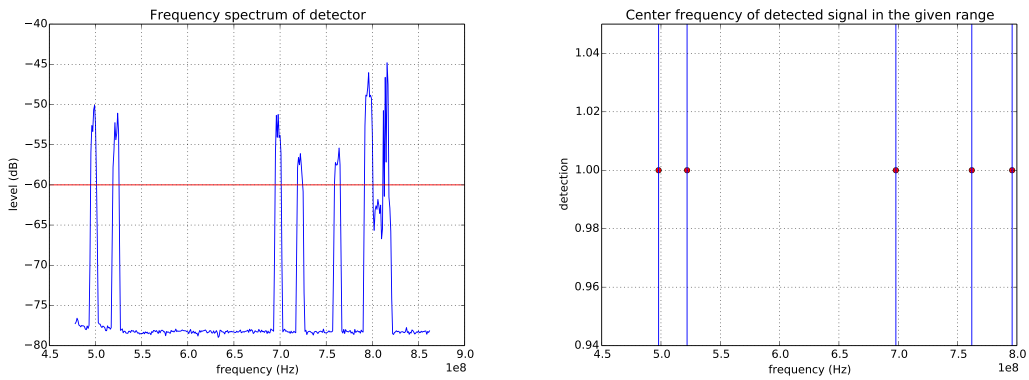


Figure 4.5: Frequency spectrum of signal with threshold -60 dB captured at Poptahof Noord

The automatic threshold finder has been used to get the lower threshold value. The results can be seen in Figure 4.6. The difference is that the false alarm within the frequency range of 790 - 820 MHz has gone. The 722 MHz frequency (RTS Bouquet 1) could be detected properly. The result is the same as the previous result, which is measured at TU Delft Library. In other words, the conditions of the channels could affect the detection depending on the threshold value being used.

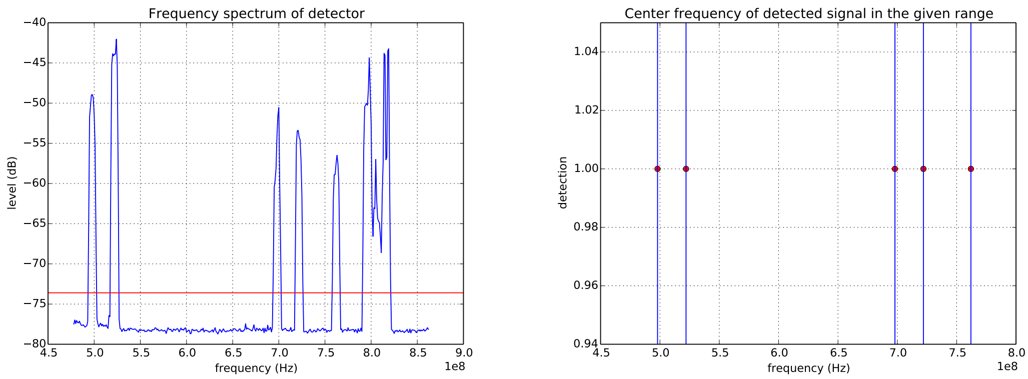


Figure 4.6: Frequency spectrum of signal with automactic threshold finder (-73.5985 dB) captured at Poptahof Noord

4.3. SIGNAL MISDETECTION

Signal misdetection occurs whenever the detector could not be able to detect the presence of the signal due to the lack of the power level and lies below the threshold, or the bandwidth is beyond a certain value. For instance, Figure 4.7 shows the unstable measurement at both edges of signal that has a center frequency of 722 MHz. Slightly changing the threshold value may lead to misdetection of the signal, see Figure 4.5 and Figure 4.6 for comparison. To tackle this problem, we can decrease the threshold. Note that lowering the threshold could trigger a false alarm. Thus, the appropriate threshold value should be selected carefully.

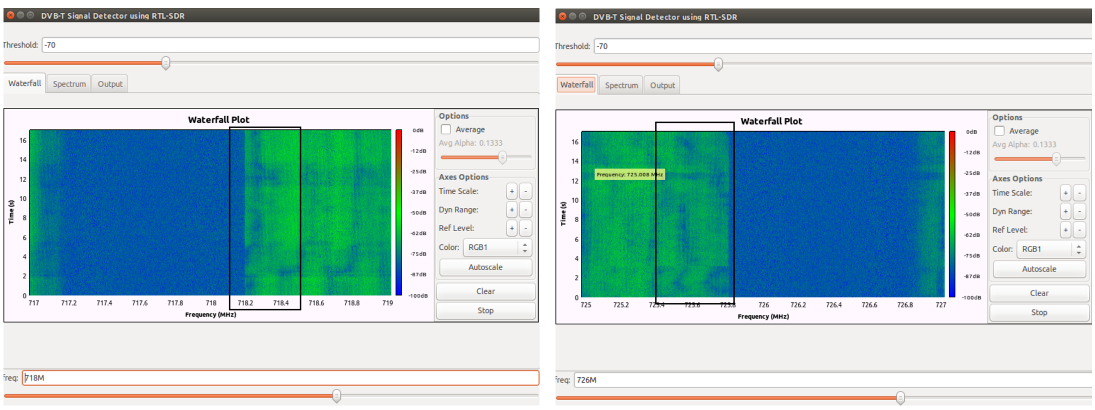


Figure 4.7: Waterfall plot of misdetected signal with center frequency 722 MHz and threshold -60 dB

4.4. NO SIGNAL PRESENCE

The detector is said to detect an empty channel when it only senses a noise level. To detect average noise level, we can take several samples of measurement and calculate the distribution, typically in a form of normal distribution. Figure 4.9 shows the detection of an empty channel at the center frequency of 600 MHz. The blue color indicates low power level measurement or can be said as noise that lies around -78 dB.

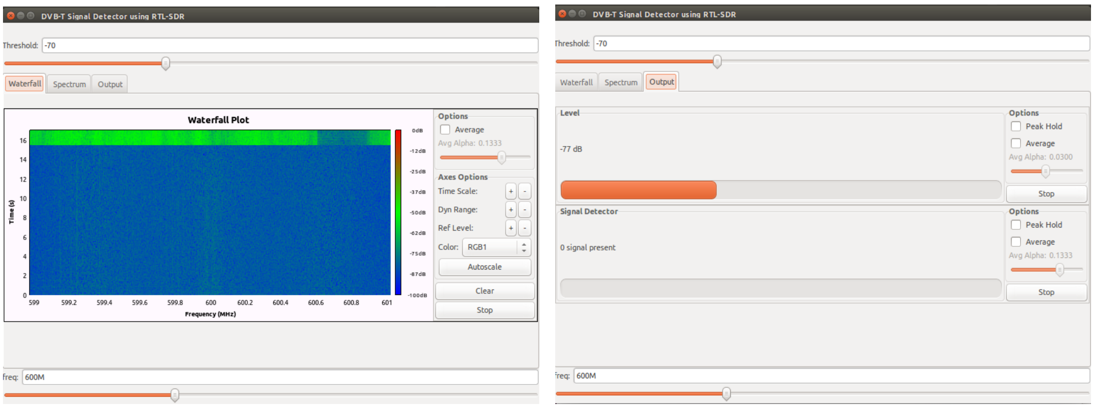


Figure 4.8: Waterfall plot of an empty channel with center frequency 600 MHz

4.5. FALSE ALARM

A false alarm occurs whenever the detector sense the presence of a signal though it supposed no to do so. The existence of carrier frequencies other than DVB-T can be one of the cause of false alarm. Moreover, the threshold value that is set nearly on the average of noise level may trigger false alarm once the noise level fluctuates and exceeds the threshold. One of the solutions is by increasing the threshold value. Figure X depicts the example of false alarm occurred at 790 - 820 MHz due to the exceeded number of bandwidth, i.e. non-DVB-T signal. See Figure 4.3 and Figure 4.4 for comparison.

Figure 4.9: Waterfall plot of signal on frequency range 780 - 820 MHz

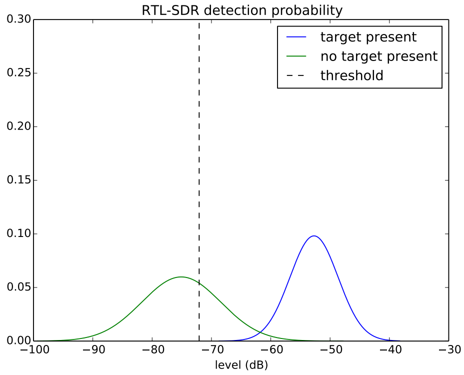
In this project, as we use a moving average block in GNU Radio Companion, the number of fluctuation of the detection is reducing a lot. Thus, it is probably hard to find a false alarm that comes from fluctuation of the noise level.

4.6. RECEIVER PERFORMANCE

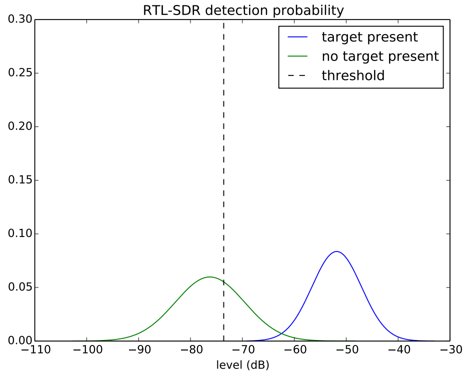
To analyze the receiver performance characteristic (ROC), first, we have created two probability density functions (PDF) that are combined into a single plot as seen in figure Figure 4.10, one from the measurement of detected signal, and one from the measurement of no empty channel. The PDF functions are generated based on the mean and the standard deviation of both type of measurements, see Table 4.1. From here, we can obtain the probability of detection and the probability of false alarm by taking the complement of cumulative distribution function (CDF) for each type of measurements. We have created a Python script to perform the ROC analysis easily. The result can be seen in Table 4.2.

Table 4.2: Signal measurement results

TU Delft Library (6th floor) (52.002947, 4.375374)			Poptahof Noord (4th floor) (51.999881, 4.351265)		
Threshold (dB)	**Probability of Detection (Pd)**	**Probability of False Alarm (Pfa)**	**Threshold (dB)**	**Probability of Detection (Pd)**	**Probability of False Alarm (Pfa)**
-72.0716	0.9999586474	0.3234032337	-73.5985	0.9998543512	0.3463925472
-62	0.9897364655	0.0264029694	-62	0.9835587505	0.0157515685



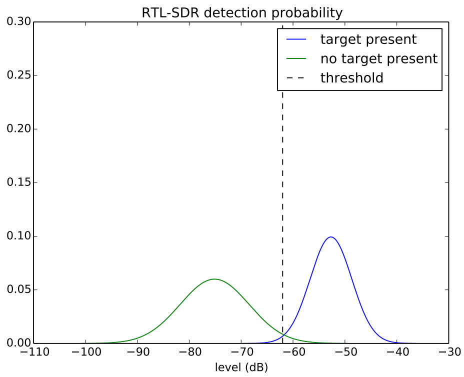
(a) TU Delft Library (threshold = -72.0716 dB)



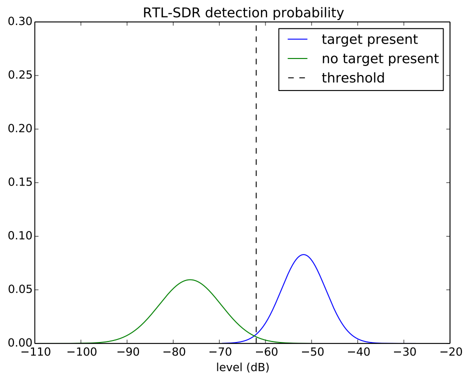
(b) Poptahof Noord (threshold = -73.5985 dB)

Figure 4.10: Receiver performance measured at two different places

From Figure 4.10 and Table 4.2, despite that the probability of detection is sufficiently high, the probability of false alarm in two different places of measurements is still high. To get the optimal results, we can increase the threshold value to the point of intersection between the distribution of no target present and the distribution of target present. For both places, we obtain around -62 dB as the optimal threshold. Figure 4.11 depicts the optimized threshold value in the PDFs.



(a) TU Delft Library (threshold = -62 dB)



(b) Poptahof Noord (threshold = -62 dB)

Figure 4.11: Optimal threshold value of receiver performance measured at two different places

Based on the PDF of detection and false alarm, we can obtain ROC curve with probability of false alarm (Pfa) on the horizontal axis and probability of detection (Pd) on the vertical axis. The ROC curve is depicted in Figure 4.12. Note that the working area of the receiver lies below the ROC curve.

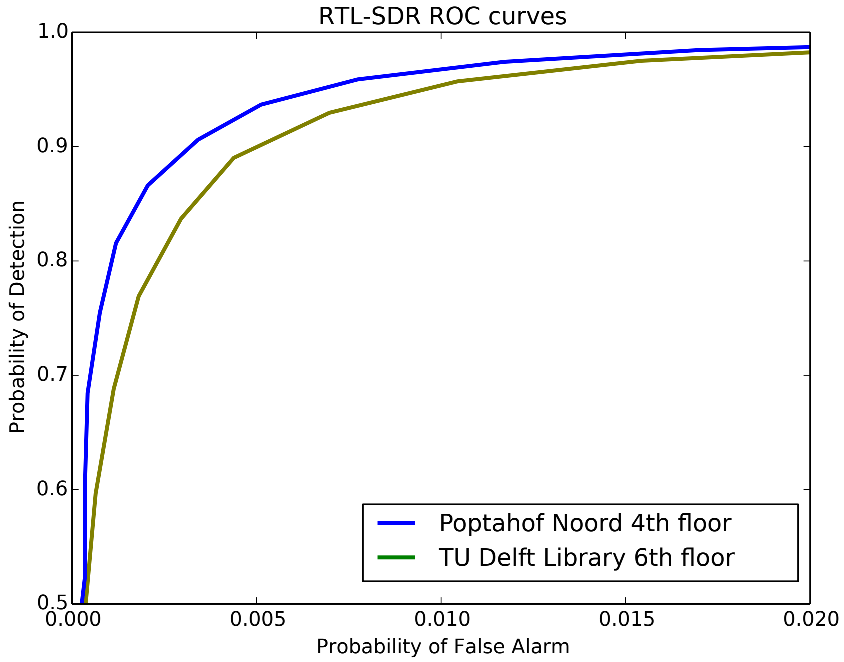


Figure 4.12: ROC curve of the receiver measured at two different places

In Figure 4.12, there is a difference between the curve obtained from the measurement at TU Delft and the measurement at Poptahof Noord. This indicates that the detector or receiver characteristic depends on the channels conditions, e.g. noise level, SNR, line of sight, and weather. The lower the probability of false alarm followed by the increasing probability of detection, the better the detector is.

5

Conclusion

Energy detection is indeed straightforward and easy to implement for detecting the presence of a signal without demodulating the signal itself, but it is hard to classify such kind of signals. By simply adding the ability of the detector to identify the signal based on bandwidth, we can improve the performance of the detector and reduce the probability of false alarm. Moreover, we can perform cooperative sensing by using multiple receivers located in different places in the area of interest. The results then can be combined and analyzed to get more precise results. Measurement sampling should also be increased to get better results when conducting the analysis of receiver characteristic (ROC).

Threshold value essentially affect the performance of the detector. Threshold value causes a trade-off between probability of detection and probability of false alarm. Therefore, threshold value should be selected carefully to get the optimal performance of the detector. By using ROC analysis, the optimal threshold value can be obtained.

References

[1] T. J. O'shea, T. C. Clancy, and H. J. Ebeid, *Practical signal detection and classification in gnu radio,* in *SDR Forum Technical Conference (SDR)* (2007).

[2] I. Poole, *What is dvb-t?* .

[3] D. G. Riviello, R. Garello, S. Benco, A. Perotti, and F. Crespi, *Sensing of dvb-t signals for white space cognitive radio systems,* (2013).

[4] Radio-TV-Nederland.nl, *List of dvb-t spectrum in netherlands,* .

[5] G. Radio, *Gnu radio wiki,* (2016).

[6] A. Csete, *Simple dvb with gstreamer and gnu radio,* (2011).

[7] S. Toledo, *Transmitting (and receiving) dvb-t on 1.2 ghz,* (2012).

[8] MatplotLib, *Plotting commands summary,* (2016).

[9] SciPy.org, *Numpy reference,* (2015).

[10] J. Johnson, *Python numpy tutorial,* (2016).

[11] R. Elsinga, *Scanner frequenties nederland,* (2016).