MIDDLE EAST TECHNICAL UNIVERSITY

ELECTRICAL AND ELECTRONICS ENGINEERING DEPARTMENT

EE435 Term Project
Phase 2

Group 10
Emre Doğan 2093656
YiğitTopoğlu 2094548

Introduction

In the first phase of this project, we had studied on the hypothesis testing of several AM/FM modulated signals. The main purpose was to design a system detecting the signal type by the characteristics of AM&FM modulating. To do this, we decided the signal type whether AM or FM in the first step. Then, we tried to decide the modulation index of the corresponding input signal. Then, we considered the real-world problems regarding to this system by adding some noise and CFO effect of the transmission process.

In this phase of the project, we were asked to identify the type of the signal that we receive using RTL-SDR dongles.Both the carrier frequency and signal type should beresultedby our system with respect to the signal type produced by the USRP device. Actually, we protected our 1st phase system to make the hypothesis decision and rearranged it to find the carrier frequency.

The general block diagram can be seen in Figure 1.

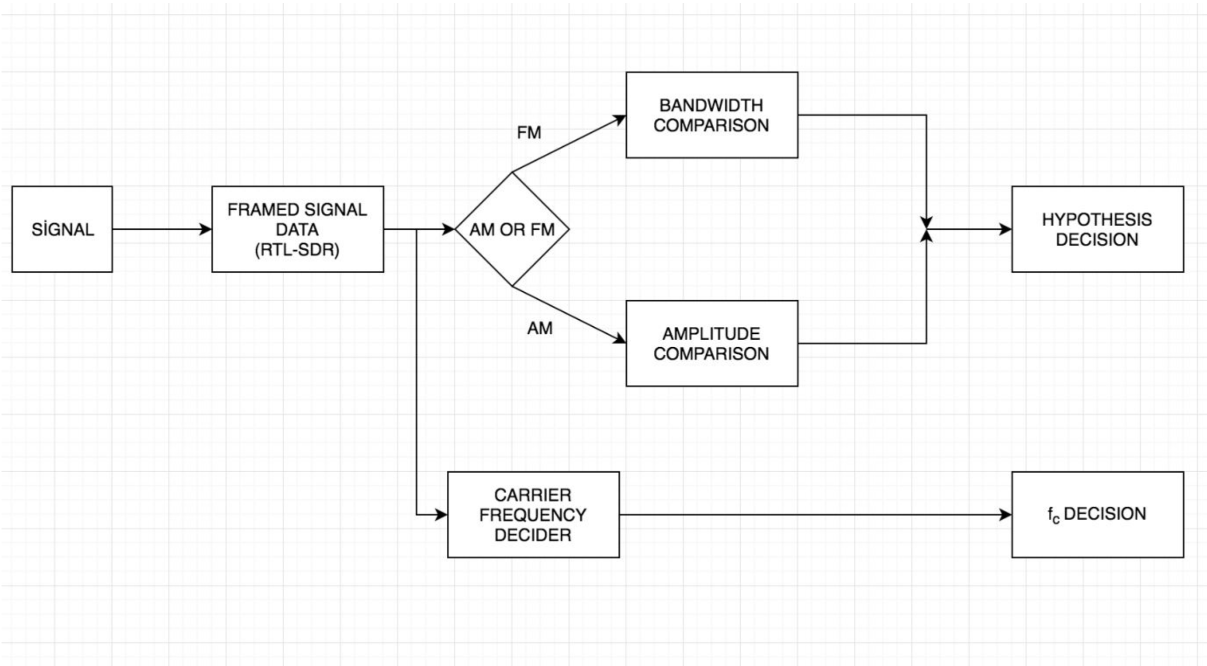

Figure 1. Overall Block Diagram of the System.

All parts will be explained in detail in the following parts.

Getting the Data & Framing

To be able to receive the signal data from RTL-SDR dongle, framing is applied such that there will be 3 frames, each consisting of 1024 data. After this procedure, we will have 3072 data corresponding to each input signal received by the receiver. For signal detection we take fft of a taken frame, and we apply a threshold value that distinguishes a noise and the signal itself. If the threshold is exceeded, this means that a signal is detected. If the signal is found , we vertically combine all three frames to make a 1x3072 sized frame. To avoid possible errors, we ignored the first 2000 samples and saved the last 1072 data to be processed. The number of frames is not certainly unalterable but we considered the speed of the system and did not increase it too much.

Then, the framed version of signal is sent to both carrier frequency identifier and signal identifier functions.

The carrier frequency identifier simply takes 3 frames -3072 data-, and decides the carrier frequency.

The signal identifier function is created in the 1st phase of the project to basically check the input signal and decides which hypothesis (AM/FM signal with high/low modulation index) it represents.

Signal Identifier

As there are 4 possible hypotheses to identify, we needed two reference signals named s2_b and s4_b to represent 2nd and 4th hypotheses, AM with high modulation index and FM with high modulation index.

```
s4_b=exp(j.*(7.5.*sin(2.*pi.*3000.*t)+2.*pi.*100000.*t));
s2_b=(1+0.2.*(cos(2.*pi.*3000.*t)+cos(2.*pi.*6000.*t)+cos(2.*pi.*9000.*t)+cos(2.*pi.*12000.*t))).*exp(j.*2.*pi.*100000.*t);
```

Figure 2. Corresponding code to implement reference signals.

After the input data is taken, the bandwidth of its fft is calculated and compared with the bandwidths of fft of s2_b and s4_b signals. This approach helps us to separate AM and FM signals as the bandwidth of FM signals is much greater than the bandwidth of AM signals.

If the signal is AM, we look at the amplitude of the message part in fft data. With changing modulation index, the amplitude of the message part in fft data changes. By using this property, we can distinguish 2 AM signals with different modulation indexes.

If the signal is FM, then we investigate the bandwidths of signal’s fft. With changing modulating index, the signal’s bandwidth changes. So, if these two bandwidths are known, the signals can be separated from each other.

The ease of the classification of FM signals with different indexes can easily be observed in Figure 3.

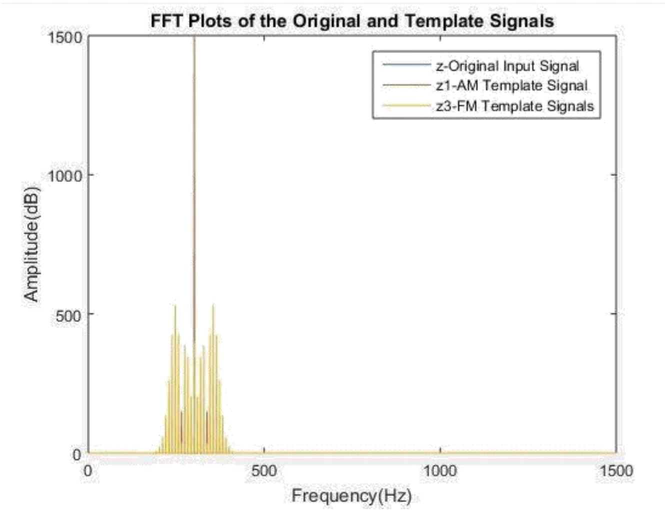

Figure 3. Bandwidth Comparison of the Original and Template Signals.

Carrier Frequency Identifier

The system should be able to detect the carrier frequency produced randomly by USRP. The center frequency is randomly selected between 200 MHz and 400 MHz. The MATLAB code to find the center frequency is shown in Figure 4.

```
d=abs(fft(rtlsdr_datagod));
f=find(d==max(d));
if (1<f && f<rtlsdr_frmlen*nfrmhold/2) || (f<rtlsdr_frmlen*nfrmhold/2+3 && f>rtlsdr_frmlen*nfrmhold/2-3)
cent_freq=round((rtlsdr_tunerfreq(c)+f/length(rtlsdr_datagod)*rtlsdr_fs)/1e6)
else if rtlsdr_frmlen*nfrmhold/2-1<f && f<rtlsdr_frmlen*nfrmhold
        cent_freq=round((rtlsdr_tunerfreq(c)+(f/length(rtlsdr_datagod)*rtlsdr_fs)-rtlsdr_fs)/1e6)
    else cent_freq=rtlsdr_tunerfreq(c)/1e6
    end
end
```

Figure 4. The center frequency finder.

As seen in Figure 4. to find the center frequency we look at the fft data of the 3072 sample we combined. Our sampling rate is 2.4 MHz, so the code is making 2.4 MHz jumps between 200 MHz and 400 MHz when taking frames. The loop stops when a signal is detected and rtlsdr_tunerfreq value will give the nearest value to the center frequency among the stopped frequencies to take frames (in our case the stopped frequencies will be 200,202.4,204.8 ...).

First we find the sample that contains the maximum of fft data. Then we define whether it is the right half of the fft or left half. If it is in the left half, we add $\frac{\text{The sample that contains the maximum of fft data}}{\text{total sample number (3072)}} * sample\ rate(2.4\ MHz)$ value to the rtlsdr_tunerfreq value we found. If it is in the right half, we first subtract 1 sampling rate (2.4 MHz) from rtlsdr_tunerfreq value, then do the same thing we did in the left half case. Finally we round it to get rid of the decimals.

Last, we send both the decision coming from our signal identifier and the center frequency value coming from the carrier frequency finder to the server.

Debugging Results

After completing the code, there was a long debugging process. We tried our system with a large number of input signals so that we could ensure that it works fine for all kind of signals.

There was not a problem due to FM signals and the system could detect Hypothesis 3&4 successfully. But in AM modulation, we faced some problems. As we used amplitude comparison in AM signals, the system was not as stable as required at the first time. So, we spent some time on deciding the threshold value to be used in AM signal detection and succeeded it.

There are some example screenshots corresponding to successful debugging trials below.

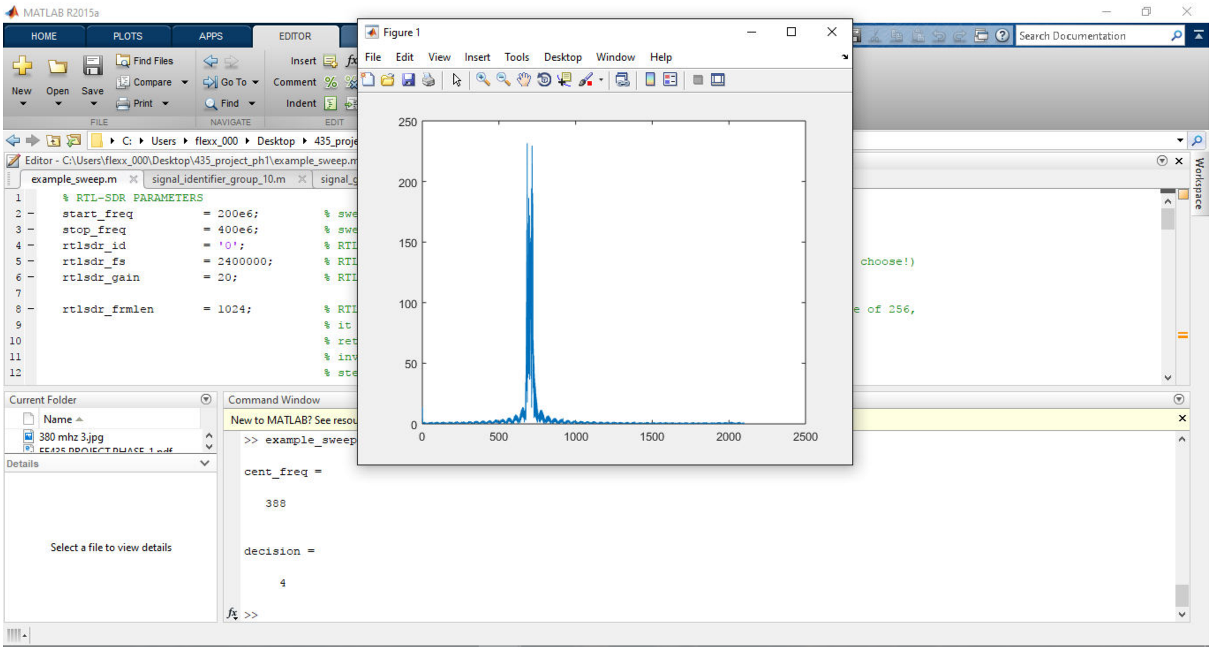

Figure 5. Debugging result of a Hypothesis 4 signal with 388 MHz carrier frequency.

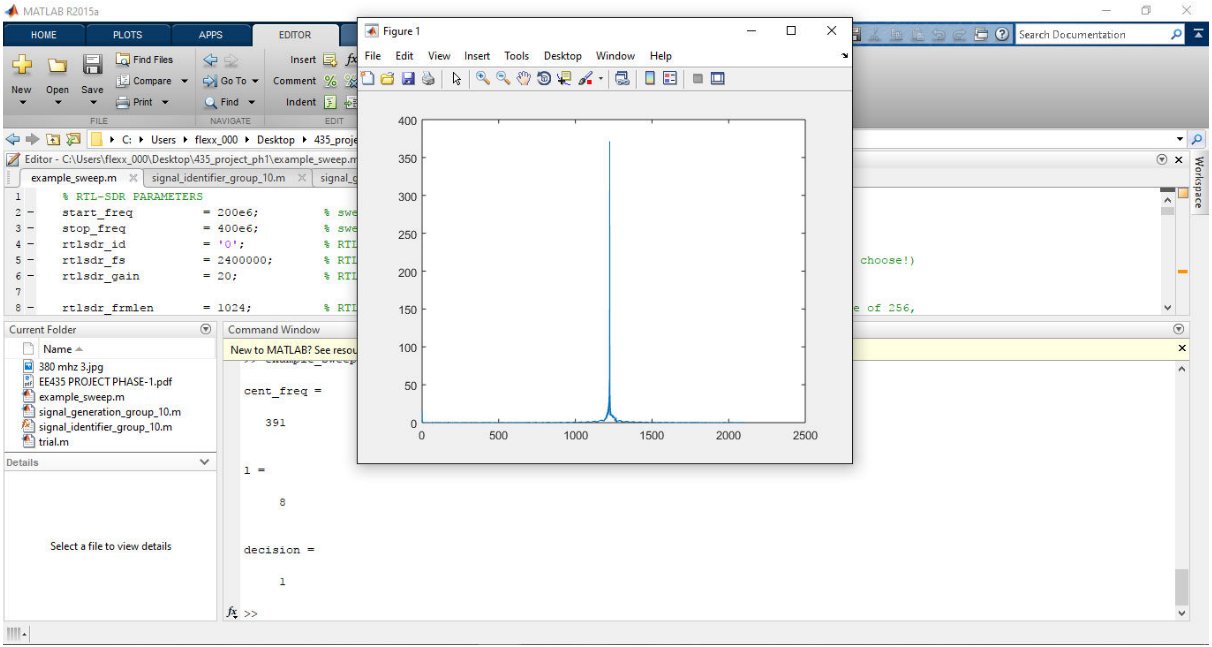

Figure 6. Debugging result of a Hypothesis 1 signal with 391 MHz carrier frequency.

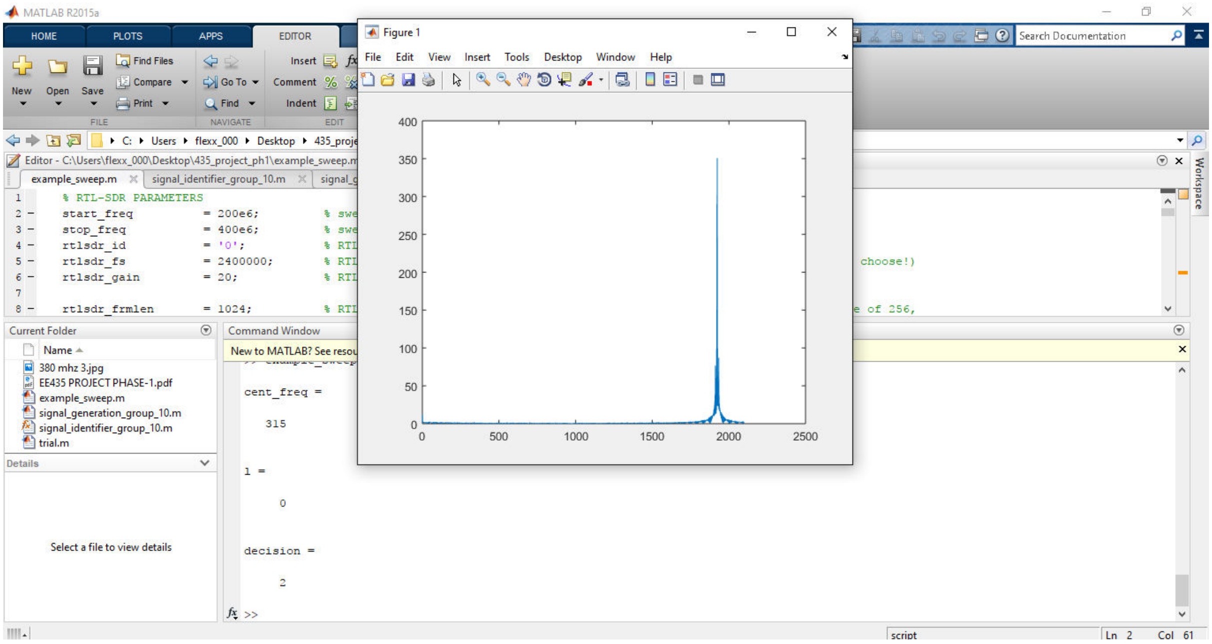

Figure 7. Debugging result of a Hypothesis 2 signal with 315 MHz carrier frequency.

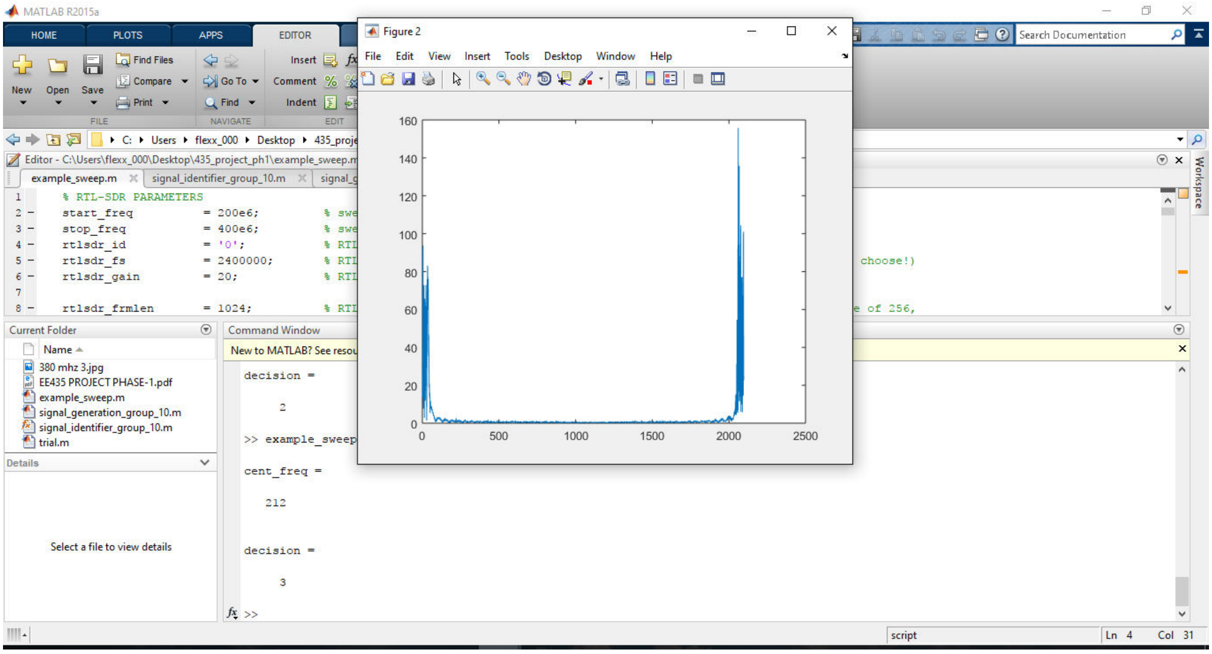

Figure 8. Debugging result of a Hypothesis 3 signal with 212 MHz carrier frequency.

Demonstration Results

In the final demonstration, the system's performance was investigated with randomly chosen several signals with different properties.

At the first stage, the USRP Device broadcasted each signal with a high SNR value through 60 seconds duration.In this step, our system successfully found the decision type and carrier frequency of all signals. All data can be seen in Table 1.

MEASUREMENT NUMBER	CARRIER FREQUENCY	DECISION
1	259	4
2	238	4
3	325	3
4	216	2
5	355	2
6	287	3
7	261	3

Table 1. Measurement Results of Demonstration when SNR is large and time duration is long

Then, the broadcasting time was lowered. But time was not a problem for us as our code results in 17 seconds with the worst case, when the carrier is higher 390 MHz. There is only one missing signal and it is not due to any system failure. We made a timing mistake during the demonstration. All results of this part can be seen in Table 2.

MEASUREMENT NUMBER	CARRIER FREQUENCY	DECISION
1	302	4
2	X	X
3	276	4
4	307	1
5	388	3
6	310	1
7	317	4

Table 2.Measurement Results of Demonstration when SNR is high and time duration is short

After this, the SNR value was lowered and at this step our system could not detect any AM signals. This happened because as we used amplitude comparison in AM detection and loss in SNR makes it harder to compare the amplitude. The resulting data can be observed in Table 3.

MEASUREMENT NUMBER	CARRIER FREQUENCY	DECISION
1	X	X
2	246	3
3	X	X
4	X	X
5	X	X
6	385	3
7	237	3
8	325	3

Table 3. Measurement Results of Demonstration when SNR is lower and time duration is short.

Then, the CFO effect is added to the input signals. The effect of CFO was not a problem for our system and it detected all signals except one, it was not a system related miss. We missed it due to wrong timing during demonstration. The resulting data of CFO effected input signal can be seen in Table 4.

MEASUREMENT NUMBER	CARRIER FREQUENCY	DECISION
1	222	3
2	282	3
3	252	4
4	342	4
5	223	2
6	X	X
7	260	3

Table 4. Measurement Results of Demonstration with CFO Effect

Lastly, SNR value of input signal is lowered too much. At this level, we did not expect to detect the signal's hypothesis decision but as our carrier frequency algorithm works separately, carrier frequency might be found. But at the final stage, only two signals could be detected and this shows the importance of SNR concept in any communication channel. The related data can be seen in Table 5.

MEASUREMENT NUMBER	CARRIER FREQUENCY	DECISION
1	346	1
2	X	X
3	X	X
4	X	X
5	246	3

Table 5. Measurement Results of Demonstration when SNR is very low.

Conclusion

In this term project, we have built and algorithm that distinguishes 2 AM signals and 2 FM signals with different modulation indexes from themselves and we have implemented this algorithm to a real-time application such by deciding which signal is which from signals taken by RTL-SDR. This project gave a good experience about real-time applications in telecommunications area. We highly believe that the experience we got throughout this semester while doing the project will be beneficial in our future engineering careers.