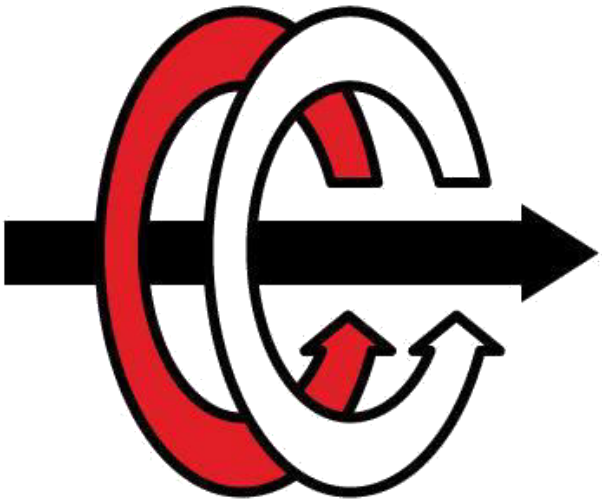

Middle East Technical University Electrical-Electronics Engineering Department

EE435 Term Project
Phase 1

#Group 10#

- **E re Doğa**
  2093656
- **Yiğit Topoğlu**
  2094548

Introduction:

In this project, we 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.

First, we plotted each possible input signal. By observing these plots, we tried to find some criteria so that we could distinguish each possible input signal.

In this report, these criteria will be discussed with the related plots & code blocks. The overall block diagram of the system is shown in Figure 1.

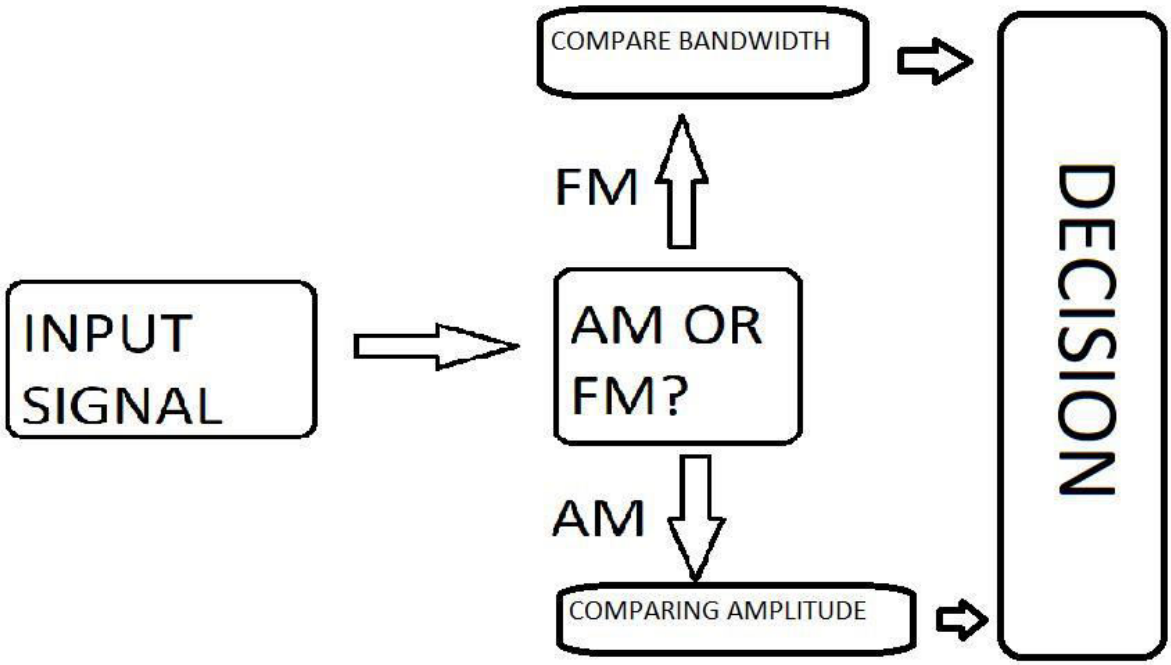

Figure 1. Overall block diagram.

At the first stage, we created 2 ideal signals with the specifications given to us. Using fft command in MATLAB, we compared the input signal with the ideal template signals defined in the function to decide whether the input signal is AM or FM modulated.

At the second stage, after deciding the type of the input, we tried to decide the modulation index of the input signal. To able to do this, we compared the Fourier transform plots of two signals. In this step, our signal is already defined as AM or FM modulated. But as there are 2 predefined signals for both AM and FM type with different modulation index, we must specify which is identical to our input signal.

Then, we considered the real-life problems regarding to this system. There will be some distortion on the receiving part due to noise and so we added a random signal to our inputs to represent the noise effect. At this part, we made some rearrangements on our code, as we needed more finely adjusted boundaries to decide the signal.

And at the last stage, we considered the CFO (Carrier Frequency Effect) of the transmission process and again made some corrections on our code, so that it works under any conditions.

Before starting to mention about our algorithm, it may be helpful to give some information of our code file. There are two .m (MATLAB Script) files consisting our system. One of them (signal_identifier_function_group_10.m) includes the function identifying the signal. This function will take two arguments (input signal and sampling frequency) and return the decision (type of the input signal).

The other file is signal_generation_group_10.m . In this file, first the fs should be defined and all the signals (with or without noise and CFO) will be generated and then an input signal must be chosen as input_sig. Then at the end of the code of this .m file, a command that runs signal_identifier_function_group_10.m with the inputs input_sig and fs is included. First signal_generation_group_10.m must be run to execute compliation of signal_identifier_function_group_10.m,.

Also, the input duration variable of the two files must be equal and adjusted manually.

1st Step:

In the first step, we are required to take s1,s2,s3,s4 (signals without noise and CFO) in the Project Phase-1 Definition as inputs, as shown in Figure 2 and define them.

```
s1=(1+0.02.*(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);
s2=(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);
s3=exp(j.*(15.*sin(2.*pi.*3000.*t)+2.*pi.*100000.*t));
s4=exp(j.*(7.5.*sin(2.*pi.*3000.*t)+2.*pi.*100000.*t));
```

Figure 2. Corresponding code to implement the s1, s2, s3, s4.

As stated in the introduction we first define the signals whether it is FM or AM from the bandwidth of the fft data. Our template signals to compare with input signals are s2 and s4. AM signals have much less bandwidth than FM signals. Hence it can be easily distinguished. Then 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. With proper filtering, we can distinguish 2 AM signals with different modulation indexes. If the signal is FM, with changing modulation index, the bandwidth of the message part in fft data changes. If the bandwidth is known, 2 FM signals with different modulation indexes can be easily classified. The results at fs=3.6 Msamp/s are shown in Figure 3, 4, 5 and 6.

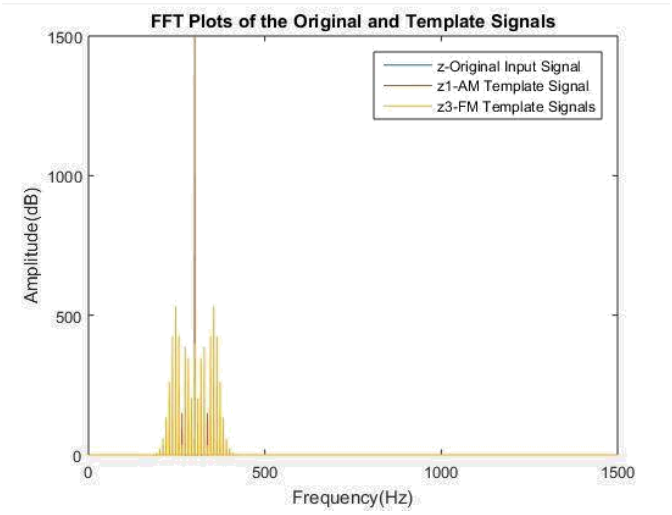

Figure 3. FFT Plots of the Original and Template Signals.

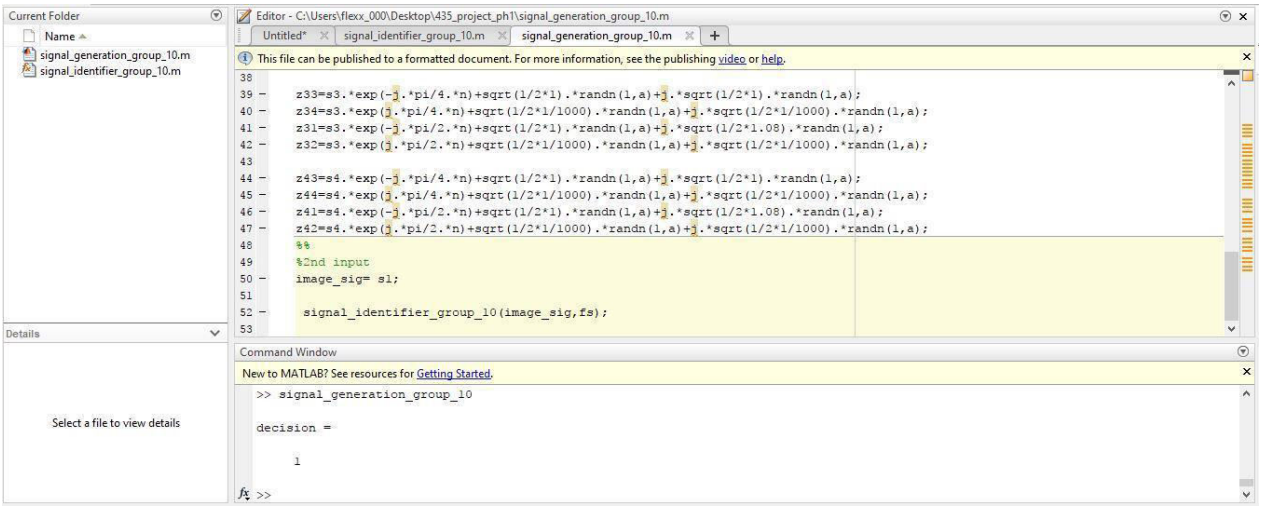

Figure 4. Screenshot of the MATLAB Interface after executing the function with input s1.

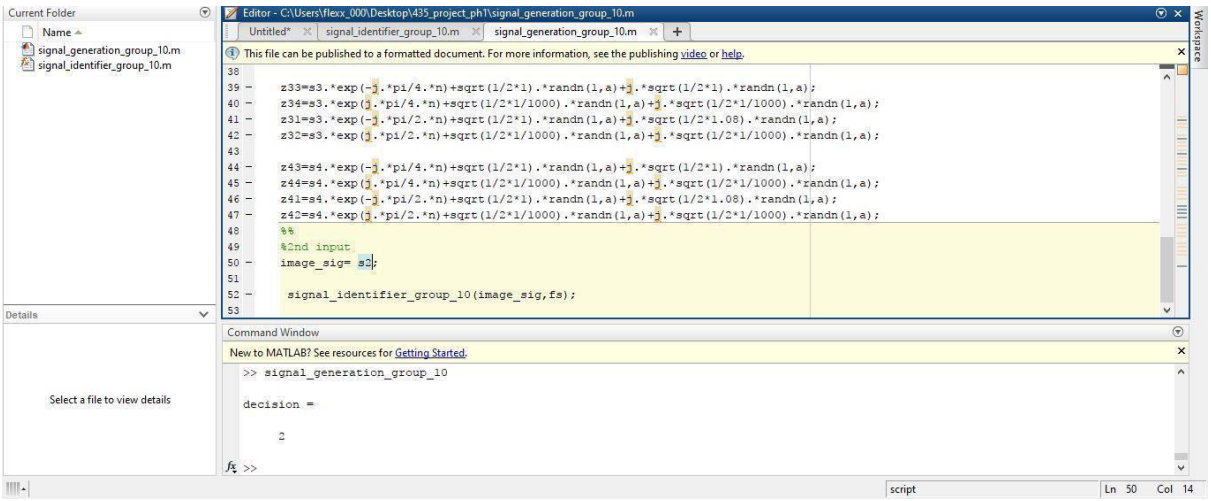

Figure 5. Screenshot of the MATLAB Interface after executing the function with input s2.

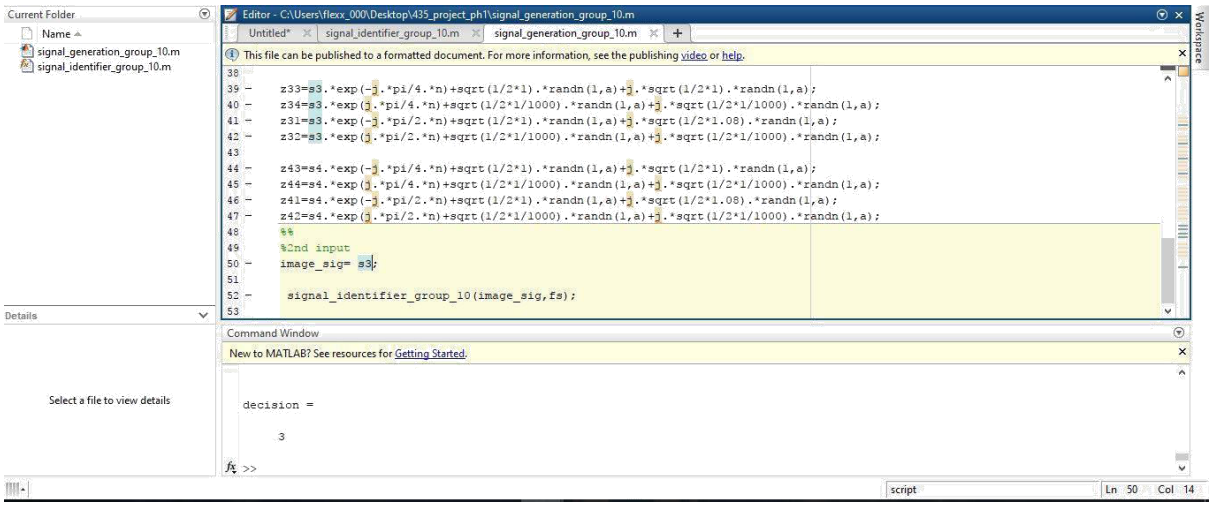

Figure 6. Screenshot of the MATLAB Interface after executing the function with input s3.

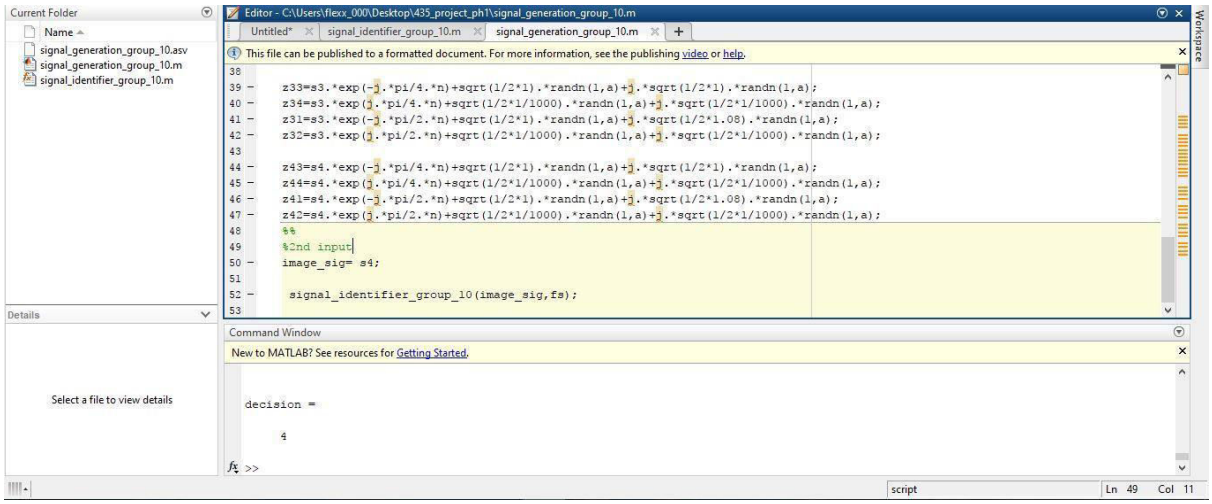

Figure 7. Screenshot of the MATLAB Interface after executing the function with input s4.

As shown in the figures, the signal identifier functions properly.

2nd Step:

In the second step, we are required to take the noisy versions of s1,s2,s3,s4; which r1, r2, r3, r4 in the Project Phase-1 Definition as inputs, as shown in Figure 8 and define them.

```
r1_0db=s1+sqrt(1/2*1.008).*randn(1,a)+j.*sqrt(1/2*1.008).*randn(1,a);
r2_0db=s2+sqrt(1/2*1.08).*randn(1,a)+j.*sqrt(1/2*1.08).*randn(1,a);
r3_0db=s3+sqrt(1/2*1).*randn(1,a)+j.*sqrt(1/2*1).*randn(1,a);
r4_0db=s4+sqrt(1/2*1).*randn(1,a)+j.*sqrt(1/2*1).*randn(1,a);
r1_10db=s1+sqrt(1/2*1.008/10).*randn(1,a)+j.*sqrt(1/2*1.008/10).*randn(1,a);
r2_10db=s2+sqrt(1/2*1.08/10).*randn(1,a)+j.*sqrt(1/2*1.08/10).*randn(1,a);
r3_10db=s3+sqrt(1/2*1/10).*randn(1,a)+j.*sqrt(1/2*1/10).*randn(1,a);
r4_10db=s4+sqrt(1/2*1/10).*randn(1,a)+j.*sqrt(1/2*1/10).*randn(1,a);
r1_20db=s1+sqrt(1/2*1.008/100).*randn(1,a)+j.*sqrt(1/2*1.008/100).*randn(1,a);
r2_20db=s2+sqrt(1/2*1.08/100).*randn(1,a)+j.*sqrt(1/2*1.08/100).*randn(1,a);
r3_20db=s3+sqrt(1/2*1/100).*randn(1,a)+j.*sqrt(1/2*1/100).*randn(1,a);
r4_20db=s4+sqrt(1/2*1/100).*randn(1,a)+j.*sqrt(1/2*1/100).*randn(1,a);
```

Figure 8. Corresponding code to implement the r1, r2, r3 and r4 under 0 dB,10 dB and 20 dB.

The noise levels are 0 dB, 10dB, and 20 dB. One problem in this step occurs, when we try to distinguish s2. Because of the random noise, it is hard to distinguish the template signal and original signal. To solve this problem, we take the difference of fft data of the message signal. When the difference becomes very little it means that our signal is very similar to the template AM signal. The results when fs=3.6Msamp/s are shown in Figure 9 to 11.

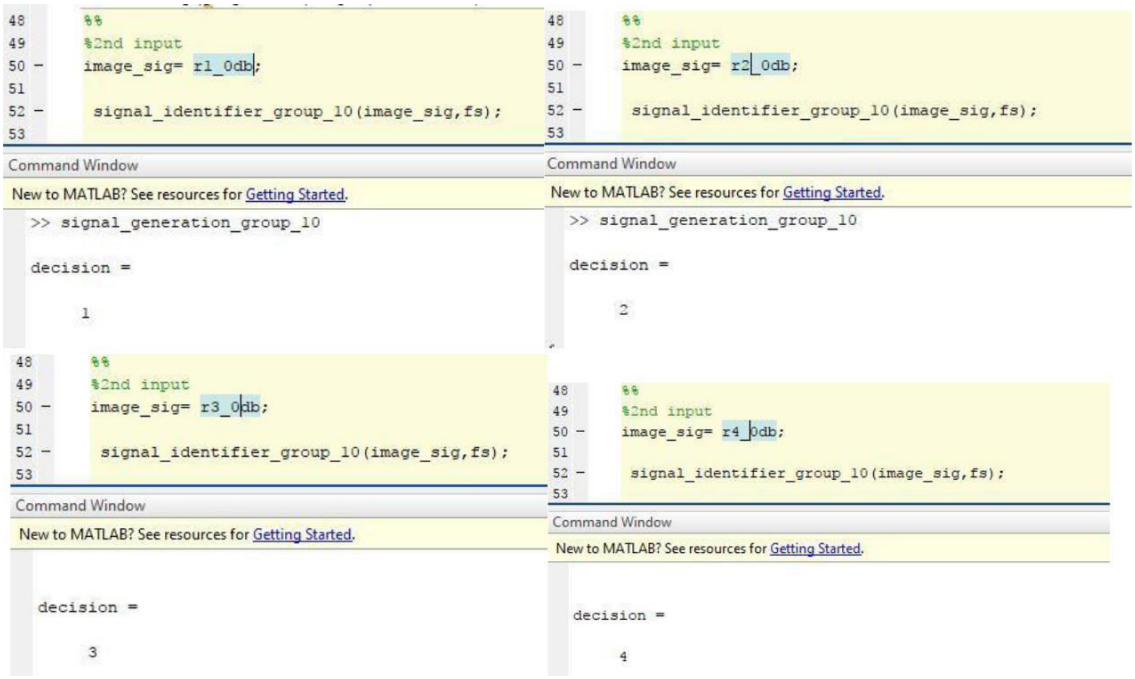

Figure 9. Screenshot of the MATLAB Interface after executing the function with input r1, r2, r3 and r4 under 0 dB.

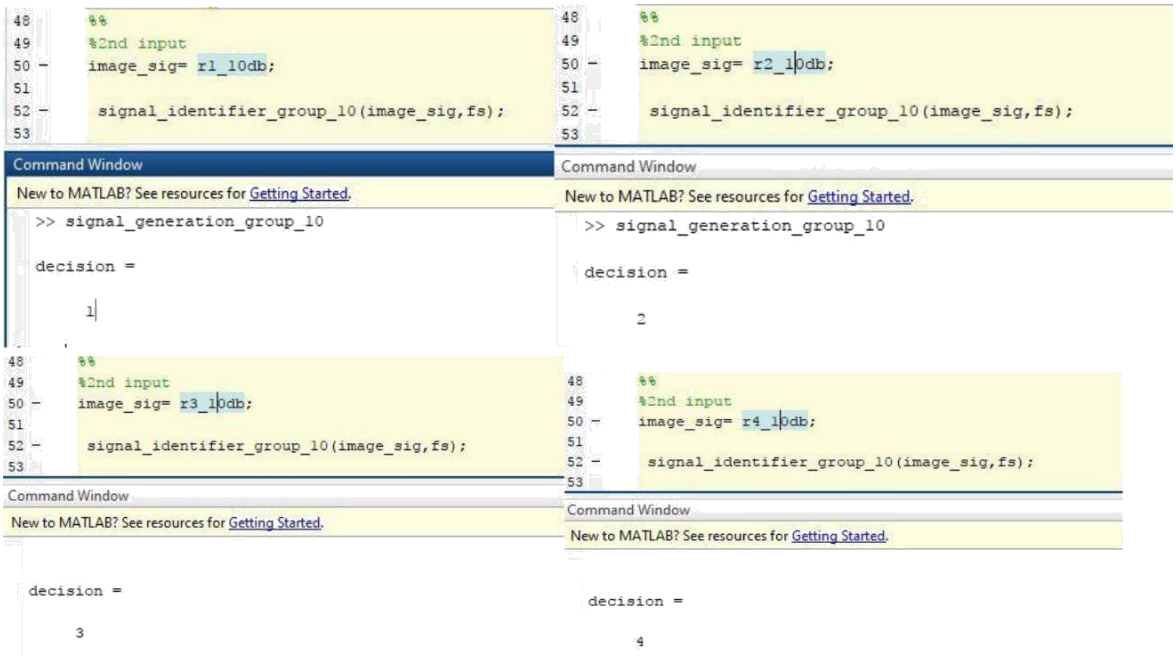

Figure 10. Screenshot of the MATLAB Interface after executing the function with input r1, r2, r3 and r4 under 10 dB.

Figure 11. Screenshot of the MATLAB Interface after executing the function with input r1, r2, r3 and r4 under 20 dB.

As shown in the figures, our identifier functions properly.

3rd Step:

In the second step, we are required to take the noisy and shifted CFO versions of s1,s2,s3,s4; which z1, z2, z3, z4 (there are 16 possible functions generated) in the Project Phase-1 Definition as inputs, as shown in Figure 12 and define them.

```
n=0:a-1;
z11=s1.*exp(-j.*pi/4.*n)+sqrt(1/2*1.008/1000).*randn(1,a)+j.*sqrt(1/2*1.008/1000).*randn(1,a);
z12=s1.*exp(-j.*pi/2.*n)+sqrt(1/2*1.008/1000).*randn(1,a)+j.*sqrt(1/2*1.008/1000).*randn(1,a);
z13=s1.*exp(j.*pi/4.*n)+sqrt(1/2*1.008).*randn(1,a)+j.*sqrt(1/2*1.008).*randn(1,a);
z14=s1.*exp(j.*pi/2.*n)+sqrt(1/2*1.008).*randn(1,a)+j.*sqrt(1/2*1.008).*randn(1,a);

z23=s2.*exp(j.*pi/4.*n)+sqrt(1/2*1.08).*randn(1,a)+j.*sqrt(1/2*1.08).*randn(1,a);
z21=s2.*exp(-j.*pi/4.*n)+sqrt(1/2*1.08/1000).*randn(1,a)+j.*sqrt(1/2*1.08/1000).*randn(1,a);
z24=s2.*exp(j.*pi/2.*n)+sqrt(1/2*1.08).*randn(1,a)+j.*sqrt(1/2*1.08).*randn(1,a);
z22=s2.*exp(-j.*pi/2.*n)+sqrt(1/2*1.08/1000).*randn(1,a)+j.*sqrt(1/2*1.08/1000).*randn(1,a);

z33=s3.*exp(j.*pi/4.*n)+sqrt(1/2*1).*randn(1,a)+j.*sqrt(1/2*1).*randn(1,a);
z34=s3.*exp(-j.*pi/4.*n)+sqrt(1/2*1/1000).*randn(1,a)+j.*sqrt(1/2*1/1000).*randn(1,a);
z31=s3.*exp(j.*pi/2.*n)+sqrt(1/2*1).*randn(1,a)+j.*sqrt(1/2*1.08).*randn(1,a);
z32=s3.*exp(-j.*pi/2.*n)+sqrt(1/2*1/1000).*randn(1,a)+j.*sqrt(1/2*1/1000).*randn(1,a);

z43=s4.*exp(j.*pi/4.*n)+sqrt(1/2*1).*randn(1,a)+j.*sqrt(1/2*1).*randn(1,a);
z44=s4.*exp(-j.*pi/4.*n)+sqrt(1/2*1/1000).*randn(1,a)+j.*sqrt(1/2*1/1000).*randn(1,a);
z41=s4.*exp(j.*pi/2.*n)+sqrt(1/2*1).*randn(1,a)+j.*sqrt(1/2*1.08).*randn(1,a);
z42=s4.*exp(-j.*pi/2.*n)+sqrt(1/2*1/1000).*randn(1,a)+j.*sqrt(1/2*1/1000).*randn(1,a);
```

Figure 12. Corresponding code to implement the signals with CFO and noise under 0 dB and 30 dB.

The main problem caused by the different CFO is the excessive sample shifting. This type of shifting makes bandwidth to be found harder, which is a problem in especially FM distinguishing stage. This problem can be solved by taking the difference of the sample length and the measured bandwidth when the bandwidth exceeds a big value.

The results of the 3rd step when fs=3.6Msamp/s are shown in Figure 13 to 16.

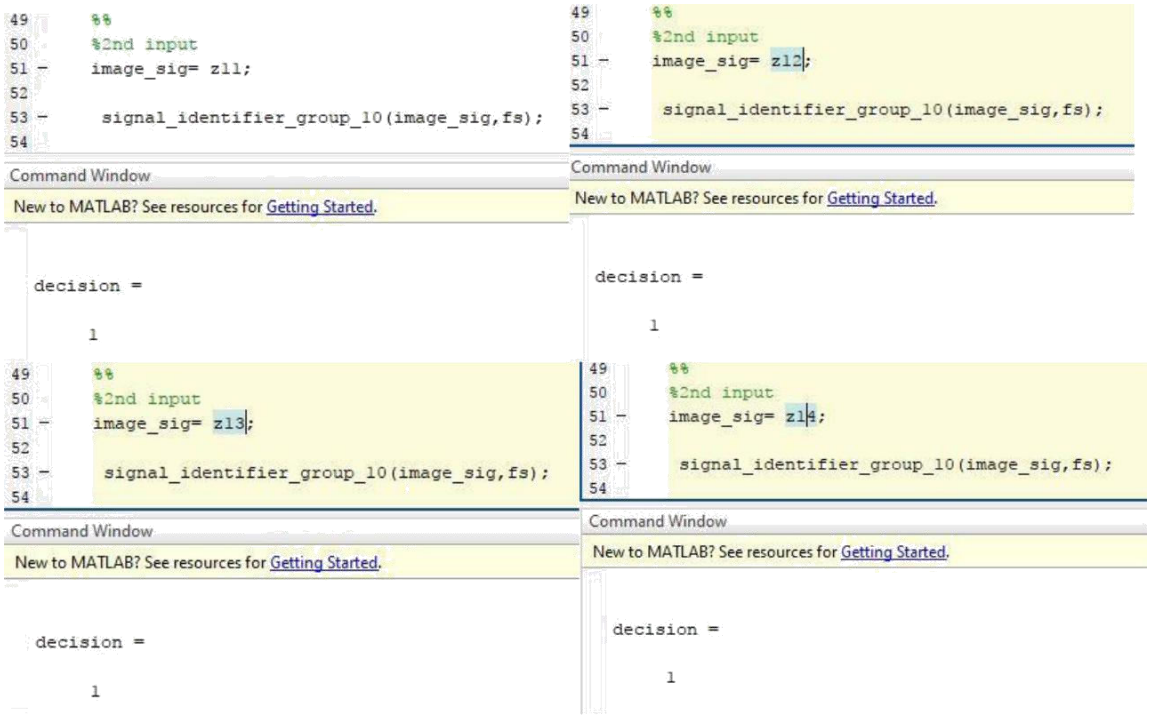

Figure 13. Screenshot of the MATLAB Interface after executing the function with the input z11, z12, z13 and z14.

Figure 14. Screenshot of the MATLAB Interface after executing the function with the input z21, z22, z23 and z24.

Figure 15. Screenshot of the MATLAB Interface after executing the function with the input z31, z32, z33 and z34.

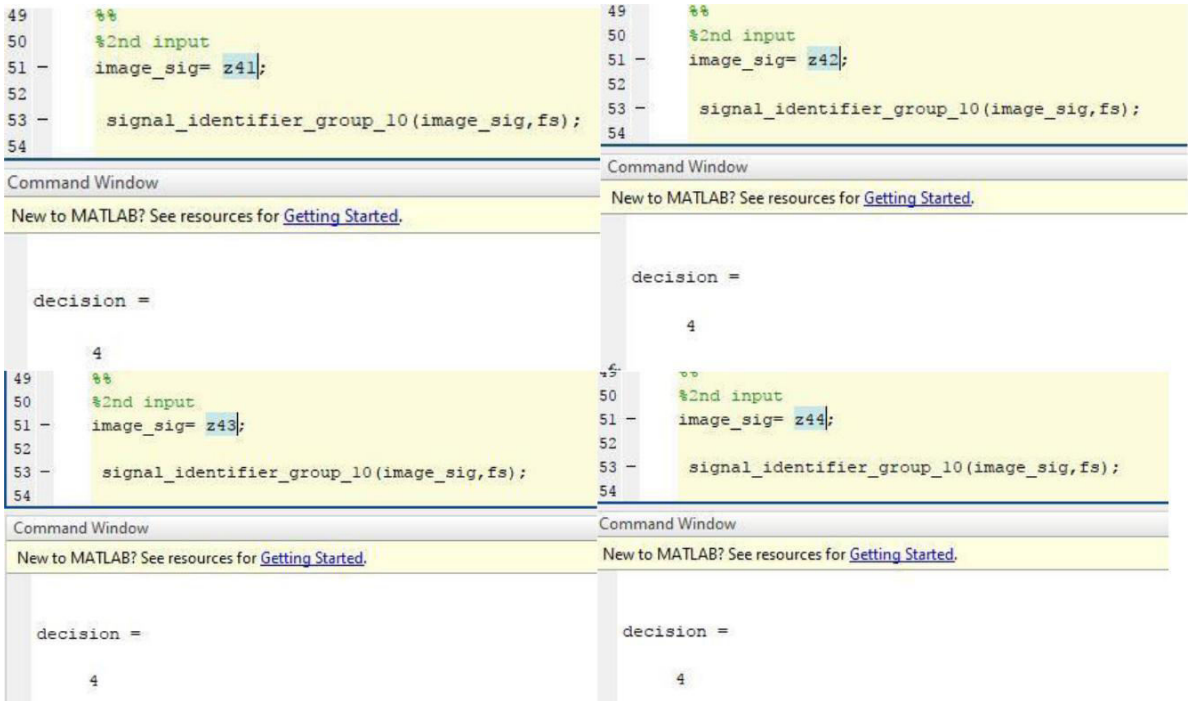

Figure 16. Screenshot of the MATLAB Interface after executing the function with the input z41, z42, z43 and z44.

As shown in the figures, our identifier functions properly.

Conclusion:

In the 1st phase of this project, we developed an algorithm to detect the type of input signal. At the first step, we tried to detect the input signal for ideal signals. Then, we tried to do the same process for the noisy input signals and the ones with noise and CFO. At each step, there were some incompatibilities occurring for each different type of signal.

During this phase, we had a great opportunity to observe the properties of AM & FM modulated signals. Especially, in the last part (the one with the CFO), we encountered some errors in the deciding part, but we solved these problems after some tries.

In the 2nd phase of the project, we will demonstrate this system with real signals, real transmission by using the RTL-SDR. So, it was important for us to observe all possible distortion problems that we will probably face in the 2nd stage.