BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Department of Electrical & Electronic Engineering

Course No : EEE 310

Course Name: Communication Laboratory

Project Title:

DESIGN ANACOM-1/1 & 1/2 BOARD AND IMPLEMENT DSB-WC & DSB-SC

Submitted to:

Dr. Md. Forkan Uddin, Professor, Dept. of EEE, BUET
Tashfiq Ahmed, Lecturer , Dept. of EEE, BUET

Submitted By

1706037 - Md. Al – Hasib

1706042 - Md. Monirul Islam

1706044 - Shuvro Chowdhury

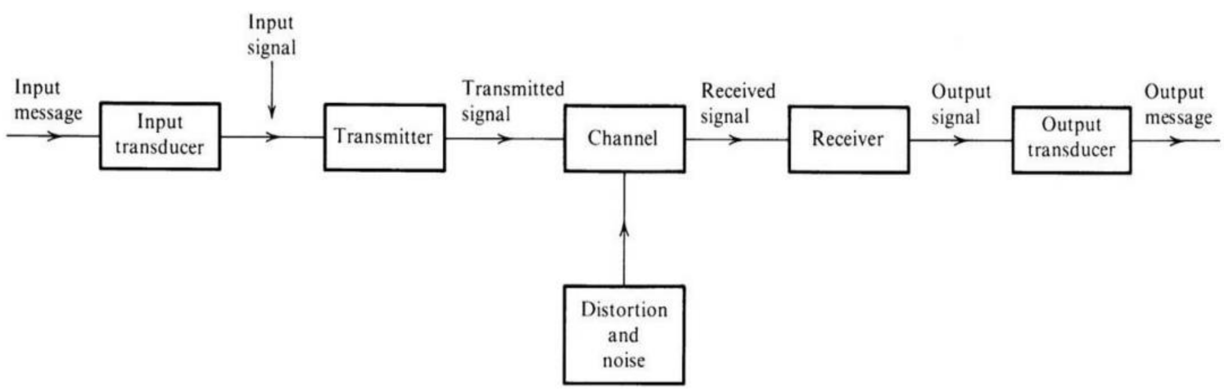
1706055 – Adib Md. Ridwan

Introduction:

An analog communication system is a communication system where the information signal sent from point A to point B can only be described as an analog signal. In practical life our baseband signal needs to transmit from one place to another place and for this purpose we need our Communication system. Generally, we modulate our intelligent signal with a high frequency carrier to transmit information effectively.

In communication system, the basic elements are :

- Input transducer
- Transmitter
- Channel
- Receiver
- Output transducer



In analog communication , we use many kind of modulation technique like amplitude modulation, Phase modulation and frequency modulation. Amplitude modulation (AM) is a modulation technique used in electronic communication, most commonly for transmitting messages with a radio wave. In AM , we modulate the amplitude of high frequency carrier signal with respect to the amplitude of the message signal .

Necessity of modulation:

a) **For practical Antenna length:** Low frequency transmission and reception is not practical due to large antennas required. Theory reveal s that “the length of the transmitting antenna should be approximately equal to tha wavelength of the transmitting wave.

b) **For Increasing Operating Range:** The energy of the wave depends on its frequency. The higher the frequency, the greater the energy. As audio signal frequencies are small they cannot the transmitted over long distance without modulation.

c) **For achieving Wireless Communication:** One desirable feature of radio transmission is that it should be carried without wires. At audio frequencies radiation is not practicable because the efficiency of radiation is poor.

d) **To Suit the Medium or Channel Requirement:** Modulation is necessary to meet the requirement of the medium.

ANACOM Board Description:

To demonstrate Amplitude modulation we generally use **Anacom 1/1 (Transmitter)** and **Anacom 1/2 (Receiver)** for the procedure.

The ANACOM 1/1 Board:

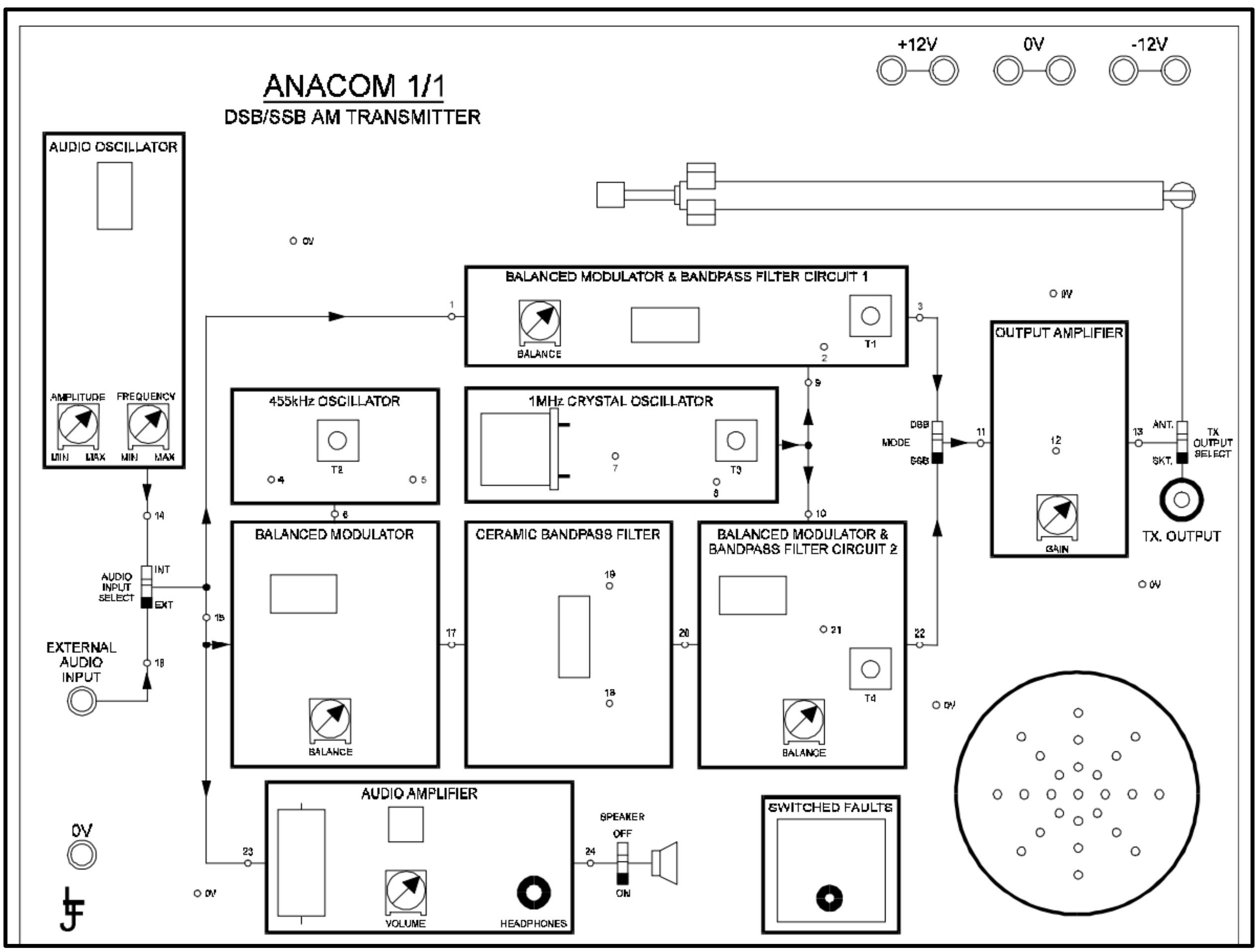


Fig: Layout Diagram of the ANACOM 1/1 Board

The ANACOM 1/1 Board Blocks:

The transmitter board can be considered as **five separate blocks**:

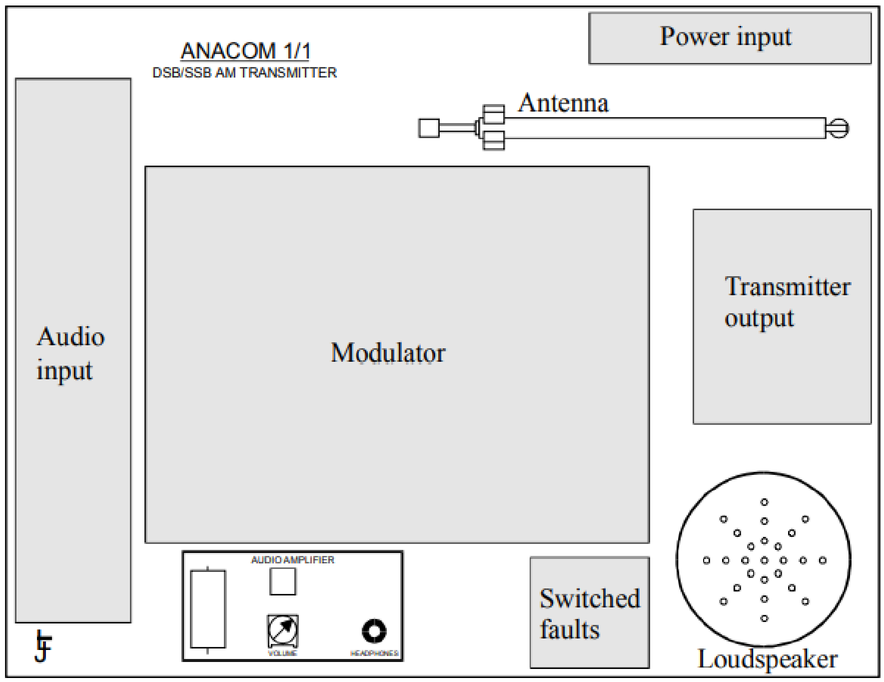


Figure 2

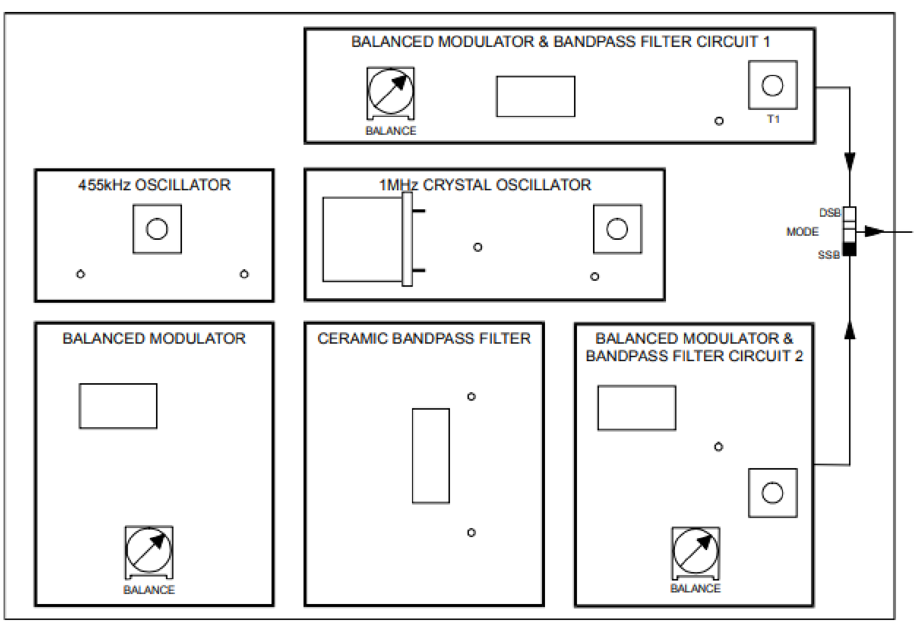


Figure 5

Modulator: This section of the board accepts the information signal and generates the final signal to be transmitted.

The ANACOM 1/2 Board:

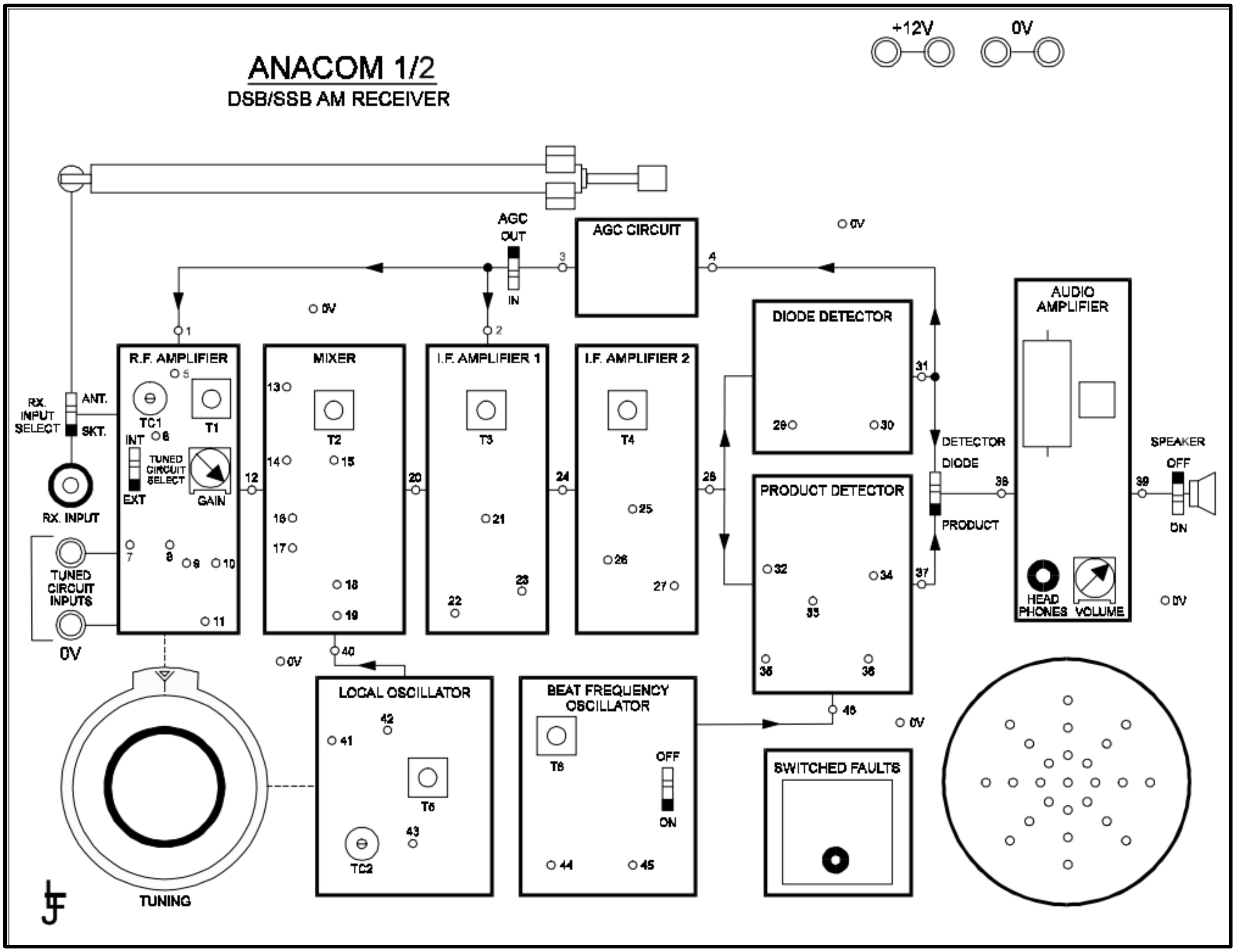


Fig: Layout Diagram of the ANACOM 1/2 Board

The ANACOM 1/2 Board Blocks:

The receiver board can also be considered as **five separate** blocks:

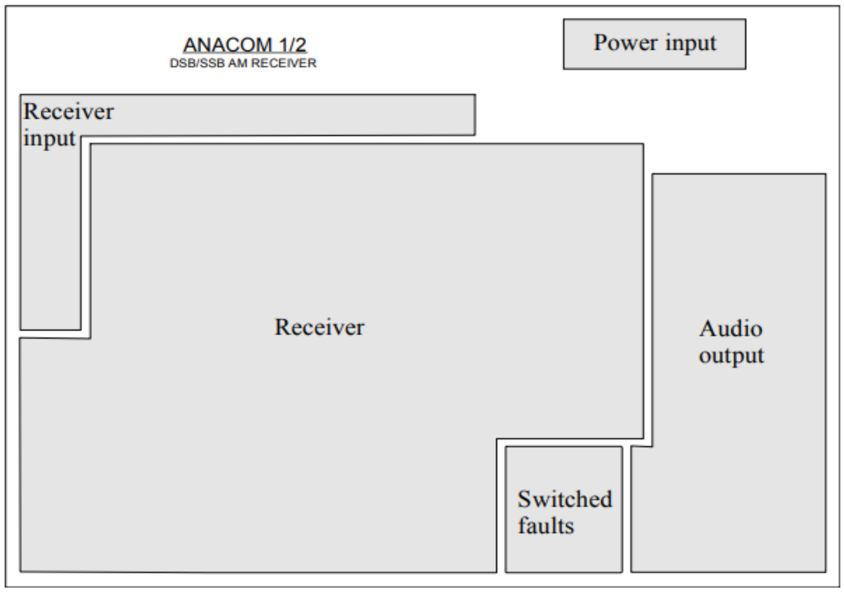


Figure 9

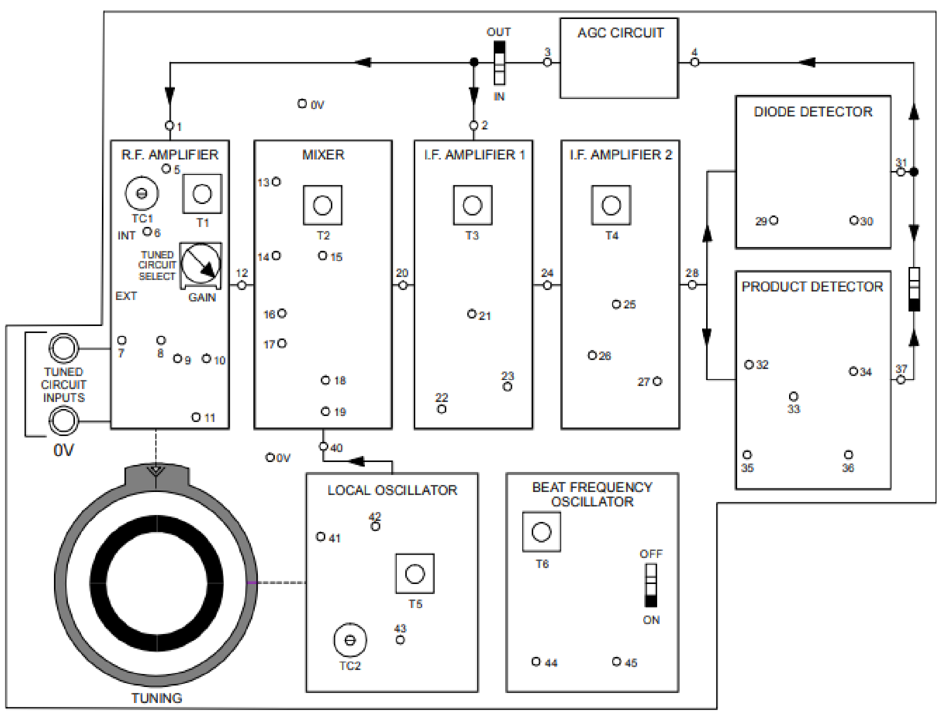


Figure 12

The Receiver: The receiver amplifies the incoming signal and extracts the original audio information signal. The incoming signals can be AM broadcast signals or those originating from ANACOM 1/1.

Methodology:

Step 1: When we start our program, the window will look like this:

Step-2: There are respective buttons for the DSB_WC and DSB_SC . If we click on DSB_WC a new window will pop up like this:

Step-3: Then by selecting ANACOM 1/1 and ANACOM 1/2 we will see this kind of figures respectively and by entering our data like A_m, f_m, A_c, f_c gain etc we can click simulate button to perform the simulation.

Then by selecting plot button we can observe the time domain and frequency domain graphs of respective section.

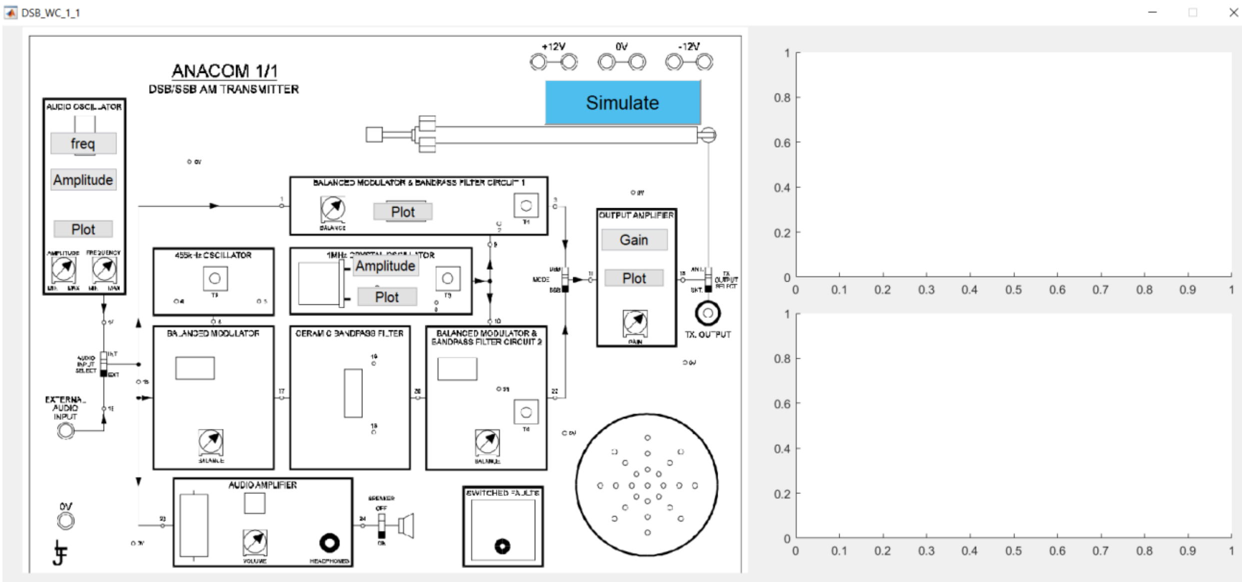


Fig: Anacom 1/1 GUI diagram for DSB_WC

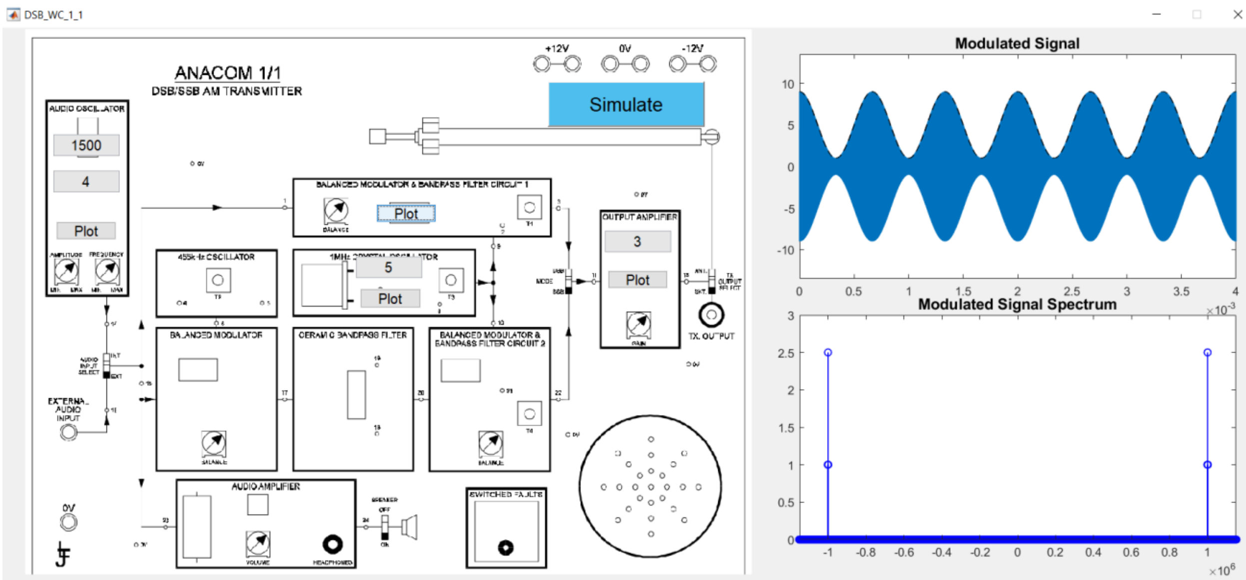


Fig: Anacom 1/1 GUI diagram for DSB_WC(with a sample modulated signal)

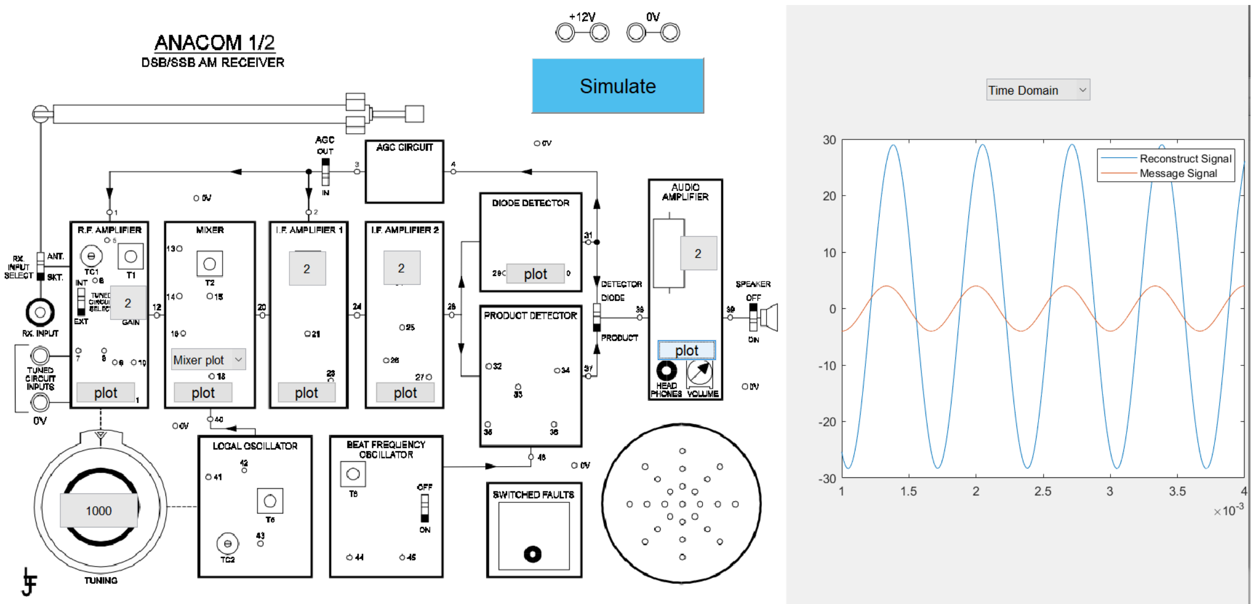


Fig: Anacom 1/2 GUI diagram for DSB_WC(with a sample output signal)

And for the DSB_SC , the Anacom boards will look like this.

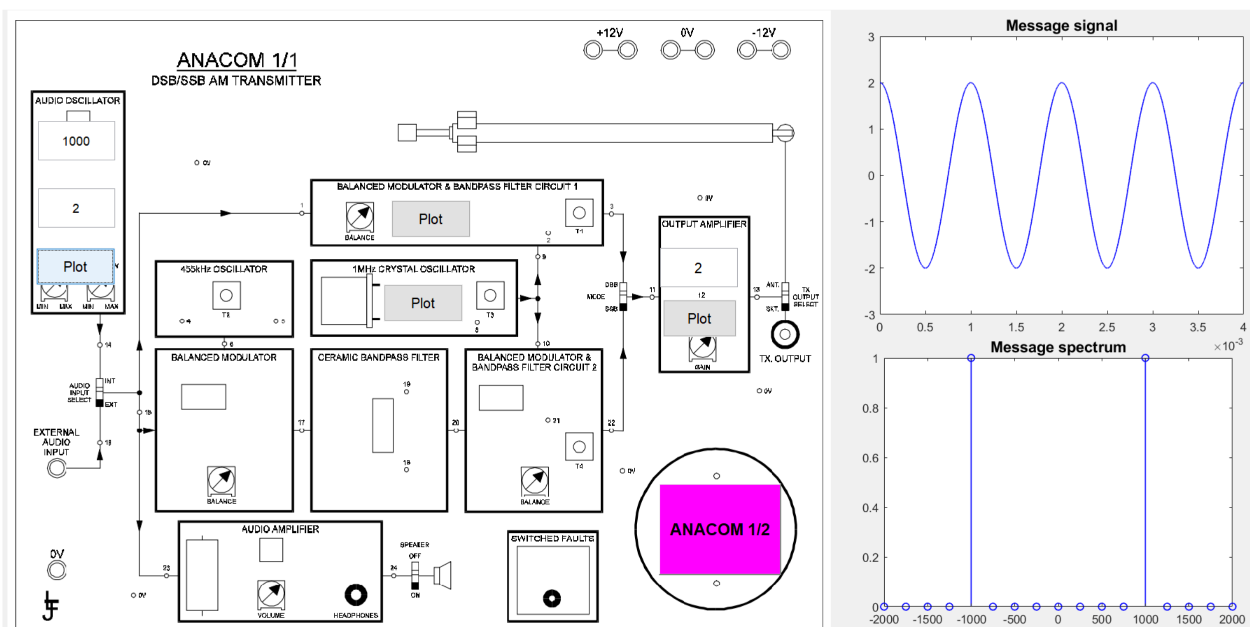


Fig: Anacom 1/1 GUI diagram for DSB_WC(with a sample message signal)

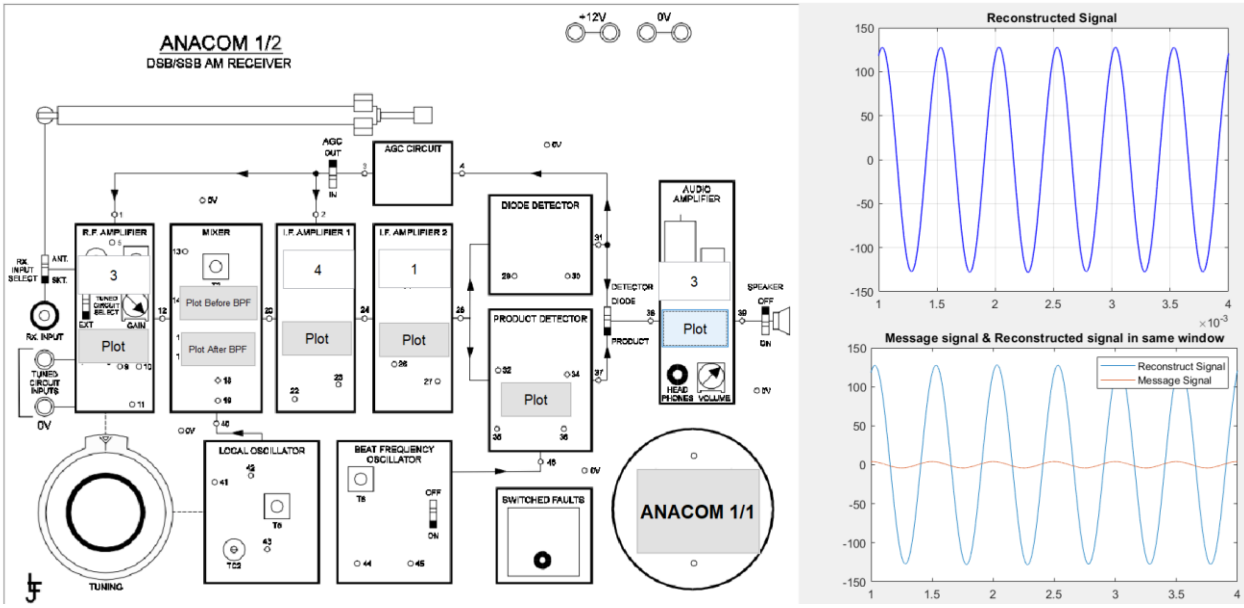


Fig: Anacom 1/2 GUI diagram for DSB_WC(with a sample output signal)

Double Sideband With Carrier (DSB WC):

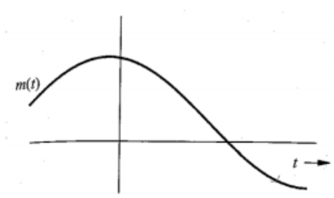
In DSB WC, double side band(DSB) means we will transmit both the USB and LSB .In DSB WC , we transmit an unmodulated carrier in addition to the modulated carrier . Some properties Of DSB WC :

- Wasteful of transmitted power: power efficiency very low.
- Wasteful of channel bandwidth: twice of the message bandwidth .
- Easy to be affected by noise .
- Simpler modulator and demodulator .
- Less expensive modulator and demodulator.

Theory:

The Carrier Signal: $c(t) = A_c \cos \omega_c t = A_c \cos 2\pi f_c t$, f_c = Carrier frequency (Hz)

The baseband message (modulating) signal: $m(t)$



The AM signal: $\varphi_{AM}(t) = [A_c + m(t)]\cos \omega_c t = A_c \cos \omega_c t \quad + \quad m(t)\cos \omega_c t$

(Carrier) (Modulated carrier)

$$= A_c\left[1 + \frac{m(t)}{A_c}\right]\cos \omega_c t = A_c[1 + k_a m(t)]\cos \omega_c t = [1 + k_a m(t)]c(t)$$

Procedure:

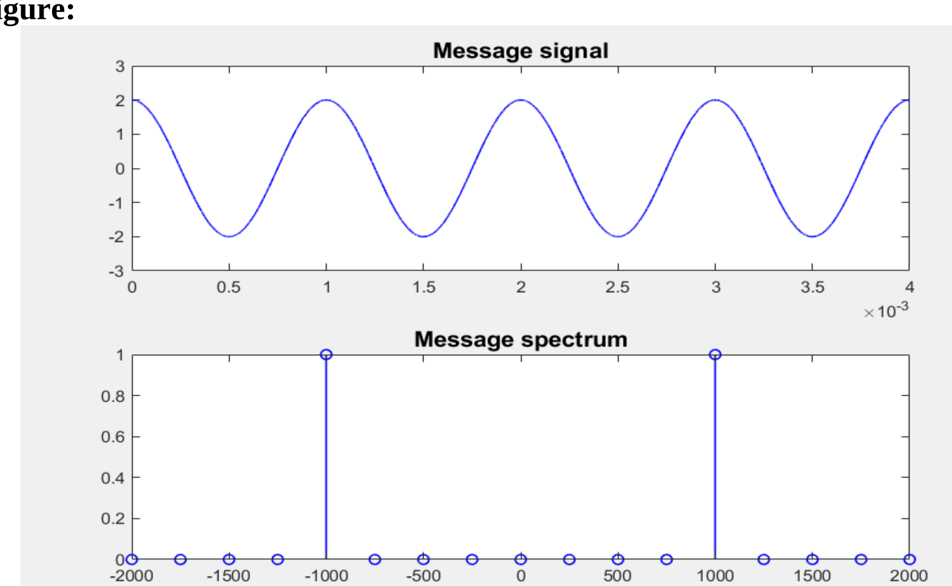
(1) Audio Oscillator:

This circuit provides an internally generated signal that is going to be used as 'information' to demonstrate the operation of the transmitter. There is also an External Audio Input facility to enable us to supply our own audio information signals. The information signal can be monitored, if required, by switching on the loudspeaker. An amplifier is included to boost the signal power to the loudspeaker. We will use the internally generated Signal as out message signal.

Matlab Code to Produce messeage signal :

```
%% Audio Oscillator

fm = 1000;        %Take as input ---> m(t) frequency
Am = 2;           %Take as input ---> m(t) amplitude

m_t = Am*cos(2*pi*fm*t);                  %message signal
M_F = abs(fftshift(fft(m_t)))/length(m_t);

figure(1);
subplot(211)
plot(t,m_t,'b','LineWidth',1);
title('Message signal','fontweight','bold','Fontsize',13);
ylim([1.5*min(m_t), 1.5*max(m_t)]);

subplot(212)
stem(fn,M_F,'b','LineWidth',1);
title('Message spectrum','fontweight','bold','Fontsize',13);
xlim([-2*fm, 2*fm]);
```

Output Figure:



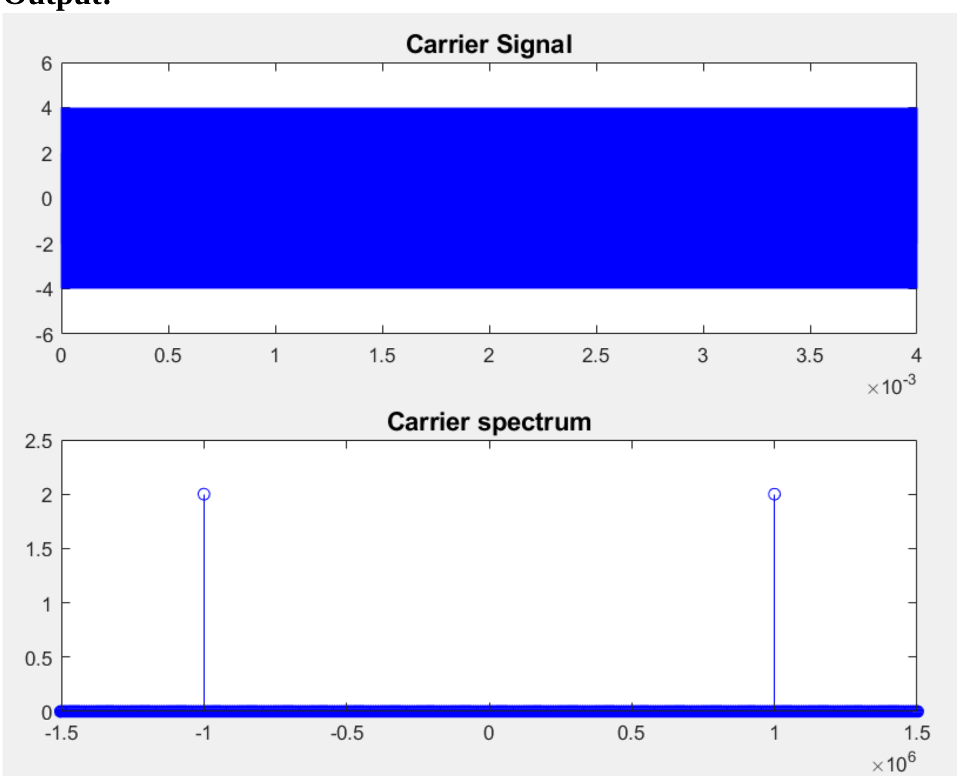
(2) 1MHz CRYSTAL OSCILLATOR:

Anacom has a built in oscillator to produce 1MHz signal, the Amplitude can be changed, but the frequency is Constant 1MHz.

To replicate this Model, Our Code Is,

```
%% 1MHz Crystal Oscillator
Ac =4; % Take as input ---> c(t) amplitude
fc = 1000e3;        %carrier frequency, ANACOM 1 - 1MHz fixed

c_t = Ac*cos(2*pi*fc*t);                    %carrier signal
C_F = abs(fftshift(fft(c_t)))/length(c_t);

figure(2);
subplot(211)
plot(t,c_t,'b');
title('Carrier Signal','fontweight','bold','Fontsize',13);
ylim([1.5*min(c_t), 1.5*max(c_t)]);
subplot(212)
stem(fn,C_F,'b');
title('Carrier spectrum','fontweight','bold','Fontsize',13);
xlim([-1.5*fc, 1.5*fc]);
```

Output:



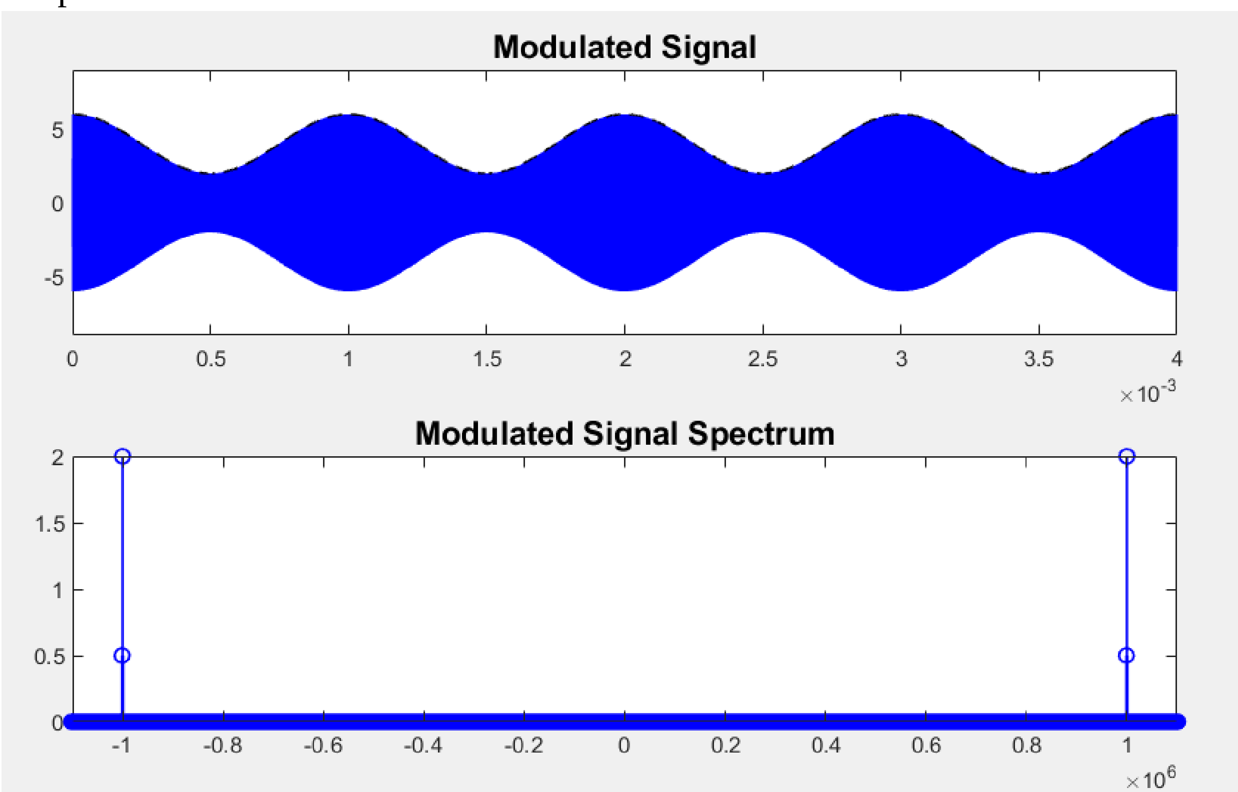
The freq spectrum shows the signal at 1 MHz (As expected)

(3) Balanced modulator:

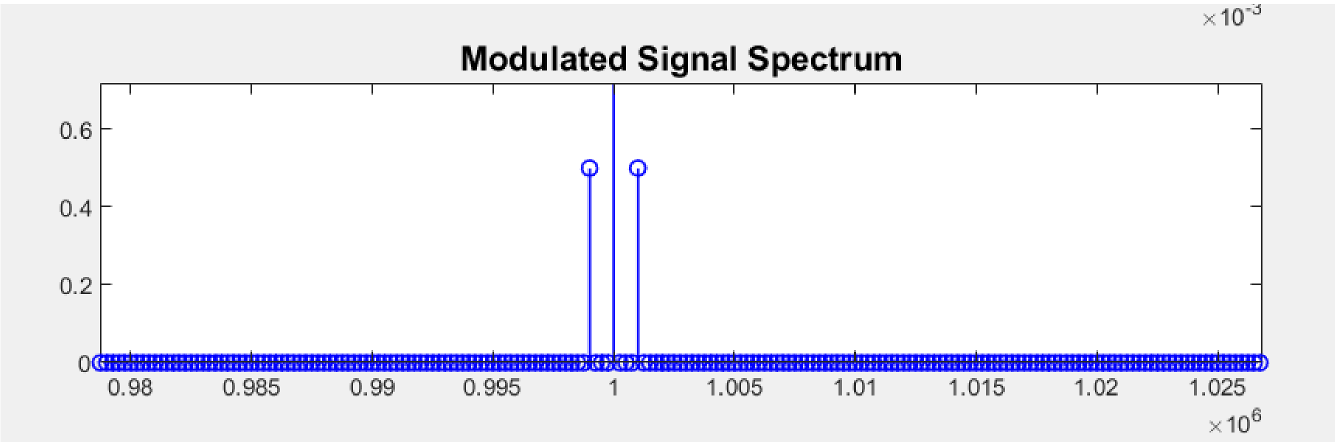
This section mainly does the modulation. Here , we modulate the amplitude of our high frequency carrier signal with respect to the amplitude of the message signal for better transmission. To imitate this section , our Code is:

```
s_t = (Ac + m_t).*cos(2*pi*fc*t);
S_F = abs(fftshift(fft(s_t)))/length(s_t);

figure(3);
subplot(211)
plot(t,s_t,'b');
hold on
en_s_t=envelope(s_t);
plot(t,en_s_t,'k--','LineWidth',1);
title('Modulated Signal','fontweight','bold','Fontsize',13);
ylim([1.5*min(s_t), 1.5*max(s_t)]);

subplot(212)
stem(fn,S_F,'b','LineWidth',1);
title('Modulated Signal Spectrum','fontweight','bold','Fontsize',13);
xlim([-fc - 100*fm, fc + 100*fm]);
```

Output:



If we zoom the Modulated Signal spectrum, we can easily observe that modulated signal has 3 freq component at f_c, f_c+f_m and f_c-f_m



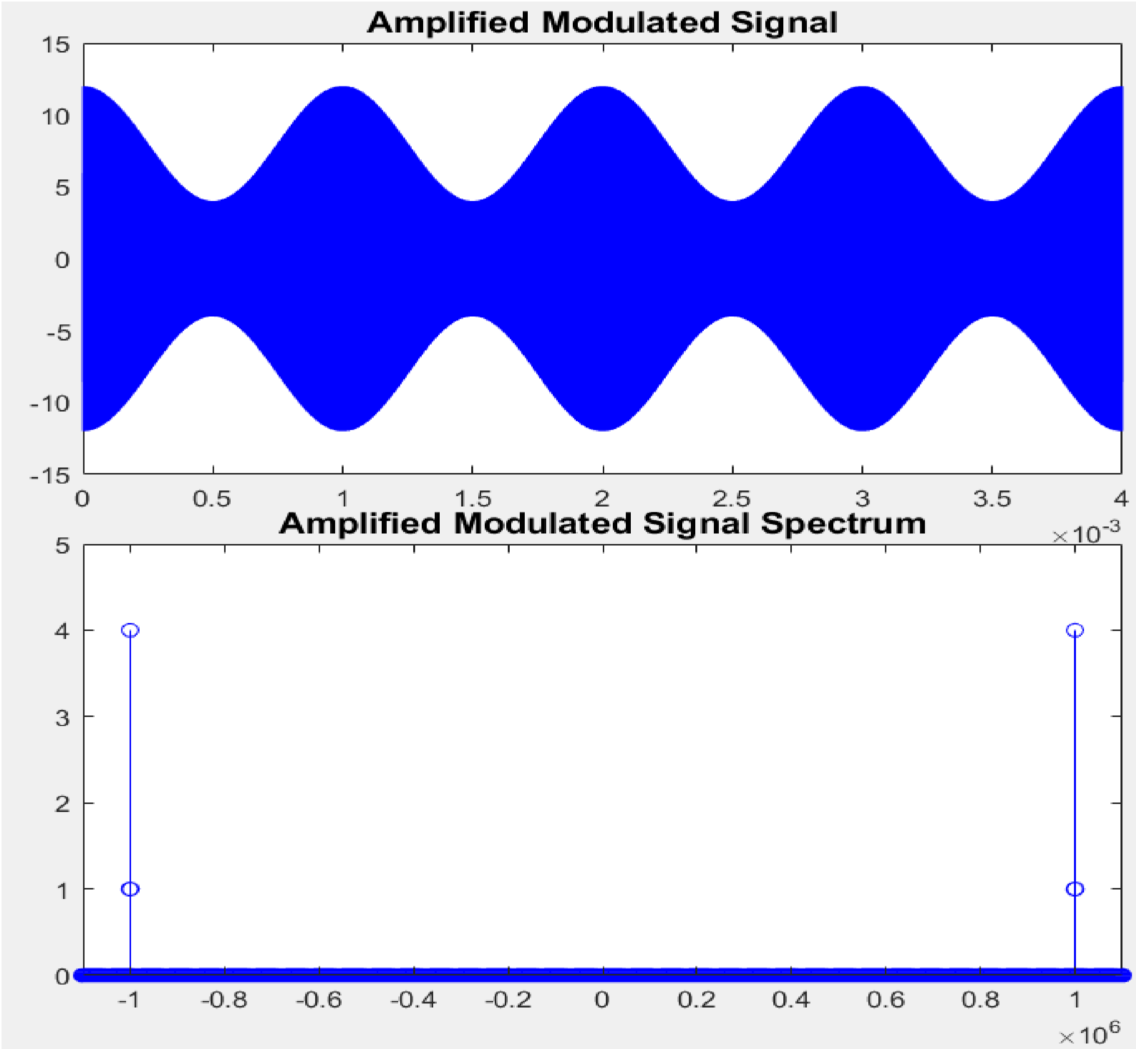
(4) Output Amplifier:

Before Transmitting, we can amplify the signal and then transmit this as for the wireless transmission attenuation is very common phenomena. But after amplification, freq component will not change , only there amplitude will vary.

Our Code:

```
%% Output Amplifier
Amp_1=2; % "must" take as input vary
s_t_Amp=Amp_1*s_t;
S_F_Amp = abs(fftshift(fft(s_t_Amp)))/length(s_t_Amp);

figure(4);
subplot(211)
plot(t,s_t_Amp,'b');
title('Amplified Modulated Signal','fontweight','bold','Fontsize',13);
ylim([1.5*min(s_t), 1.5*max(s_t)]);

subplot(212)
stem(fn,S_F_Amp,'b');
title('Amplified Modulated Signal
Spectrum','fontweight','bold','Fontsize',13);
xlim([-fc - 100*fm, fc + 100*fm]);
```

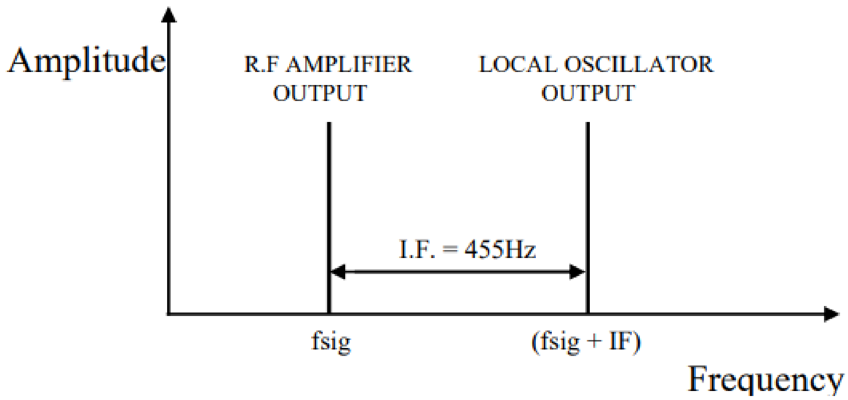
Plot:



Demodulator Part:

(1) Local Oscillator

In the demodulator block at first, we need to shift the cattier frequency of the received signal to 455kHz. To do that we need to mix the received signal with a frequency of (455kHz + fsig) where the fsig is the frequency of the carrier.



Our code:

```
%% Local Oscillator
IR=455e3;
f_local = IR + fc ; %local oscillator frequency 455kHz + fc
local_t = cos(2*pi*f_local*t);
```

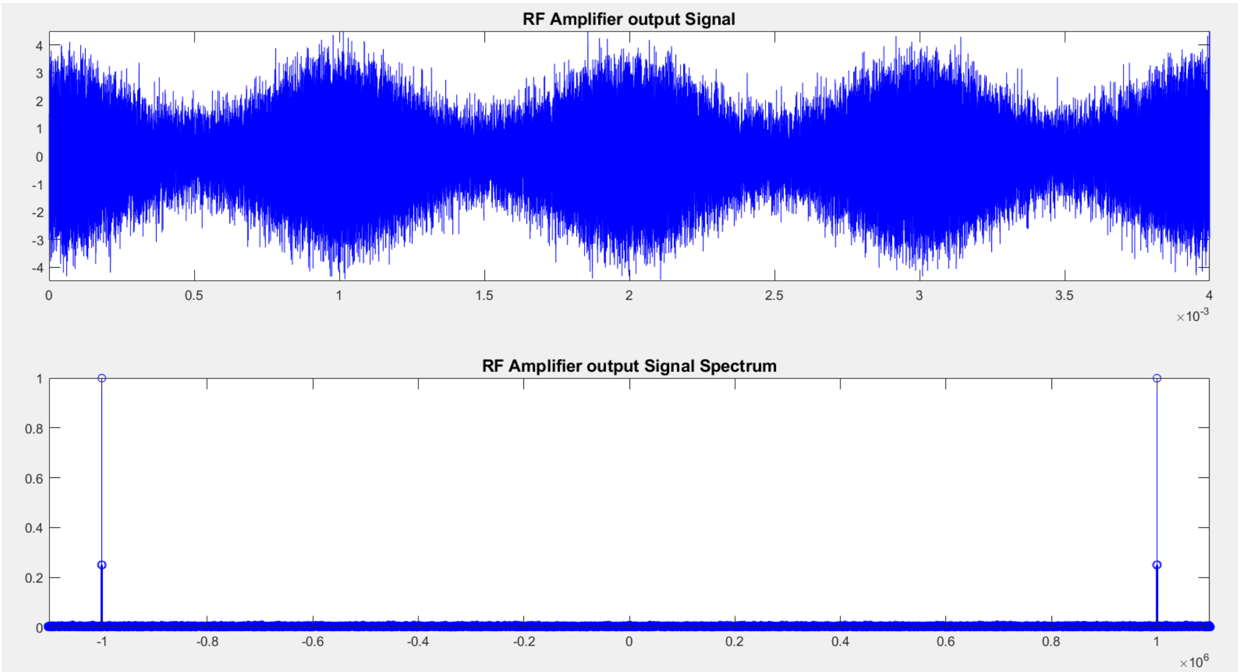
(2) RF Amplifier:

Before we mix the received signal, we need to amplify the signal so that signal has fairly large amplitude. However this process does not affect the frequency spectrum of the signal.

Our Code:

```
%% RF Amplifier
Amp_2=2;
s_t_rf=Amp_2*s_t_Amp;
S_F_rf = abs(fftshift(fft(s_t_rf)))/length(s_t_rf);

figure(5);
subplot(211)
plot(t,s_t_rf,'b');
title('RF Amplifier output Signal','fontweight','bold','Fontsize',13);
ylim([1.5*min(s_t), 1.5*max(s_t)]);

subplot(212)
stem(fn,S_F_rf,'b');
title('RF Amplifier output Signal
Spectrum','fontweight','bold','Fontsize',13);
xlim([-fc - 100*fm, fc + 100*fm]);
```

Output:



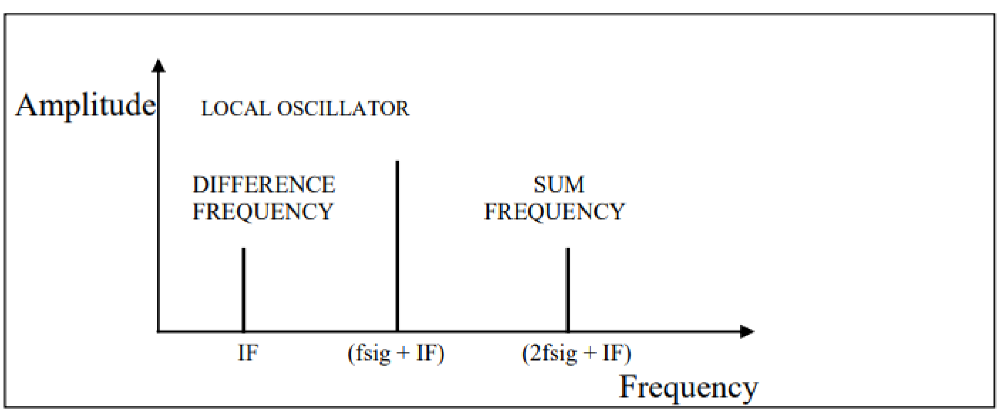
Here we can see some distortion which is due to noise in the transmission media added to the transmitted signal.

(3) Mixer:

In the mixer block we mix the received signal with the local oscillator frequency which is tuned so that the result of the process will contain a signal at 455kHz.

The local oscillator frequency =(fsig +IF)
The sum of the original two frequencies, fsum =(2fsig + IF)
The difference between the original two frequencies fdiff = (fsig + IF – fsig) = IF

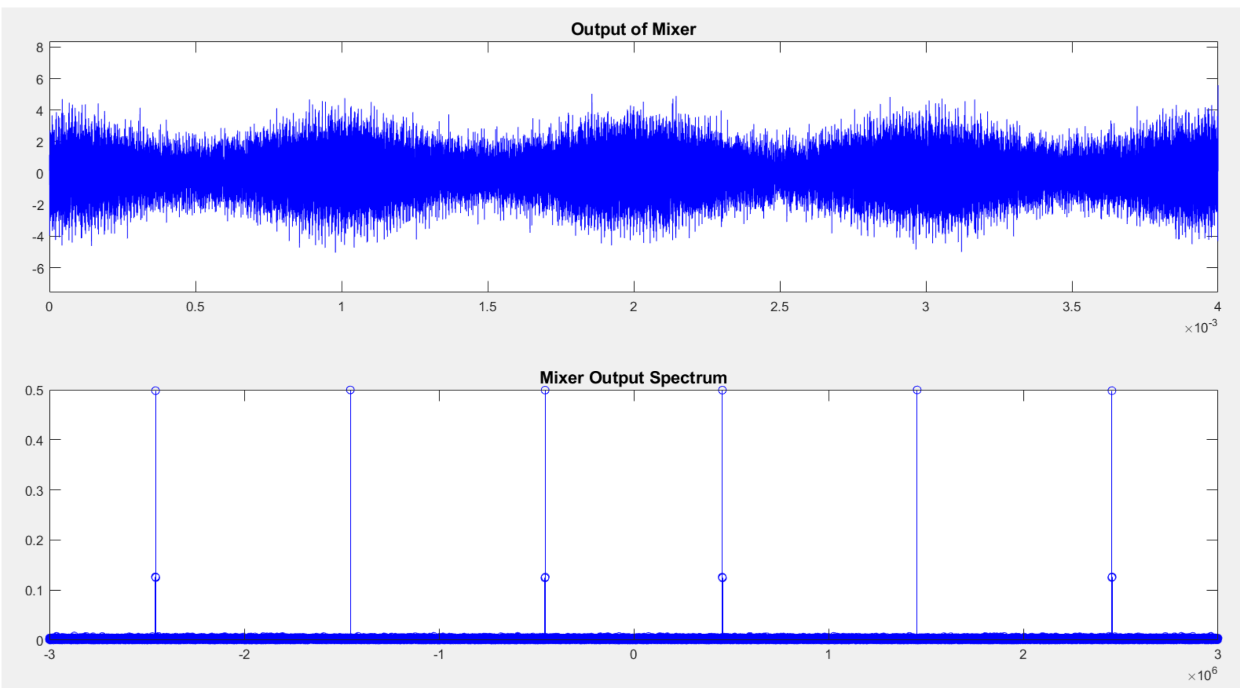
These three frequency components are shown below :

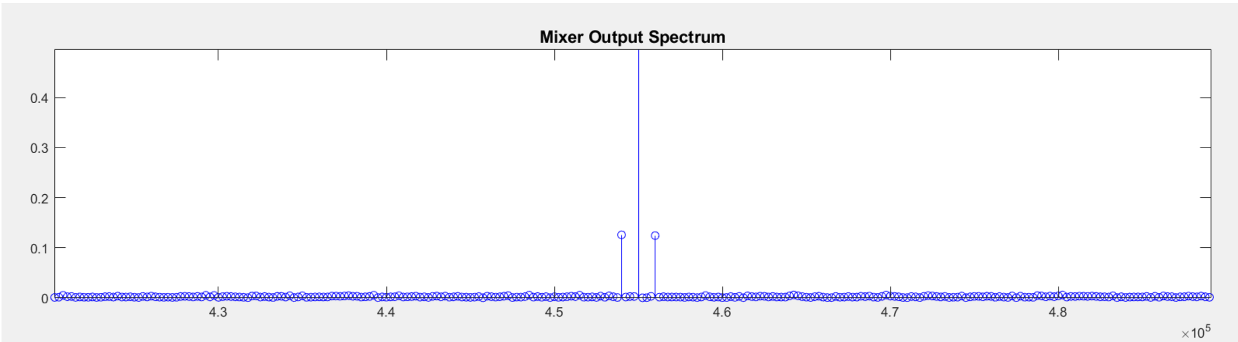


Our code:

```
%% Mixer
mixer_t =(1+ s_t_rf).* local_t;

Mixer_F = abs(fftshift(fft(mixer_t)))/length(mixer_t);

figure(6);
subplot(211)
plot(t,mixer_t,'b');
title('Output of Mixer','fontweight','bold','Fontsize',13);
ylim([1.5*min(mixer_t), 1.5*max(mixer_t)]);

subplot(212)
stem(fn,Mixer_F,'b');
title('Mixer Output Spectrum','fontweight','bold','Fontsize',13);
```

Output:





Here we can see that we have a frequency component at 455kHz.

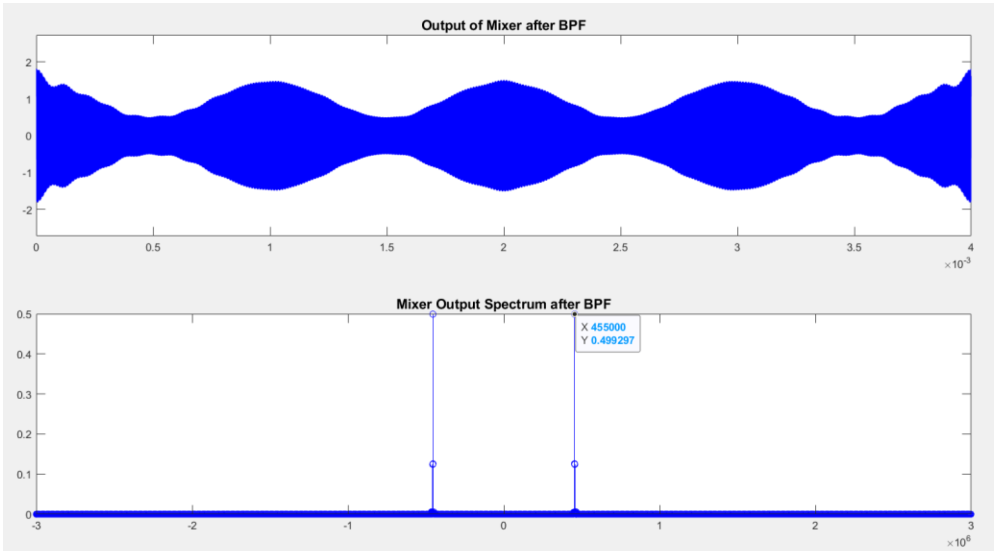
(4) Band pass Filter

In order to remove all other frequency component other than the component at 455kHz the band pass filter inside the mixer block of the anacom ½ block was used. This process removes all unnecessary frequency component of mixer output for further processing the signal in next steps.

Our code:

```
%% After Bandpass filter,
bpf=zeros(1,length(fn));
bpf(find(abs(fn)<465e3 & abs(fn)>445e3))=1;
Mixer_F_bpf=bpf.*Mixer_F;

mixer_t_bpf= ifft(abs(fftshift(Mixer_F_bpf))).*length(Mixer_F_bpf);

figure(7);
subplot(211)
plot(t,mixer_t_bpf,'b');
title('Output of Mixer after BPF','fontweight','bold','Fontsize',13);
ylim([1.5*min(mixer_t_bpf), 1.5*max(mixer_t_bpf)]);

subplot(212)
stem(fn,Mixer_F_bpf,'b');
title('Mixer Output Spectrum after BPF','fontweight','bold','Fontsize',13);
```

Results:

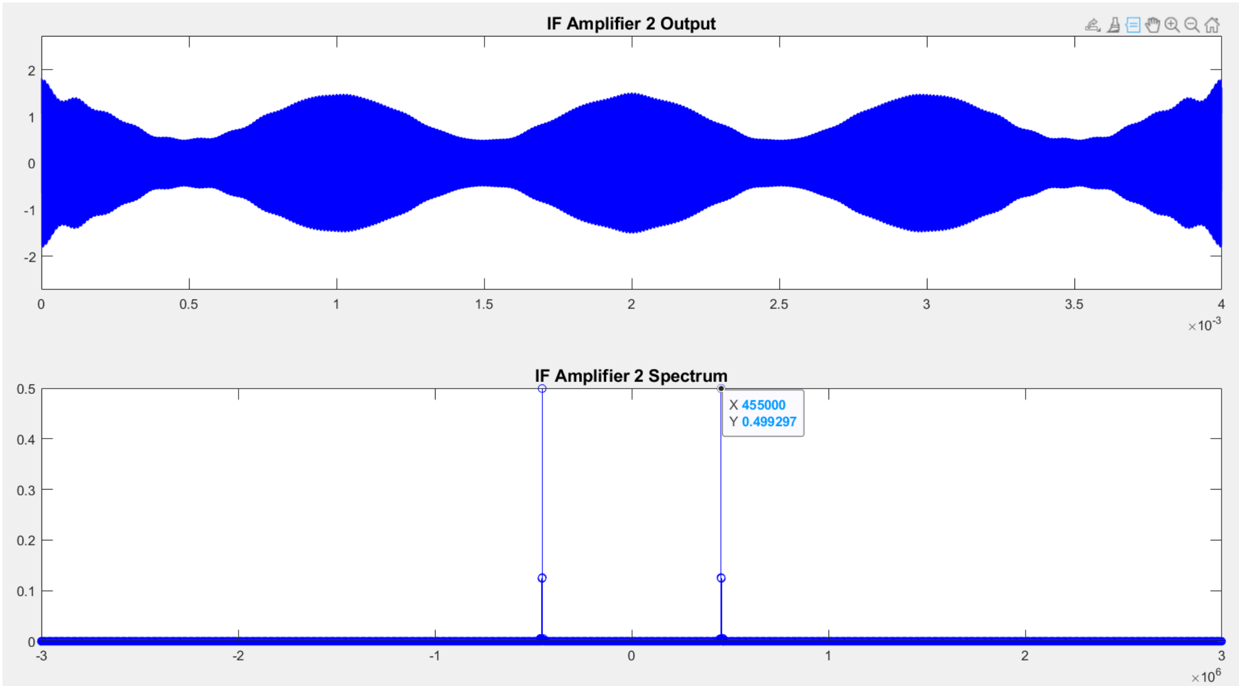


(5) Amplification of the filtered signal

After filtering the amplitude of the signal is reduced. To continue the signal processing we need to amplify the signal. Here we used two stages of amplification so that the signal does not cross into the non-linear region of the amplifier.

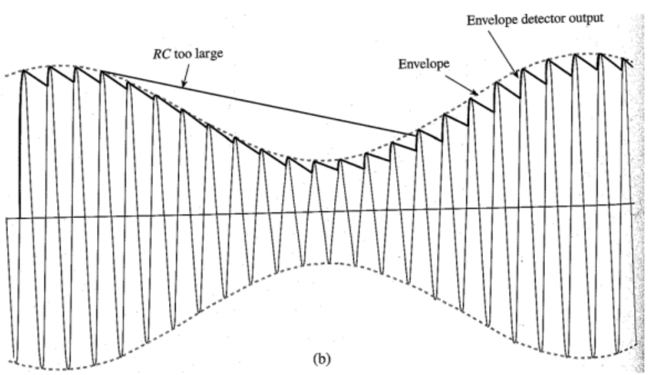
Our code:

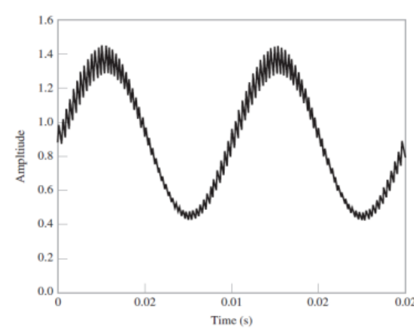
```
%% IF Amplifier 1

Amp_3=1; %"must" take as input
Amp_mixer_t = Amp_3*mixer_t_bpf;

Amp_mixer_F = abs(fftshift(fft(Amp_mixer_t )))/length(Amp_mixer_t);

figure(8);
subplot(211)
plot(t,Amp_mixer_t,'b');
title('IF Amplifier 1 Output','fontweight','bold','Fontsize',13);
ylim([1.5*min(Amp_mixer_t), 1.5*max(Amp_mixer_t)]);
subplot(212)
stem(fn,Amp_mixer_F,'b');
title('IF Amplifier 1 Spectrum','fontweight','bold','Fontsize',13);

%% IF Amplifier 2

Amp_4=1; %"must" take as input
Amp_filter_t_2 = Amp_4*Amp_mixer_t;
FILTER_F_2 = abs(fftshift(fft(Amp_filter_t_2)))/length(Amp_filter_t_2 );

figure(9);
subplot(211)
plot(t,Amp_filter_t_2,'b');
title('IF Amplifier 2 Output','fontweight','bold','Fontsize',13);
ylim([1.5*min(Amp_filter_t_2), 1.5*max(Amp_filter_t_2)]);
subplot(212)
```



(6) Diode Detector:

Diode detector is a type of envelope detector which detects the positive side of the envelope of the amplified signal.



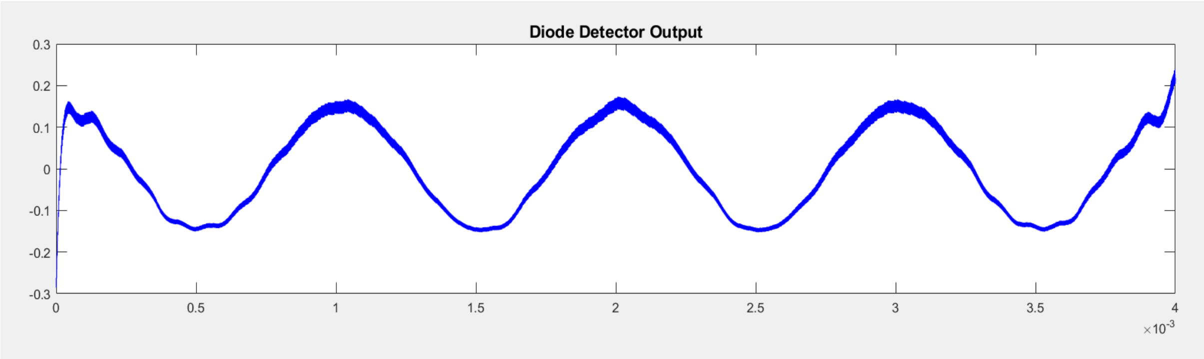


The envelope-detector output is $v_C(t) = A + m(t)$ with a ripple of frequency ω_c. The dc term A can be blocked out by a capacitor or a simple RC high-pass filter. The ripple may be reduced further by another (low-pass) RC filter.

Our Code:

```
%% Diode Detector
%changeable parameters
Rd = 20;              %Diode Resistance
R = 1000;             %Resistance
C = 1e-6;             %Capacitance
Vc = 0;               %Capacitor Voltage
detector_out = zeros(1,N);
for index = 1:N
    if Amp_filter_t_2(1,index)<0
        Amp_filter_t_2(1,index)=0;
    else
        continue
    end
end

%simulating the diode detector
for index = 1:N
    V = Amp_filter_t_2(1,index);
    if  V > Vc
        Vc = Vc + 1/Fs*(V-Vc)/(Rd*C);
    elseif V < Vc
        Vc = Vc - 1/Fs*Vc/(Rd*C);
    end
    detector_out(1,index) = Vc;
end

reconstructed_signal=detector_out-mean(detector_out); % DC block
Re_F = abs(fftshift(fft(reconstructed_signal)))/length(reconstructed_signal);

figure(10);
subplot(211)
plot(t,reconstructed_signal,'b');
title('Diode Detector Output','fontweight','bold','Fontsize',13);
%xlim([0.0001, max(t)]);
subplot(212)
stem(fn,Re_F,'b','Linewidth',2);
title('Diode Detector Spectrum','fontweight','bold','Fontsize',13);
xlim([-5*fm, 5*fm]);
```

Result:



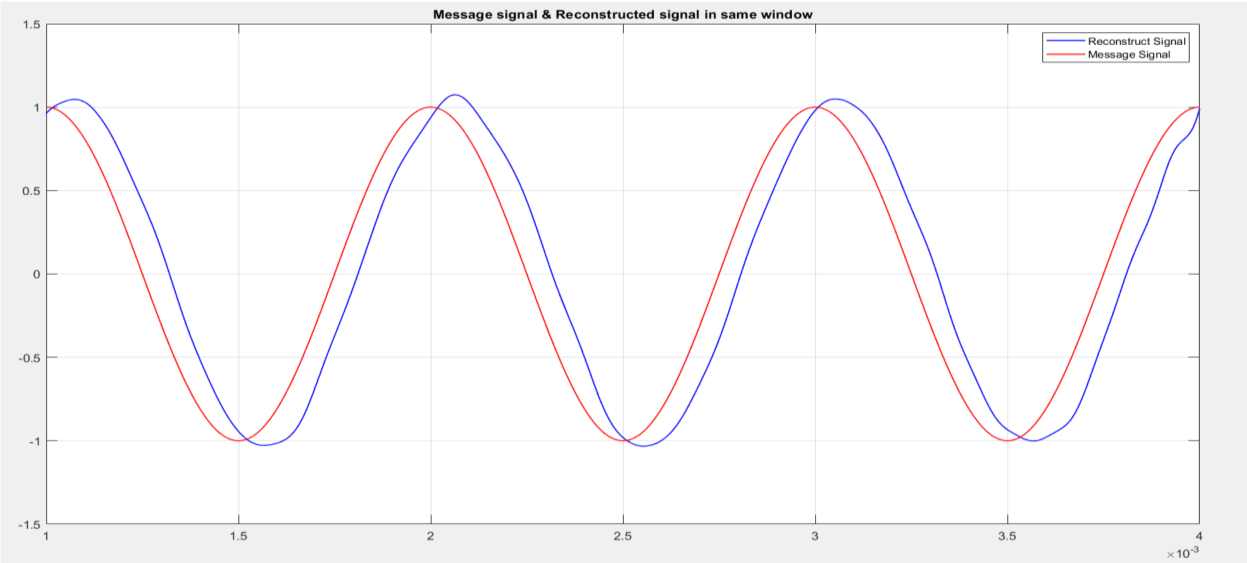
(7) Output Amplifier:

In the last part of the demodulation process we need to remove all the high frequency components of the output of the diode detector. To do that we used the Butterworth Low pass Filter. Finally we amplified the signal and displayed the reconstructed signal and message signal at the same window.

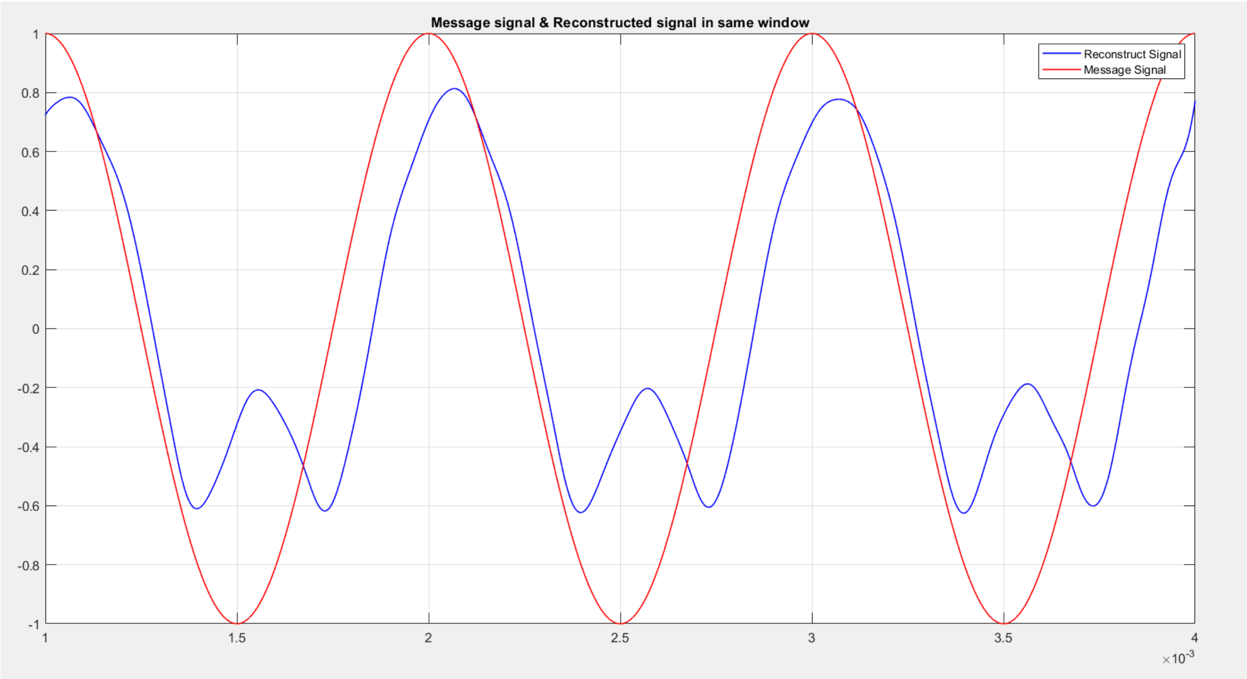
Code:

```
%% Audio Amplifier Output

lpf=5*fm;
[b,a] = butter(2,lpf/(Fs/2));                  %LowPass filter

final_signal = filter(b,a,reconstructed_signal);

Ampli=7; %must take as input
Amp_final_signal = Ampli*final_signal;

figure(11);
plot(t,Amp_final_signal,'b', 'linewidth',1);
hold on;
plot(t,m_t,'r', 'linewidth',1);
xlim([t_s, cycle*t_s]);
legend('Reconstruct Signal','Message Signal');
grid on
title('Message signal & Reconstructed signal in same
window','fontweight','bold');
```

Results:



Special case: Over modulation



Here, Ac > Am and dc offset is changed.

Double Sideband with Suppressed Career:

In DSB-SC only the upper and lower side bands are transmitted. A DSB-SC signal can be obtained by multiplying the message signal m(t) with the carrier signal c(t).

$$S(t)= m(t)\cos(2\Pi f_c t)$$

The modulated wave undergoes a phase reversal whenever the message signal crosses zero.

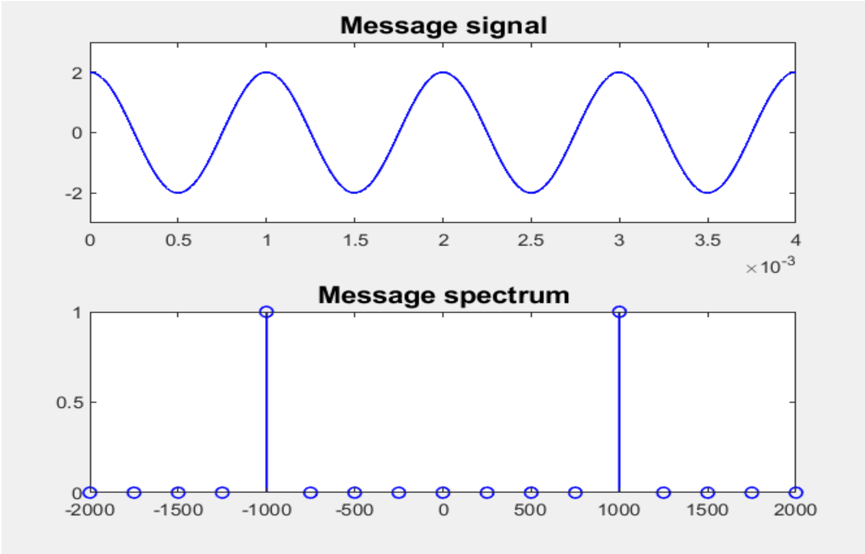
1) Audio Oscillator:

This circuit provides an internally generated signal that is going to be used as 'information' to demonstrate the operation of the transmitter. There is also an External Audio Input facility to enable us to supply our own audio information signals. The information signal can be monitored, if required, by switching on the loudspeaker. An amplifier is included to boost the signal power to the loudspeaker. We will use the internally generated Signal as our message signal.

Matlab code to produce message signal m(t) and its frequency spectrum M(f):

```
%% Audio Oscillator
fm = 1000;      %Take as input ---> m(t) frequency
Am = 2;         %Take as input ---> m(t) amplitude


m_t = Am*cos(2*pi*fm*t);              %message signal
M_F = abs(fftshift(fft(m_t)))/length(m_t);


figure(1);
subplot(211)
plot(t,m_t,'b','LineWidth',1);
title('Message signal','fontweight','bold','Fontsize',13);
ylim([1.5*min(m_t), 1.5*max(m_t)]);


subplot(212)
stem(fn,M_F,'b','LineWidth',1);
title('Message spectrum','fontweight','bold','Fontsize',13);
xlim([-2*fm, 2*fm]);
```

Output figure:

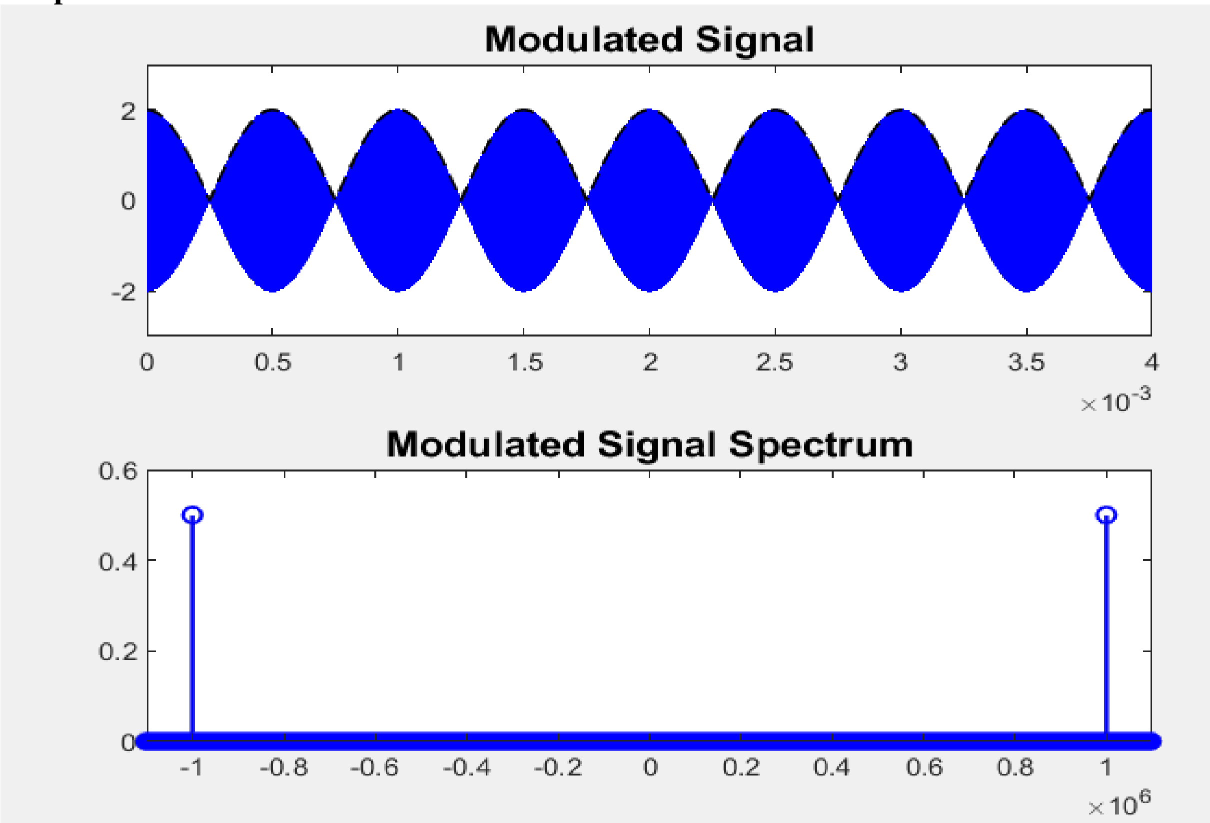


2)Balanced modulator:

This section mainly does the modulation. Here, we modulate the amplitude of our high frequency carrier signal with respect to the amplitude of the message signal for better transmission. To imitate this section , our Code is:

```
%% Balanced modulator output
s_t = m_t.*cos(2*pi*fc*t);
S_F = abs(fftshift(fft(s_t)))/length(s_t);


figure(3);
subplot(211)
plot(t,s_t,'b');
hold on
en_s_t=envelope(s_t);
plot(t,en_s_t,'k--','LineWidth',1);
title('Modulated Signal','fontweight','bold','Fontsize',13);
ylim([1.5*min(s_t), 1.5*max(s_t)]);


subplot(212)
stem(fn,S_F,'b','LineWidth',1);
title('Modulated Signal Spectrum','fontweight','bold','Fontsize',13);
xlim([-fc - 100*fm, fc + 100*fm]);
```

Output:



3)Output Amplifier:

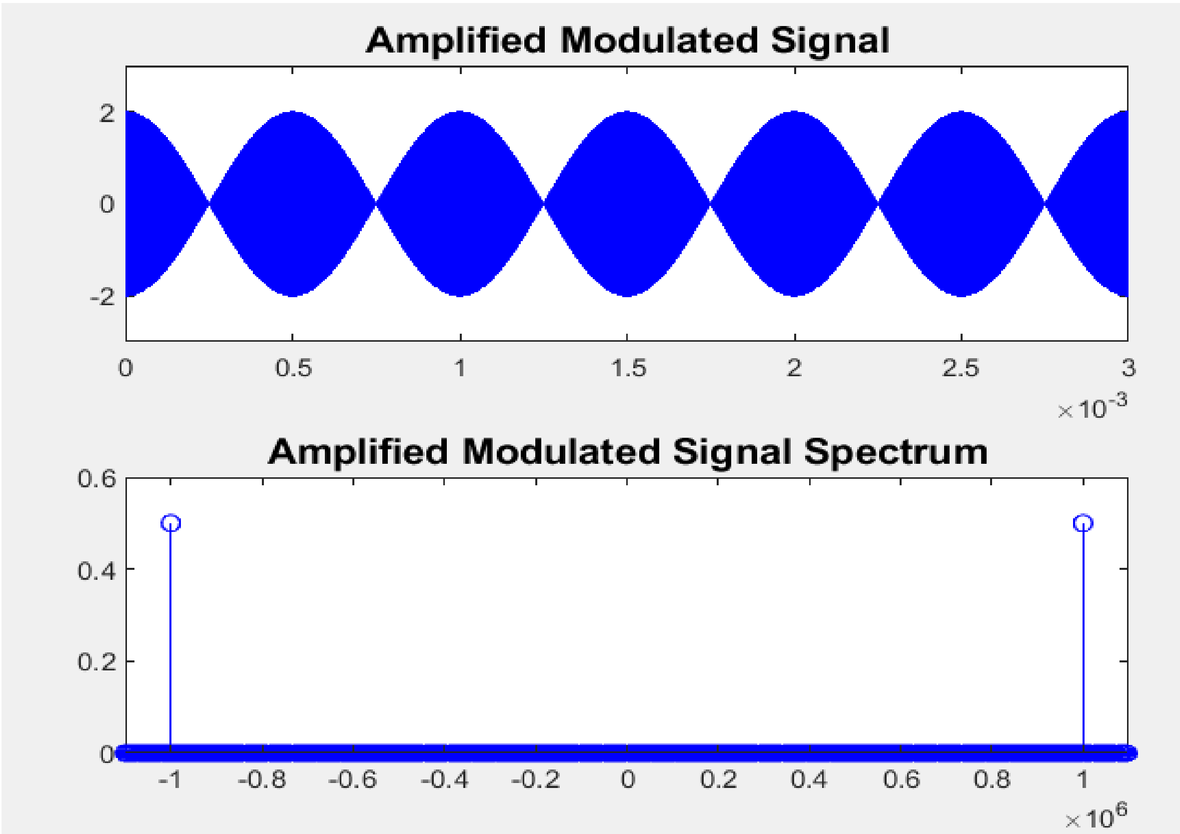
Before Transmitting, we can amplify the signal and then transmit this as for the wireless transmission attenuation is very common phenomena. But after amplification, freq component will not change , only there amplitude will vary.

Matlab code:

```
%% Output Amplifier
Amp_1=1; % "must" take as input vary
s_t_Amp=Amp_1*s_t;
S_F_Amp = abs(fftshift(fft(s_t_Amp)))/length(s_t_Amp);


figure(4);
subplot(211)
plot(t,s_t_Amp,'b');
title('Amplified Modulated Signal','fontweight','bold','Fontsize',13);
ylim([1.5*min(s_t), 1.5*max(s_t)]);


subplot(212)
```

```
stem(fn,S_F_Amp,'b');
title('Amplified Modulated Signal Spectrum','fontweight','bold','Fontsize',13);
xlim([-fc - 100*fm, fc + 100*fm]);
```

Output:



4) RF Amplifier:

After transmission, the amplitude of the modulated signal is reduced. Moreover, there are some small ripples due to channel noise. In this case, we consider SQNR of the channel noise is 5dB.

So, RF Amplifier is improved the amplitude of the modulated signal. It is connected to a tunning Nobe which can tune the local oscillator signal. So, the local oscillator signal is

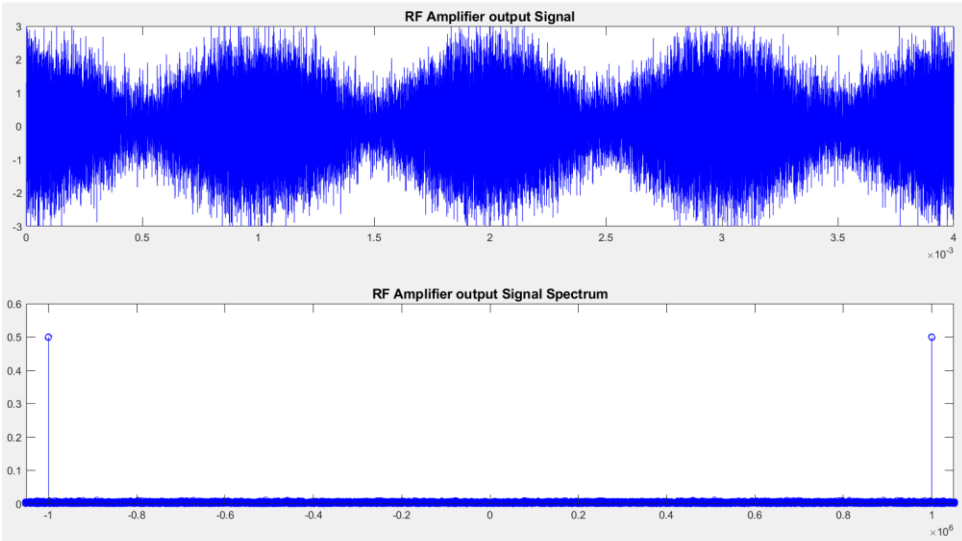
$$f_{LO} = f_{sig} + IF$$

Where, f_{sig}=Modulated Signal's Frequency=1MHz

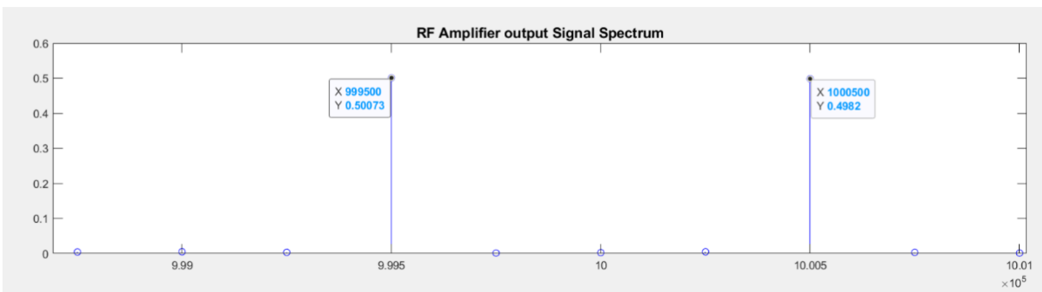
IR=455KHz

So, Local Oscillator Frequency=1455KHz

Plot:



In Zoom view,



The Frequency Components are at 1.0005MHz & 0.9995MHz as predicted.

Matlab Code:

```
%% (ANACOM 1/2)
s_t_Amp_noise=awgn(s_t_Amp,5); % Add white Gaussian noise with SQNR=5 dB


%% Local Oscillator
IF=455e3;
f_local = IF + fc ; %local oscillator frequency 455kHz + fc
local_t = cos(2*pi*f_local*t);

%% RF Amplifier
Amp_2=1; % "must" take as input
s_t_rf=Amp_2*s_t_Amp_noise;
S_F_rf = abs(fftshift(fft(s_t_rf)))/length(s_t_rf);
```

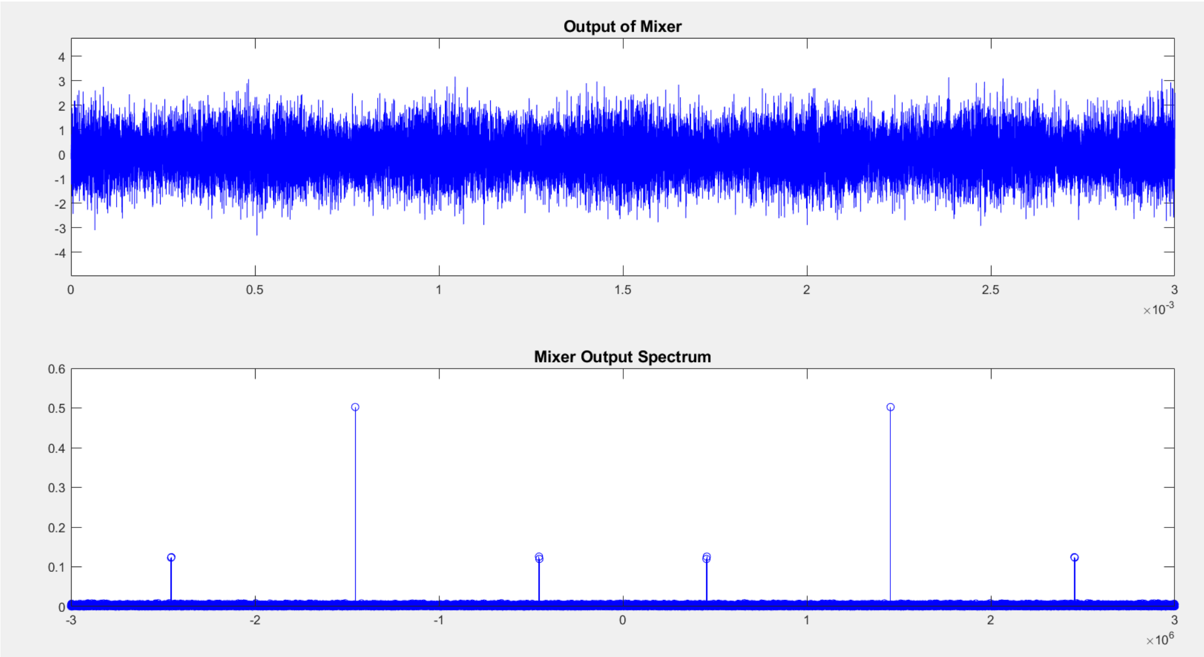
```
figure(5);
subplot(211)
plot(t,s_t_rf,'b');
title('RF Amplifier output Signal','fontweight','bold','Fontsize',13);
ylim([1.5*min(s_t), 1.5*max(s_t)]);

subplot(212)
stem(fn,S_F_rf,'b');
title('RF Amplifier output Signal Spectrum','fontweight','bold','Fontsize',13);
xlim([-fc - 100*fm, fc + 100*fm]);
```

5) Mixer Block:

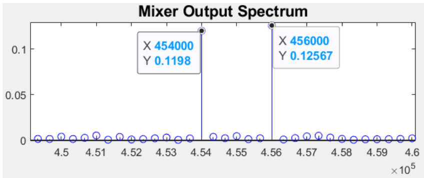
The mixer block output combines the RF Amplifier signal and the local oscillator signal output. Hence, we get a three frequency mixed signal of the shape.

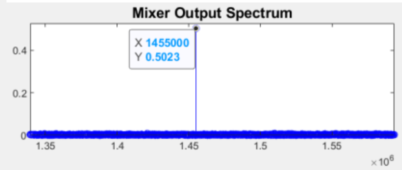
The frequencies are: IF, f_{sig} +IF, $2f_{sig}$+IF.
Then we send this into a BPF and we get only IF frequency's signal. There are small ripples which are caused by noise.

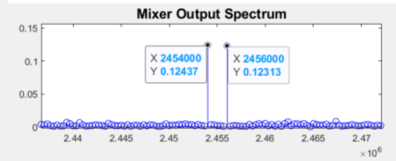
Before BPF



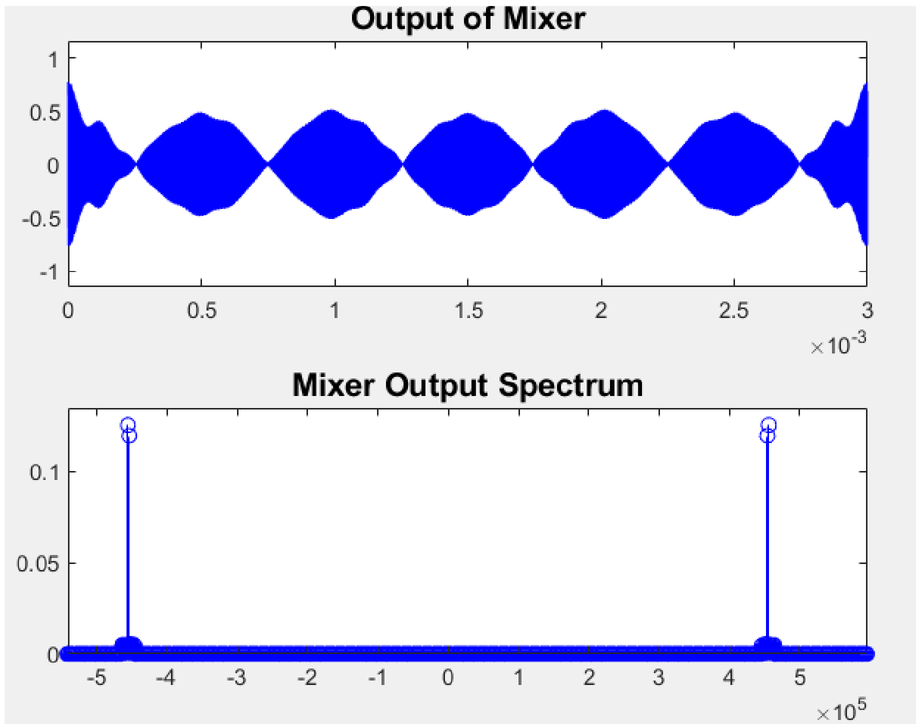
In separated view



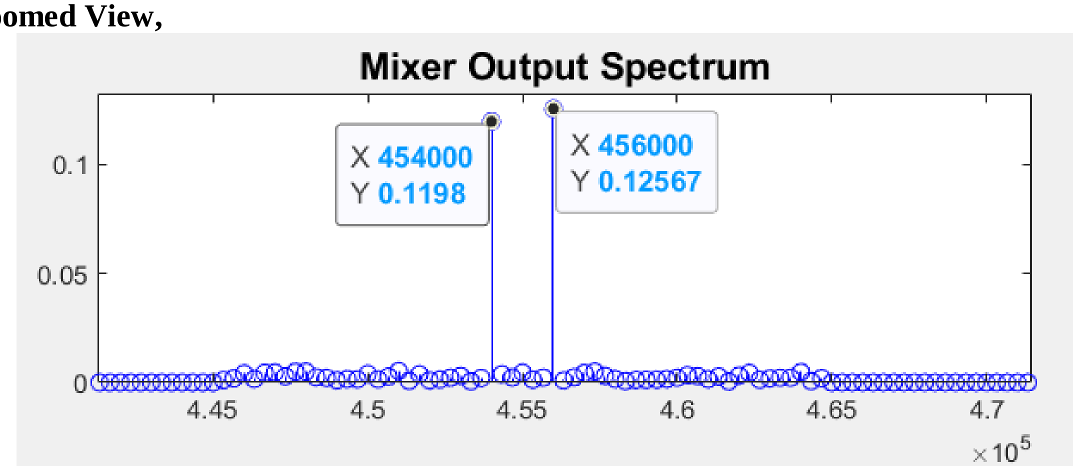




After BPF



In Zoomed View,



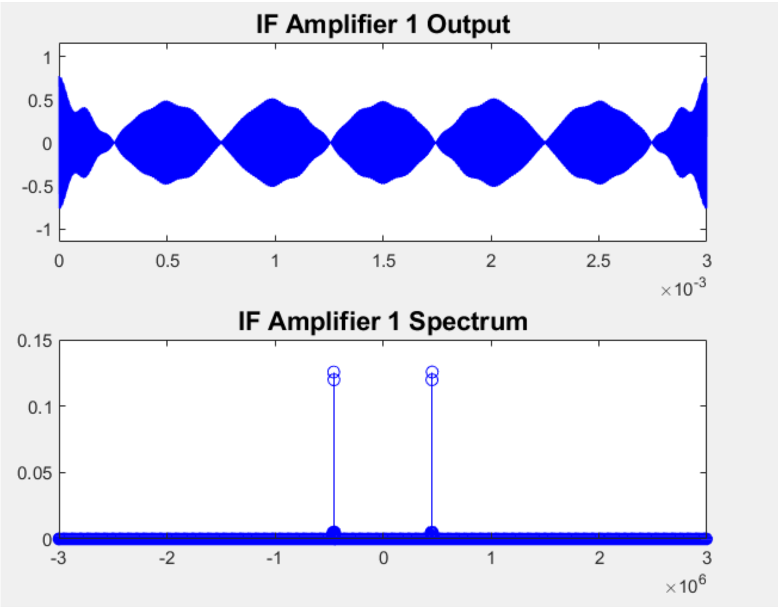
Matlab Code:

```
%% Mixer
mixer_t =(1+s_t_rf).* local_t;

Mixer_F = abs(fftshift(fft(mixer_t)))/length(mixer_t);

figure(6);
subplot(211)
plot(t,mixer_t,'b');
title('Output of Mixer','fontweight','bold','Fontsize',13);
ylim([1.5*min(mixer_t), 1.5*max(mixer_t)]);

subplot(212)
stem(fn,Mixer_F,'b');
title('Mixer Output Spectrum','fontweight','bold','Fontsize',13);

%% After Bandpass filter,
bpf=zeros(1,length(fn));
bpf(find(abs(fn)<465e3 & abs(fn)>445e3))=1;
Mixer_F_bpf=bpf.*Mixer_F;

mixer_t_bpf= ifft(abs(fftshift(Mixer_F_bpf))).*length(Mixer_F_bpf);

figure(7);
subplot(211)
plot(t,mixer_t_bpf,'b');
title('Output of Mixer','fontweight','bold','Fontsize',13);
ylim([1.5*min(mixer_t_bpf), 1.5*max(mixer_t_bpf)]);

subplot(212)
stem(fn,Mixer_F_bpf,'b');
title('Mixer Output Spectrum','fontweight','bold','Fontsize',13);
```

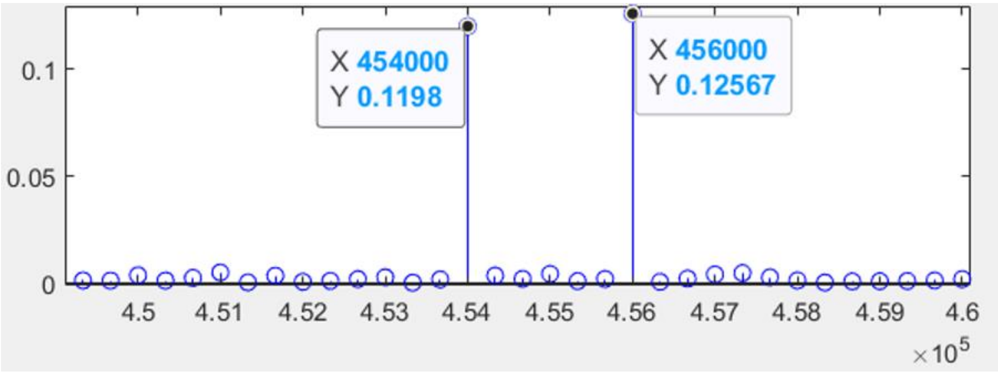
6) IF Amplifier:

The main purpose is to use the IF Amplifier is amplified the output of mixer block.

In this case, we use 2 IF Amplifier blocks. Basically, if we use only one amplifier and set the gain is high, then the signal can go into non-linear region. So, we get high frequency component in the modulated signal. So, we cannot recover the message signal from this situation. So, we use 2 IF Amplifier block and work in the linear region.



In zoomed view,



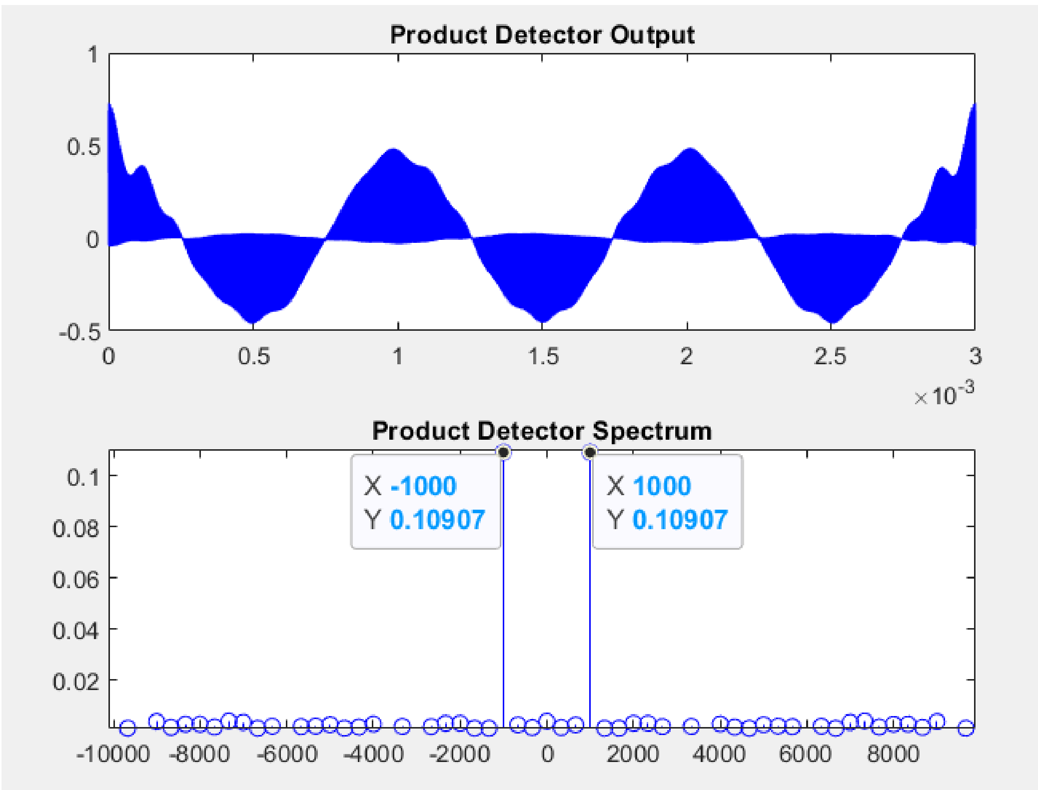
MATLAB Code:

```
%% IF Amplifier
Amp_3=1; %"must" take as input
Amp_mixer_t = Amp_3*mixer_t_bpf;

Amp_mixer_F = abs(fftshift(fft(Amp_mixer_t)))/length(Amp_mixer_t);

figure(8);
subplot(211)
plot(t,Amp_mixer_t,'b');
title('IF Amplifier 1 Output','fontweight','bold','Fontsize',13);
ylim([1.5*min(Amp_mixer_t), 1.5*max(Amp_mixer_t)]);
subplot(212)
stem(fn,Amp_mixer_F,'b');
```

```
title('IF Amplifier 1 Spectrum','fontweight','bold','Fontsize',13);
```

Product Detector:

In this block, the output of IF amplifier signal is modulated with local oscillator frequency. And we get message signal with high frequency components.



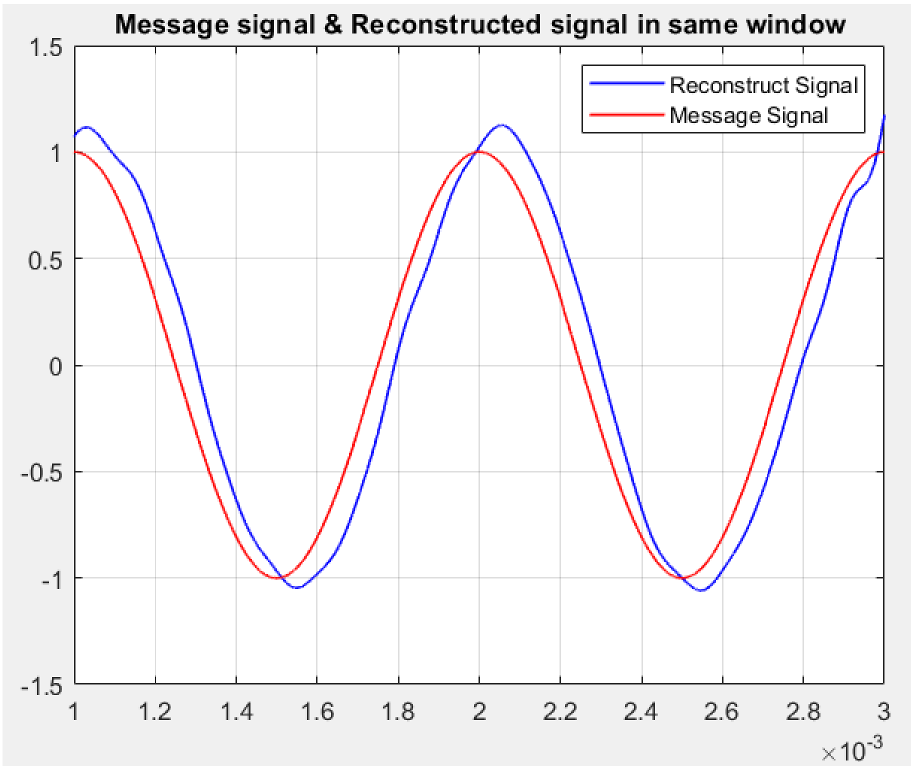
MATLAB Code:

```
%% Product Detector
Product_out=Amp_filter_t_2.*cos(2*pi*IF*t);
product_out_F=abs(fftshift(fft(Product_out)))/length(Product_out);

figure(10);
subplot(211)
plot(t,Product_out,'b');
title('Product Detector Output','fontweight','bold');

subplot(212)
stem(fn,product_out_F,'b');
```

```
title('Product Detector Spectrum','fontweight','bold');
xlim([-10*fm, 10*fm]);
```

Audio Amplifier Block:

The demodulated signal m'(t) and the original signal m(t) are shown in the graph. Here, m(t) and m'(t) both has phase shift due to using filters, amplifiers. But the frequency components are same. And we add a gain to amplify the signal. We also add a LPF to remove the high frequency components.



MATLAB Code:

```
%% Audio Amplifier Output
lpf=5*fm;
[b,a] = butter(2,lpf/(Fs/2));          %LowPass filter

final_signal = filter(b,a,Product_out);

Ampli=5; %must take as input
Amp_final_signal = Ampli*final_signal;

figure(11);
plot(t,Amp_final_signal,'b', 'linewidth',1);
hold on;
```

```
plot(t,m_t,'r', 'linewidth',1);
xlim([t_s, cycle*t_s]);
legend('Reconstruct Signal','Message Signal');
grid on
title('Message signal & Reconstructed signal in same window','fontweight','bold');
```

Limitations of AM Modulation:

Amplitude modulation is the oldest and simplest method of performing modulation. Its greatest virtue is the simplicity of implementation. The equipment's required to make a receiver for amplitude modulated wave is very cheap and available. This is why most national radio transmission still uses AM wave. However, amplitude modulation do suffer from two major limitations:

1) Amplitude modulation is wasteful of power. The carrier wave c(t) is completely independent of the information bearing signal m(t). The transmission of c(t) therefore represents a wastage of power.
2) Only one sideband is necessary for communication while in AM both the side bands are transmitted, So it is a wastage of bandwidth
3) Also AM waves are highly vulnerable to noise which is seen from our project. Basically practical filters are not sharp. So, it can not remove noise.

Conclusion:

According to this project, we learnt about the generation of DSB-WC & DSB-SC. We implement ANACOM board to generate, transmit and receive signal.

Graphical user interfaces (**GUIs**), in Matlab, also known as apps, provide point-and-click control of our software applications, helps us a lot to make our demonstration more simpler and user friendly. To make this project like real life, we also use additive white Gaussian noise after transmitting the signal. This project can be very useful to demonstrate Amplitude modulation like in this pandemic situation, as we can't use our hardware lab, it can be very handy.
After all, we successfully complete our project and the project is really an eye opener for us.