GNU Assignment

Shashwat Verma 4507746

April 30,2016

Contents

1 Introduction

Dynamic Spectrum Management(DSM) is required for finding empty frequency in the spectrum. White spectrum are the secondary frequency which is not used or rarely used by service provider. Purpose of DSM is to find out this white space and make it available to secondary user in such a way no primary users are affected.

In this assignment we will be scanning the Digital Video Broadcasting Terrestrial(DVB-T) range to find out white spaces and other broadcasting channels. Table 1 are the channels owned by kpn for DVB-T broadcast near South-Holland area.

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: DVB operator and frequency

Cities near delft have same channels at same frequency, but each city have their own broadcasting stations (towers) in order to attain more energy in near by areas.

2 Objective

In order to find out white space, valid channels must be detected. Objective of the assignment is to detect signals presented in table 1 in order to find white space in DVB-T frequency spectrum. Then probability of detection, miss-detection and false detection would be calculated.

3 Implementation

Implementation for this problem statement is done by implementing energy detection algorithm which detects signals greater than defined threshold value[1]. Figure 2 shows the algorithm used in deciding whether signal is present or not. Once signal is detected for particular frequency, signal processing is done on received signal in order to calculate energy of received signal in dB. Once energy is calculated, decision is made based on threshold value.

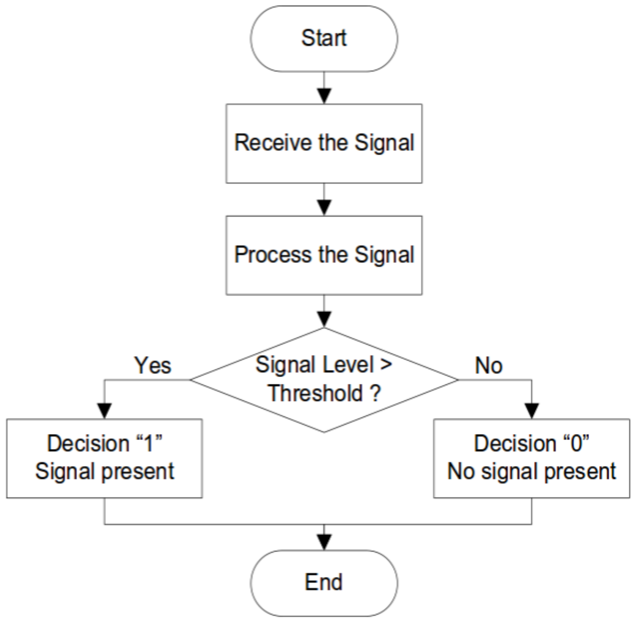


Figure 2: Energy [1]

Processing block in figure 2 includes a method to convert detected energy to dB value as shown in figure 3, it is done by squaring and taking FFT of received signals. Then this signal is converted into dB and passed to threshold block.

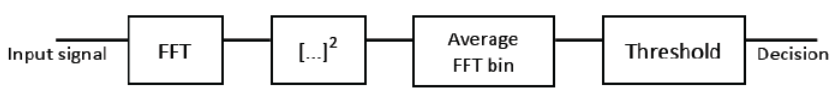


Figure 3: Signal processing [1]

Implementation of fig 2 and fig 3 is shown in fig 4 in GNU radio. The signal is received through RTL-SDR source. For this assignment, $DVB - T + DAB + FMUSB2.0$ dongle is used.
Stream to vector is used for converting input values into floating point vector array. Using these values FFT is calculated, these FFT result is used to generate dB values by using log10 block in

GNU radio software.This db value is passed through threshold block. Value of threshold output (1 or 0). The dB signal values are written into data file for further analysis through python script. The purpose of writing these values to file is to take average over 2M samples (1second), so the sudden jitter in signal is leveled out and average value is processed.

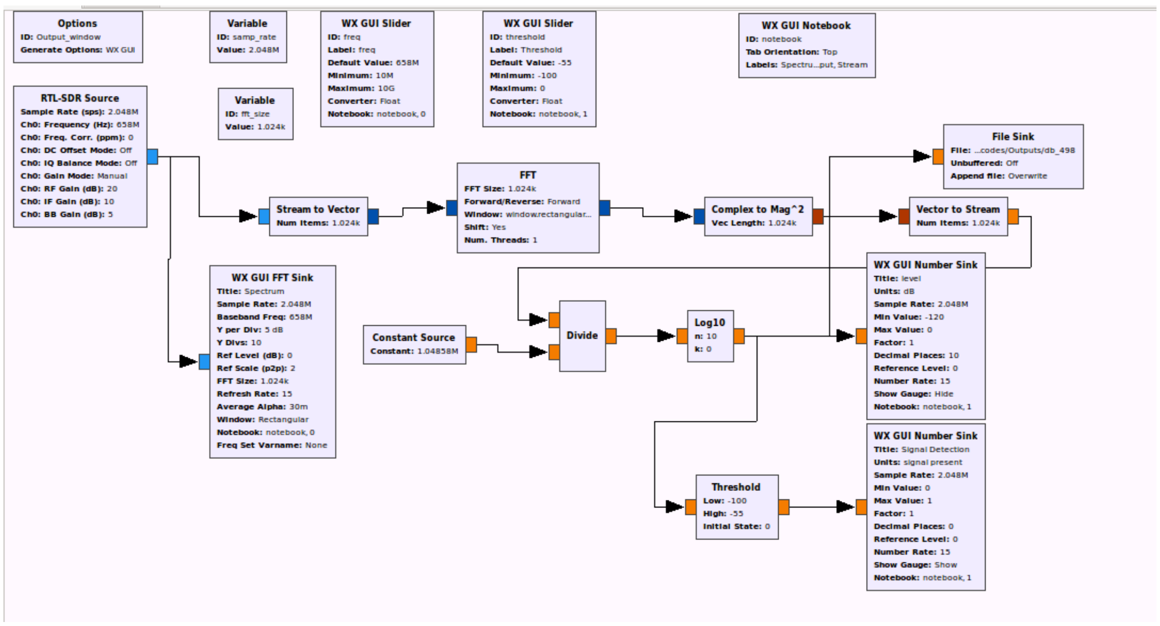


Figure 4: GNU radio code for energy detection

4 Results & Analysis

Following table shows values taken at TU Delft library and Poptahof Noord 2624RR. For these readings `detect.grc` and `file_read.py script` is used. Probability of detection and false detection are evaluated based on following table1.

Table 1: Signal Measurement Results

Frequency User	Center Frequency (MHz)	TU Delft Library Top Floor Average Level(dB)	Poptahof Noord 2624RR Average Level(dB)
RTS Bouquet 1	722	-51.15	-53.78
NTS1 Bouquet 2	698	-55.79	-59.68
NTS2 Bouquet 3	762	-58.18	-57.32
NTS3 Bouquet 4	498	-50.47	-54.78
NTS4 Bouquet 5	522	-54.93	-57.45
Unknown #1	550	-76.40	-70.27
Unknown #2	650	-75.96	-75.98
Unknown #3	750	-76.91	-74.23
Unknown #4	850	-77.10	-78.39
Unknown #5	800	-51.09	-51.35
Unknown #6	792	-50.71	-51.42
Unknown #7	796	-45.60	-45.47

In table 1 a non-listed channel is found at 796MHz. To make sure that its a channel and not a point jitter, readings were taken at bandwidth edges. These readings show that it is indeed a channel.

4.1 Signal Detection

Signal detection scenario is shown in fig 5 using graphic interface of GNU radio. In addition `file_read.py` will also decide whether signal is detected or not depending on average value. Readings are taken at frequency of 498MHz.

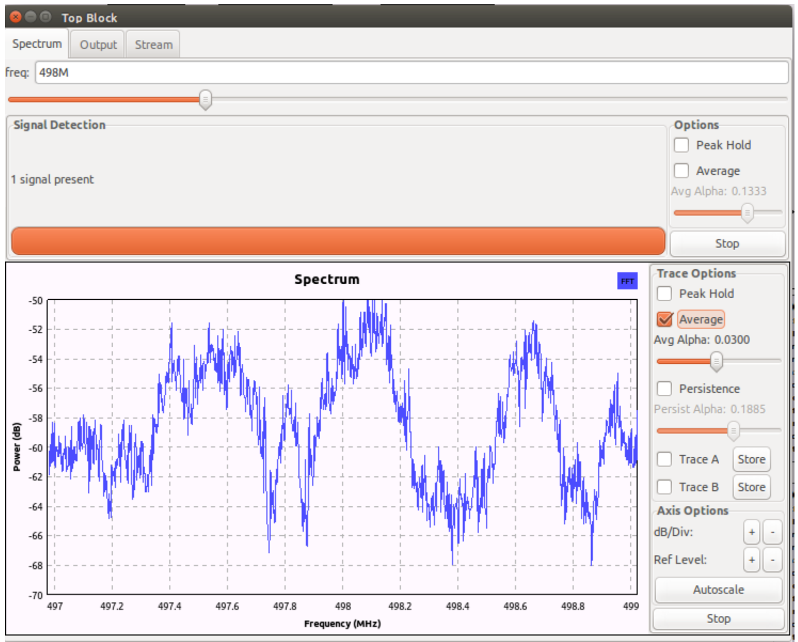


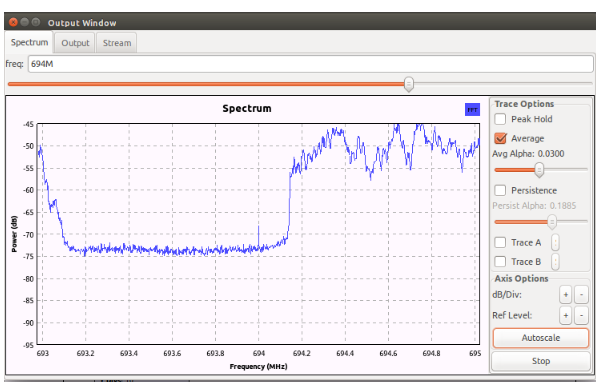
Figure 5: Signal detection

4.2 Bandwidth Scanning

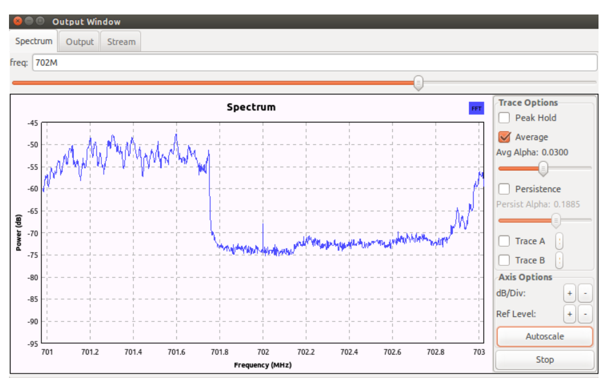
DVB-T channels under consideration have 8MHz bandwith as shown in Figure 1. After checking signal strength at boundry values it is confirmed that these channels have 8MHz bandwidth.

In Figure 6 edges for channel 498MHz are shown.

Taking average signal strngth we can conclude that indeed signal is present there.



(a) Rising edge of bandwidth at 494MHz



(b) Falling edge of bandwith at 502MHz

Figure 6: Edges of 8MHz bandwidth of channel at 498MHz

4.3 Signal Miss Detection

When searching for corner signals, few ripples were observed in spectrum. This particular frequency of 480MHz was miss detected but a signal was present there. This was maybe due to poor strength of received signal.

It can be seen in figure 7

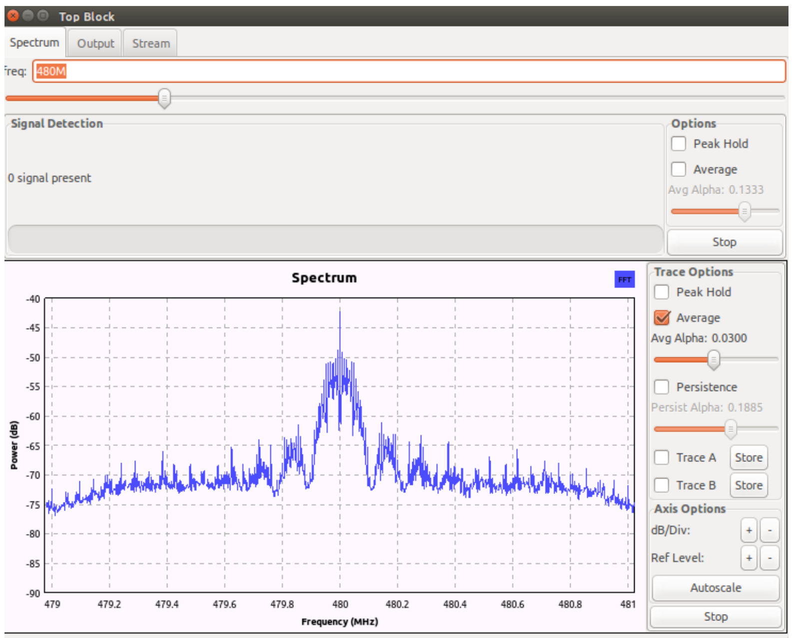


Figure 7: Miss detection scenario

4.4 Signal False Detection

While scanning the spectrum certain frequencies were present, but they don't have any bandwidth. One of these frequency is detected at 403.2MHz.

This may lead to loss in free spectrum usage. This scenario is shown in fig 8. We can observe that signal strength is -72db which is less than threshold. But still we are getting signal as present.

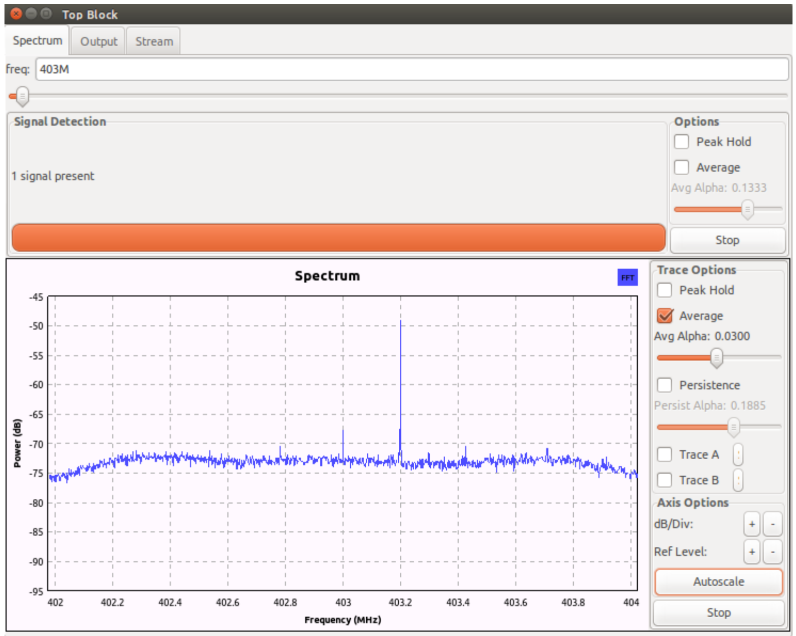


Figure 8: Miss detection scenario

5 Probabilities & ROC

Probabilities of detection and false alarm can be plotted using matlab code. Curves are shown in fig 9. This curve is plotted for threshold -60db from the readings shown in above tables. From the same matlab code values for detection and false alarm probabilities are calculated and roc curve is plotted in fig 10.

$Prob\_detection = 0.9527$

$Prob\_false = 7.0875e - 04$

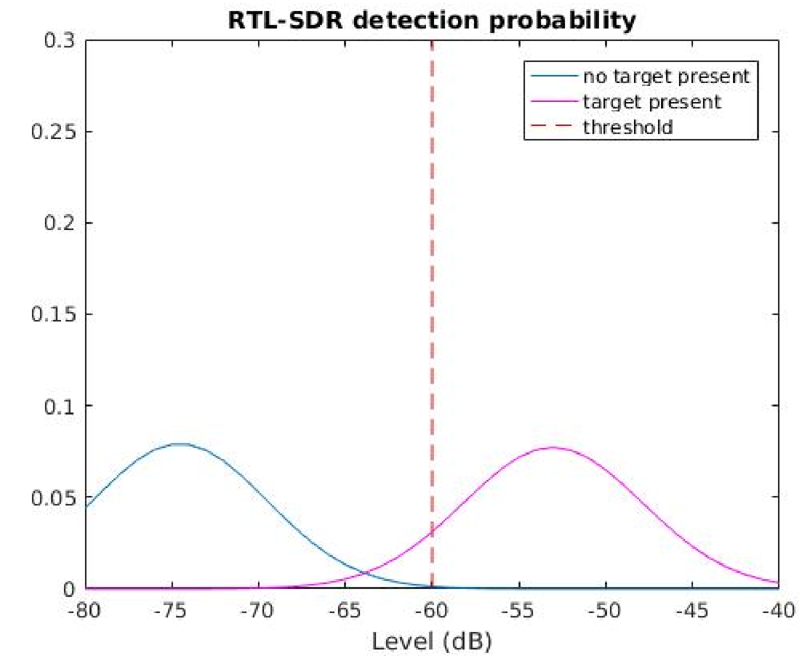


Figure 9: Detection Probability

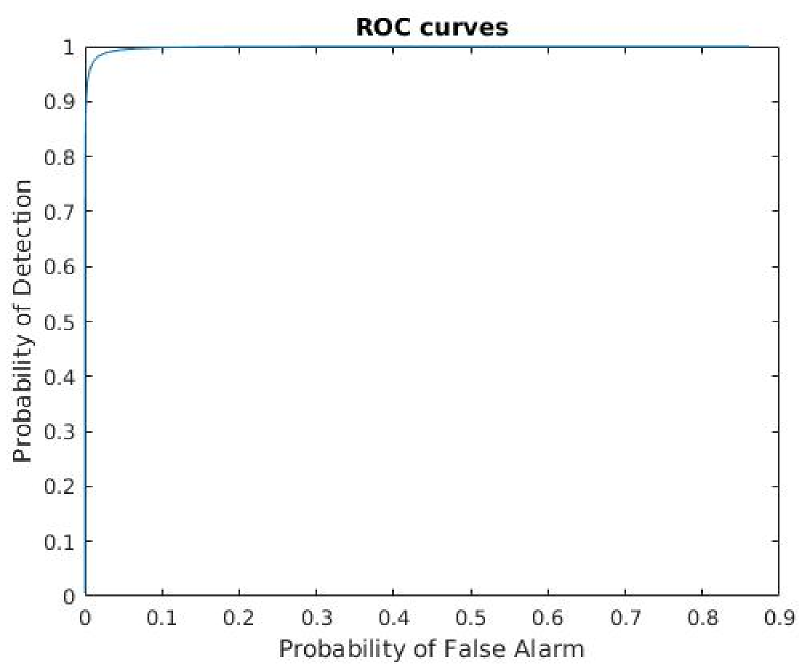


Figure 10: ROC Curve for Detection and False detection

6 Conclusion

From the result it can be concluded that energy detection method can be used for signal detection in order to find out white space. While using this method for energy detection there is always chance of miss detection or false alarm.

To overcome the scenario of miss detection and false alarm many readings should be taken from nearby places to confirm the result. Best method is taking multiple readings at one place in order to get a final result.

There is always a trade-off when between false-detection and detection when lowering threshold value, as our system can detect high frequency white-noise present in surroundings. So threshold should be kept in between obtained probabilities for best result.

7 References

- Lecture slides
- http://gnuradio.org/
- F. E. Visser, G. Janssen, P. Paweczak, Multinode Spectrum Sensing Based on Energy Detection for Dynamic Spectrum Access, Proc. IEEE
- Design from Rizqi Hersyandika(4410106)

8 Appendix

8.1 Python Script for average

```
import scipy
data = scipy.fromfile(open('db_498'), dtype=scipy.float32)
file2 = open('db_values.txt','a')
print 'Length of data is %d' % (len(data))
avg=sum(data[-2000000:]) / float(len(data[-2000000:]))
print 'Average of last 2M samples are: ', avg
```

```
file2.writelines('Value for 498 MHz: '+ str(avg)+'\n')
```

8.2 Matlab code for ROC and probabilities

```
%distribution mean for channel not detected
Pnotarget = makedist('Normal','mu', -74.532,'sigma',3.037);

%distribution mean for channel detected
Ptarget = makedist('Normal', 'mu', -53.046,'sigma',4.163);
threshold = -60 ; %dB

Pfa = 1 - cdf(Pnotarget,threshold) % prob 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('no target present', 'target present','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')
```