Wireless Networking - GNU Radio Assignment Report

Pranav Sailesh Mani
Student No. 4508467
TU Delft

April 30, 2016

Contents

1 Introduction

The main purpose of this assignment is to find the unused spectrum in the Television frequency band from 478 MHz to 862 Mhz. DVB-T standard is used in Europe for transmission of TV signals. DVB-T is an abbreviation for "Digital Video Broadcasting — Terrestrial"; it is the DVB European-based consortium standard for the broadcast transmission of digital terrestrial television that was first published in 1997[1]. In Delft, five DVB-T multiplexed operators are operating. The operator and the channel details are shown in Figure 1.

MUX Operator	Tx Location	Center Freq. (MHz)	Channel No.	Bandwidth (MHz)	ERP (kW)
RTS Bouquet 1	Delft	722	52	8	1
NTS1 Bouquet 2	Delft	698	49	8	1
NTS2 Bouquet 3	Delft	762	57	8	1
NTS3 Bouquet 4	Delft	498	24	8	1
NTS4 Bouquet 5	Delft	522	27	8	1

Figure 1: Details of channels used and the corresponding operator[2,3].

In this assignment, the main task would be to device a methodology for this purpose and detect if these channels are being used. This algorithm can be a starting point for a cognitive radio, which makes maximum utilization of empty channels. A USB based SDR RTL2832U is used for scanning the channels and GNU Radio software is used to check whether the channel is being used or is empty. We just make a measurement to check for channel usage and do not transmit any data.

2 Detection Methodology

In order to measure whether the channel is being used or not, we need to device a detection method. Th following steps are performed in the GNU Radio software to detect the presence of a signal:

1. The signal is first received from the USB dongle on the computer using the GNU radio software.

2. FFT is performed on the signal. This complex FFT vector is squared and divided by a factor so as to get the absolute value in different channels.

3. This signal is then compared with a threshold based on which we can infer whether the signal is present or not.

The flow chart for this method is shown in Figure 2 [2].

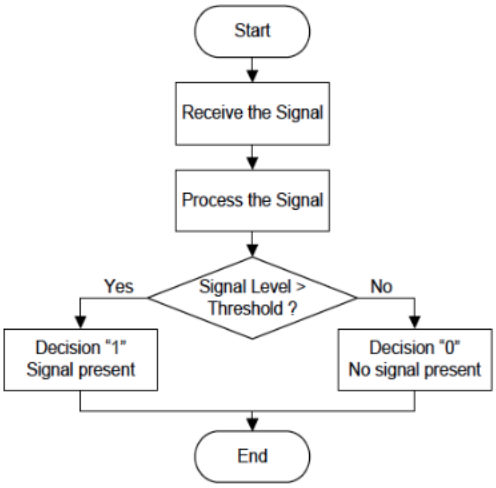


Figure 2: FlowChart of the algorithm[2,3].

The implemented software in GNU radio is shown in Figure 3.

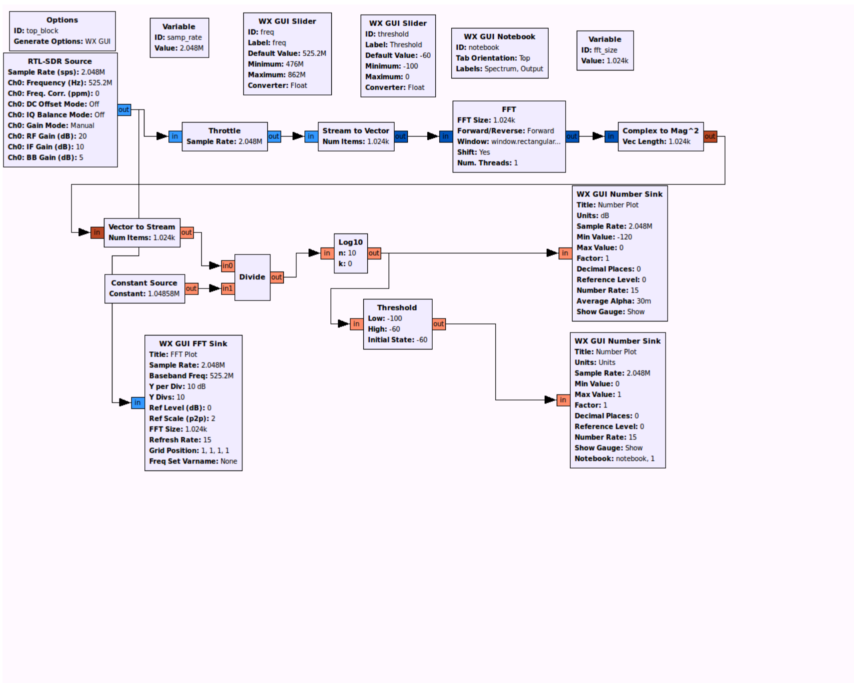


Figure 3: Implemented software in GNU Radio

3 Results and Analysis

In order to check for channel usage, it was decided to test the receiver at two places. One of the places chosen was Aan't Verlaat(Delft) which is the place where I live and the second place I chose was the TU Delft Library to check for channels. Aan't Verlaat is an housing based locality whereas TU Delft library is a university environment. The software was used to scan and find out 10 occupied and 10 empty channels at each location. The same channel frequencies were scanned at each location. The results obtained in Aan't Verlaat is tabulated in Table 1 and the results obtained in TU Delft Library is tabulated in Table 2. The frequency channel is taken as detected if the output is 1 in the GNU radio software. The mean and the standard deviation of the power of signals in detected channels is shown in Table 3.

Table 1: Obtained Results in Aan't Verlaat

Frequency Channel Scanned	Energy in dB	Detection
722(RTS Bouquet 1)	-59.53	Yes
698(NTS Bouquet 2)	-59.86	Yes
762(NTS Bouquet 3)	-57.23	Yes
498(NTS Bouquet 4)	-54.27	Yes
522(NTS Bouquet 5)	-57.89	Yes
480	-49.34	Yes
792	-57.56	Yes
600	-62.50	Yes
800	-59.86	Yes
816	-58.80	Yes
506	-71.20	No
516	-71.71	No
552	-70.72	No
590	-72.36	No
630	-73.35	No
670	-72.03	No
709	-71.38	No
720	-72.60	No
784	-74.32	No
823	-72.69	No

Table 2: Obtained Results in TU Delft Library

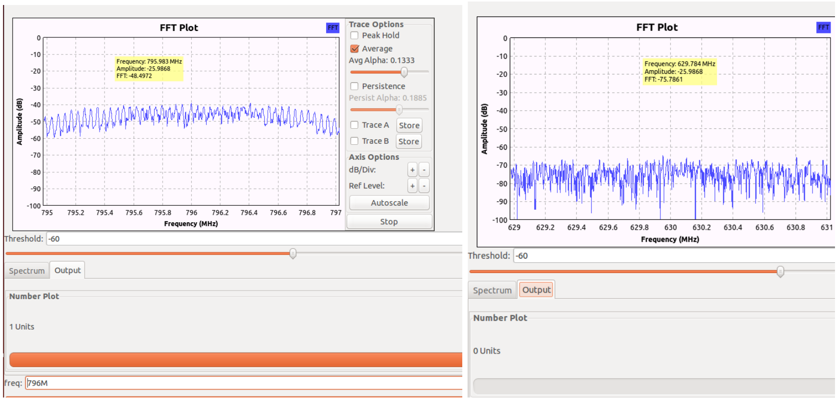
Frequency Channel Scanned	Energy in dB	Detection
722(RTS Bouquet 1)	-50.65	Yes
698(NTS1 Bouquet 2)	-51.64	Yes
762(NTS1 Bouquet 3)	-57.89	Yes
498(NTS1 Bouquet 4)	-52.10	Yes
522(NTS1 Bouquet 5)	-53.61	Yes
480	-48.02	Yes
792	-49.91	Yes
600	-62.5	Yes
800	-48.67	Yes
816	-57.23	Yes
506	-69.40	No
516	-72.01	No
552	-73.02	No
590	-72.60	No
630	-71.71	No
670	-72.36	No
709	-72.03	No
720	-73.20	No
784	-73.02	No
823	-71.38	No

Table 3: Mean and Standard Deviation of Signals

Signal	Mean(dB)	Standard Deviation
Detected Signals	-55.45	4.654
Non-Detected Signals	-72.15	1.068

Signal Detection in Software

The channel is taken as occupied/detected if the output is 1 in the GNU radio software. Examples of detected and undetected signals are shown in Figure 4. When testing, a lot of different kinds of signals were observed.



(a) Detected Signal at 796MHz (b) Channel Empty at 630MHz

Figure 4: Examples of occupied and empty channels

There were also instances where the software gave a wrong output due to signal noise. This is shown in Figure 5.

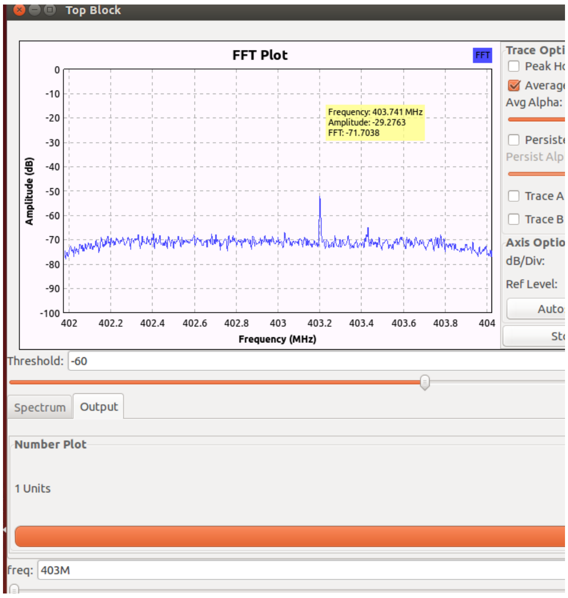
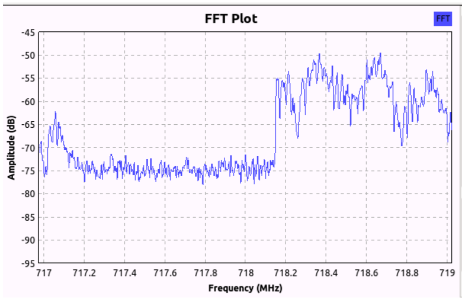
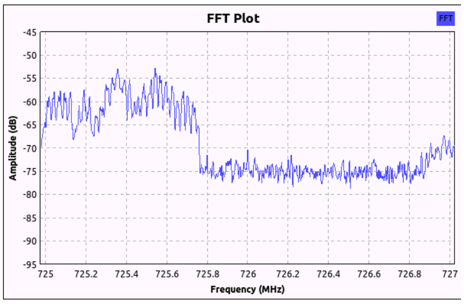


Figure 5: Wrong output due to Noise at 403 MHz

The starting and the ending point of the channels could also be observed during running of the program. This is shown in Figure 6 for channel at 722 MHz.



(a) Starting of Channel at 718MHz



(b) Ending of Channel at 726MHz

Figure 6: Starting and Ending points of Channel at 722MHz

Unknown Signals could also be detected at unexpected frequencies. An example of such a signal is shown in Figure 7.

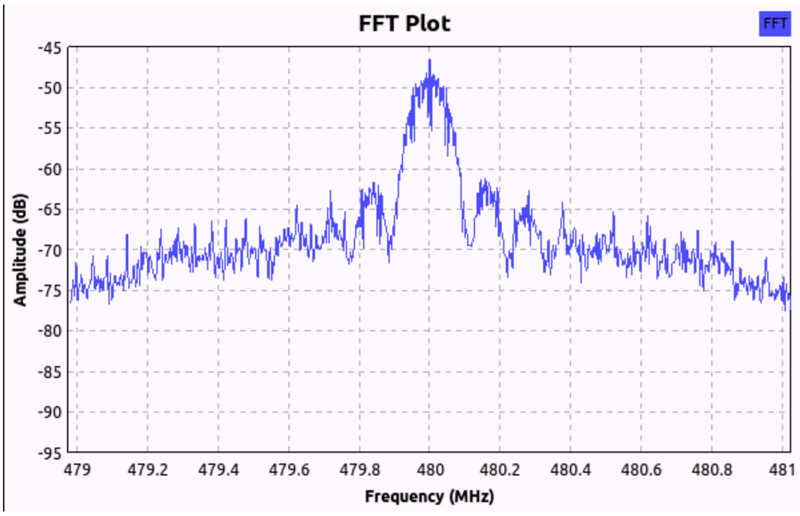


Figure 7: Unknown Signal at 480MHz

Receiver performance

In order to simulate and get an actual idea of the receiver performance and operating characteristics, the same Matlab script developed in [3] was used and the signal properties were changed to suit my antenna. Both the detected channel signals and the undetected channel signals are plotted using normal distribution. The distribution for the signals are shown in Figure 8.

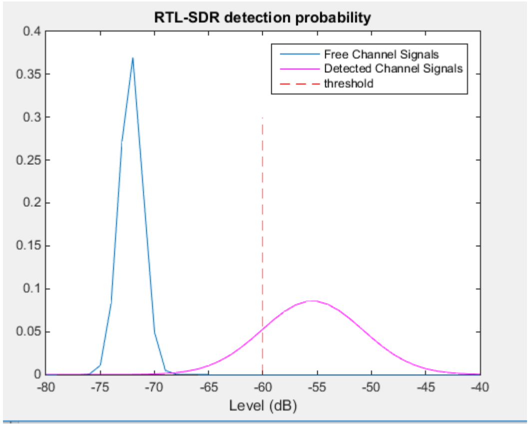


Figure 8: Signal characteristics

The probability of detection $P_d =$ is 0.8359 and the probability of false alarm P_{fa} is 0 for threshold = -60dB. We need to increase the number of detected packets i.e P_d. So, we can vary the value of threshold and calculate P_d and P_{fa} respectively. The values for different thresholds is tabulated in Table 4.

Table 4: P_d and P_{fa} for different values of threshold

Threshold(dB)	P_d	P_{fa}
-60	0.8359	0
-65	0.980	1.08e-11
-70	0.991	0.0221

From this table, we can infer that threshold value of -65 offers the best P_d and P_{fa}. Even though at threshold of 70, we have better detection, the false alarm probability also increases. So, the threshold vale of -65 would be the best for this receiver. It can be seen that there is a tradeoff between P_d and P_{fa}. The receiver operating characteristics were ideal and P_d almost reaches 1 sufficiently quickly. This is shown in Figure 9.

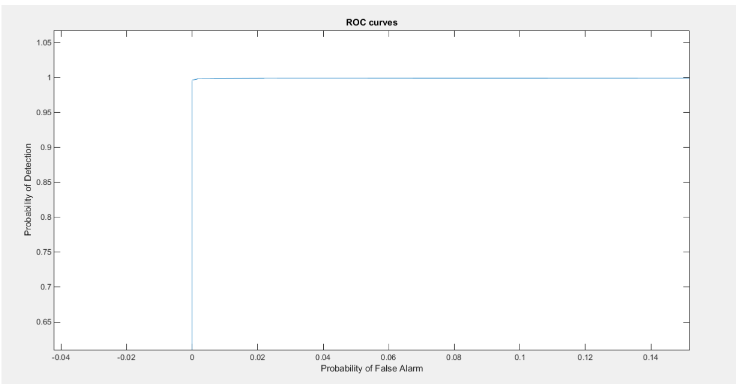


Figure 9: Receiver operating characteristics

4 Conclusion

In this assignment, only a single receiver was used. Multiple receivers could be used to improve detection. The algorithm depends on FFT to detect the signals due to which we could extract only limited information from the signals. However, this algorithm does the job well as the P_d and P_{fa} are sufficiently good for a threshold of -65dB. One disadvantage of this algorithm is that it depends on the receiver operating characteristics and the same results may not be observed for other receivers. Other new algorithms can be used which is not dependent on the receiver characteristics.

References

1. https://en.wikipedia.org/wiki/DVB-T
2. Lecture Slides
3. GNU Radio Project by Rizqi Hersyandika

5 Appendix

The Matlab script shown below was modified according to requirements of my receiver.

```
%This is a modified progam of the script in [3].
% Non-Detected Signal distribution
Pnotarget = makedist('Normal','mu', -72.15,'sigma',1.068);
% Detected Signal distribution mean = -59.95 dB & stdev = 9.32
Ptarget = makedist('Normal', 'mu', -55.453,'sigma',4.65);
threshold = -65 ; % threshold  in dB. Change according to needs
Pfa = 1 - cdf(Pnotarget,threshold) % probability of false alarm
Pd = 1 - cdf(Ptarget,threshold) % probability of detection
Level=[-80:-40];
figure(1);
plot(Level,Pnotarget.pdf(Level));
hold on
plot(Level,Ptarget.pdf(Level),'m');
title('RTL-SDR detection probability')
hold on
Y = 0:0.1:0.3;
X = threshold * ones(size(Y));
plot(X, Y, 'r--')
legend('Free Channel Signals ','Detected Channel Signals', 'threshold')
xlabel ('Level (dB)')
Pfa_ROC = 1 - cdf(Pnotarget,Level); % prob of false alarm
Pd_ROC = 1 - cdf(Ptarget,Level); % prob of detection
figure(2);
plot(Pfa_ROC,Pd_ROC);
title('ROC curves')
ylabel ('Probability of Detection')
xlabel ('Probability of False Alarm')
```