Ministry of Higher Education

Higher Technological Institute

10th of Ramadan City

6th of October Branch

Electrical and Computer Engineering Department

Communication Systems

EET 139 (G 92)

MATLAB Task

Under Supervision: Prof. Dr. Ali Elmoghazy

Eng. Mahmoud hamed/ Eng. Kareem Badawi

Name / Muhammad Magdy Srour

ID/ 32015251

Table of Contents

1 - AM (Amplitude modulation)

1.1 – introduction

amplitude modulation (AM) is a modulation technique used in electronic communication, most commonly for transmitting information via a radio carrier wave. In amplitude modulation, the amplitude (signal strength) of the carrier wave is varied in proportion to that of the message signal being transmitted. The message signal is, for example, a function of the sound to be reproduced by a loudspeaker, or the light intensity of pixels of a television screen. This technique contrasts with frequency modulation, in which the frequency of the carrier signal is varied, and phase modulation, in which its phase is varied.

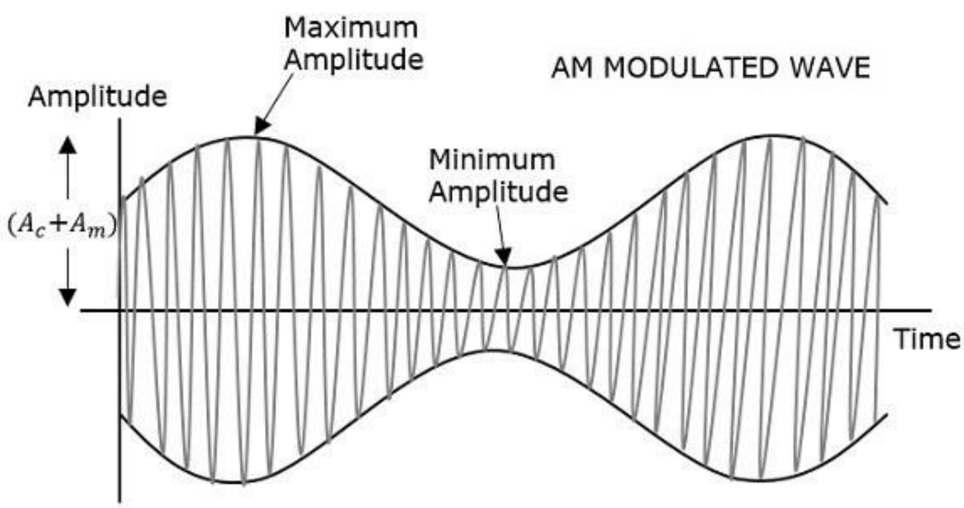

Figure 1 AM signal

1.2 – time domain modulation equation

S(t) = (Ac+m(t)) * cos(Wc*t)

Where

S(t) = AM modulated signal .

AC = carrier amplitude.

M(t) = massage signal .

1.3 – modulation method

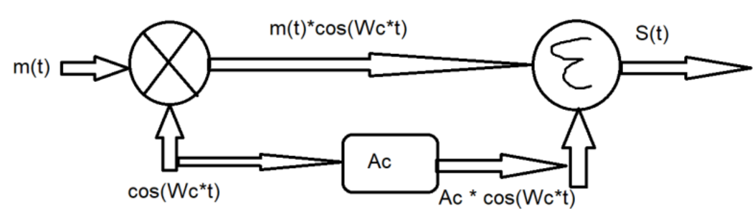

Figure 2 AM modulation method

1.4 – de-modulation method

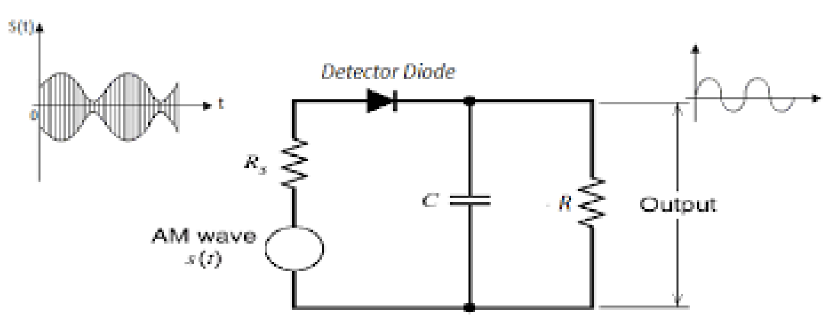

Figure 3 AM de-modulation method

1.5 – MATLAB code

Set the code parameters

```
%time setting
Fs = 800000;              %sample freq
tmin = 0;                 %initial time
tmax = 0.001;             %max time
step = 1/Fs;              %sample time
t = tmin:step:tmax;       %time periode of signal
%amplitude setting
Am = 1;                   %msg amplitude
Ac = 2;                   %carrier amplitude
Fm = 2000;
Fc = 100000;

```

Figure 4 AM Set the code parameters

Generate massage and carrier

```
%generate msg & carrier & modulated signals
m = Am*cos(2*pi*Fm*t);    %msg
c = Ac*cos(2*pi*Fc*t);    %carrier
```

Figure 5 AM Generate massage and carrier

Plotting the massage and carrier

```
figure(1);
%msg
subplot(2,1,1);
plot(t,m);
xlabel ('time');
ylabel ('amplitude');
title('msg signal');


%carrier
subplot(2,1,2);
plot(t,c);
xlabel ('time');
ylabel ('amplitude');
title('carrier signal');

```

Figure 6 AM Plotting the massage and carrier

massage and carrier

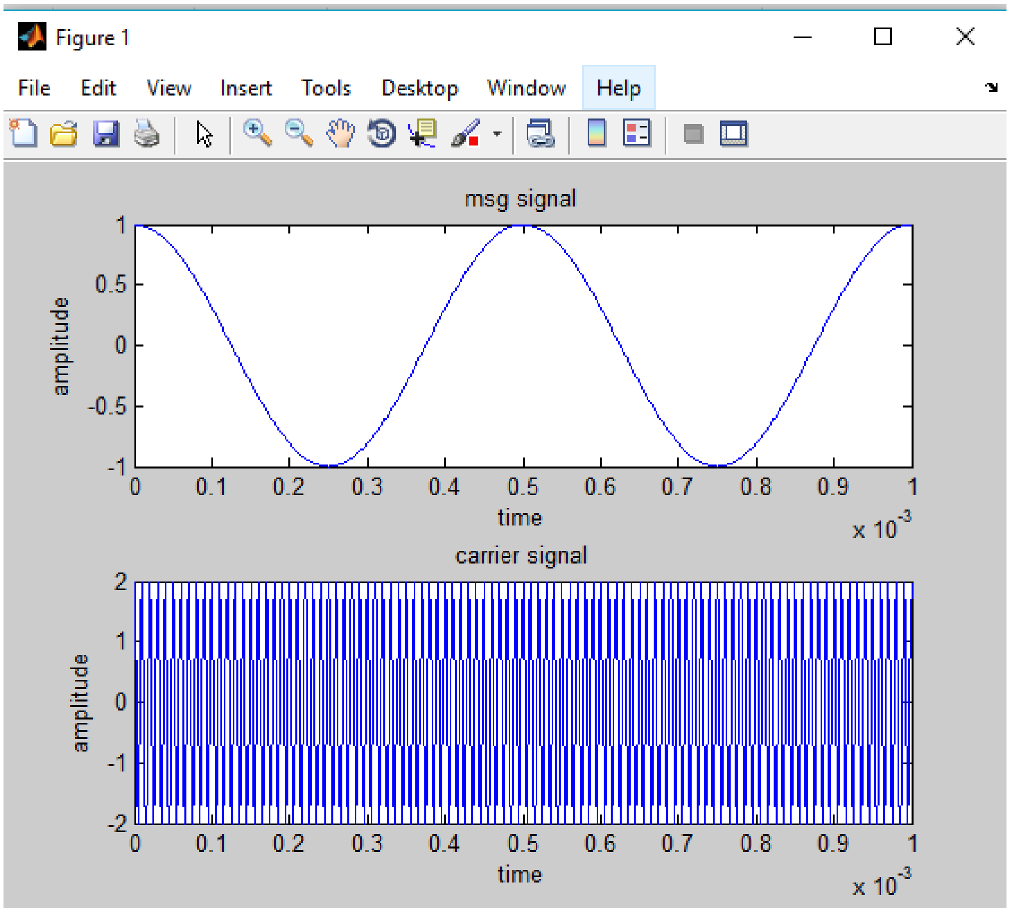

Figure 7 AM massage and carrier

Modulation

```
s = (1+m/Ac).*c;          %AM modulated signal
```

Figure 8 AM Modulation equation

Demodulation

```
%demodulation of AM
y = amdemod(s,Fc,Fs);     %built in function for AM-DEmodulation
```

Figure 9 AM Demodulation equation

Plotting modulated and demodulated signals

```
    figure(2);
    %AM modulated signal
    subplot(2,1,1);
    plot(t,s); hold on;
    plot (t,Ac*(1+m/Ac),'r:'); hold on;
    plot (t,-Ac*(1+m/Ac),'r:'); hold on;
    xlabel ('time');
    ylabel ('amplitude');
    title('AM-modulated signal');


    %AM demodulated signal
    subplot(2,1,2);
    plot(t,y);
    xlabel ('time');
    ylabel ('amplitude');
    title('AM-demodulated signal');

```

Figure 10 AM Plotting modulated and demodulated signals

modulated and demodulated signals

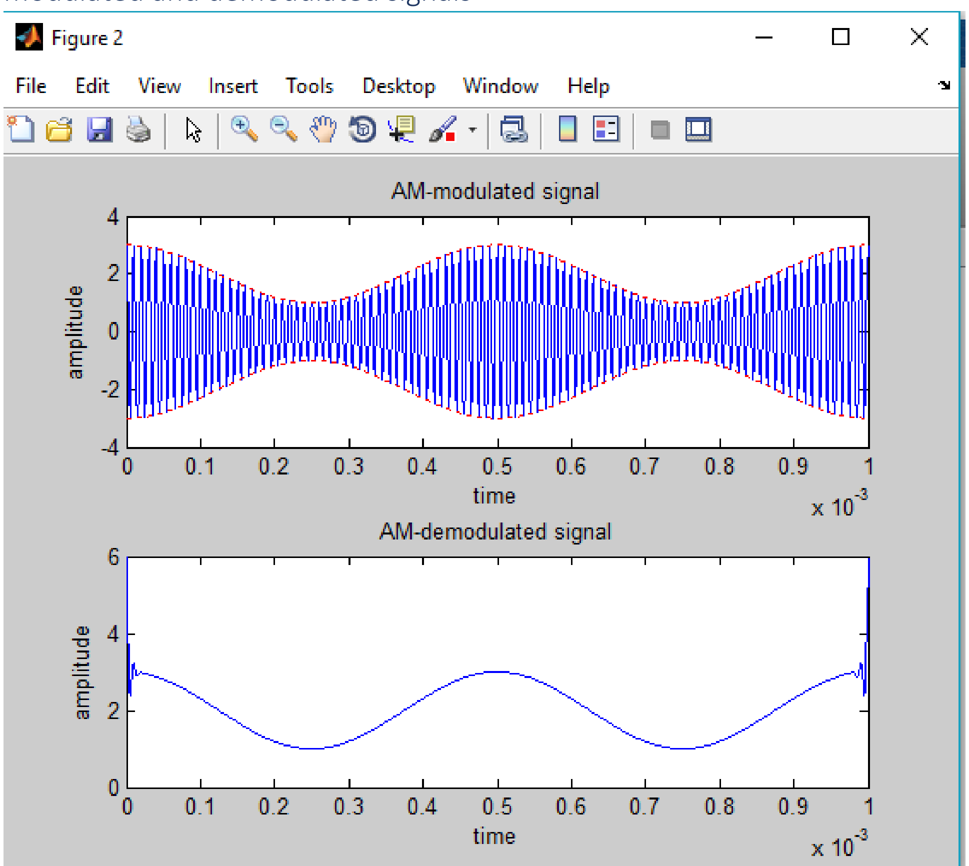

Figure 11 AM modulated and demodulated signals

2 - DSB-SC (Double-sideband suppressed-carrier transmission)

2.1 – introduction

is transmission in which frequencies produced by amplitude modulation (AM) are symmetrically spaced above and below the carrier frequency and the carrier level is reduced to the lowest practical level, ideally being completely suppressed.

In the DSB-SC modulation, unlike in AM, the wave carrier is not transmitted; thus, much of the power is distributed between the side bands, which implies an increase of the cover in DSB-SC, compared to AM, for the same power used.

DSB-SC transmission is a special case of double-sideband reduced carrier transmission. It is used for radio data systems.

2.2 – time domain modulation equation

S(t) = m(t) * Ac * cos(Wc*t)

Where

S(t) = DSB-SC modulated signal .

AC = carrier amplitude.

M(t) = massage signal .

2.3 – modulation method

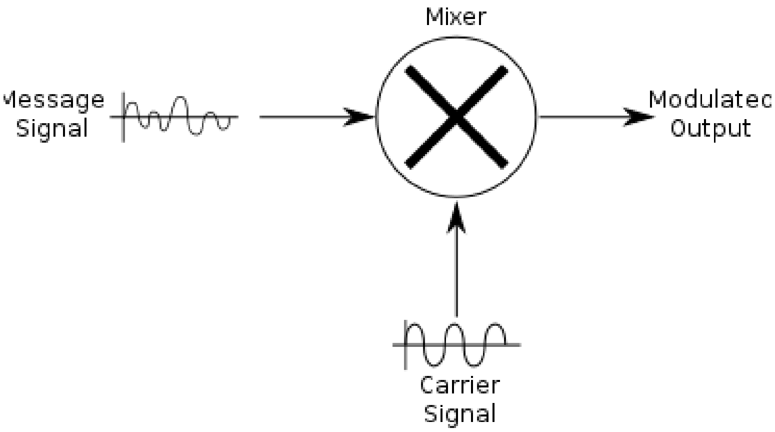

Figure 12 DSB-SC modulation method

2.4 – de-modulation method

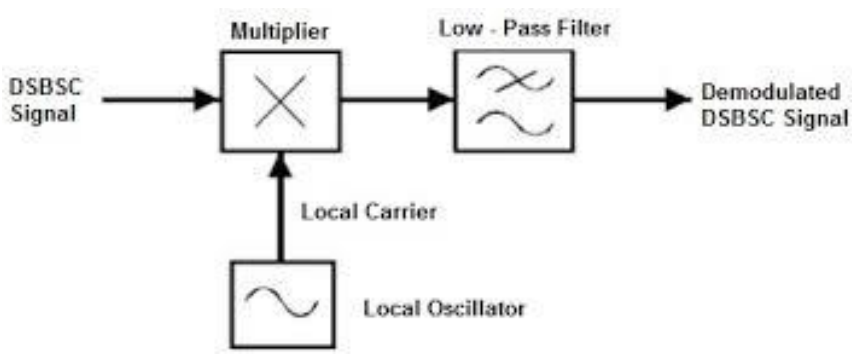

Figure 13 DSB-SC demodulation method

2.5 – MATLAB code

Set the code parameters

```
%time setting
Fs = 800000;                %sample freq
tmin = 0;                   %initial time
tmax = 0.001;               %max time
step = 1/Fs;                %sample time
t = tmin:step:tmax;         %time periode of signal
%amplitude setting
Am = 1;                     %msg amplitude
Ac = 2;                     %carrier amplitude
Fm = 2000;
Fc = 100000;

```

Figure 14 DCB-SC Set the code parameters

Generate massage and carrier

```
%generate msg & carrier & modulated signals
m = Am*cos(2*pi*Fm*t);    %msg
c = Ac*cos(2*pi*Fc*t);    %carrier
```

Figure 15 DCB-SC Generate massage and carrier

Plotting the massage and carrier

```
figure(1);
%msg
subplot(2,1,1);
plot(t,m);
xlabel ('time');
ylabel ('amplitude');
title('msg signal');


%carrier
subplot(2,1,2);
plot(t,c);
xlabel ('time');
ylabel ('amplitude');
title('carrier signal');
```

Figure 16 DSB-SC Plotting the massage and carrier

massage and carrier

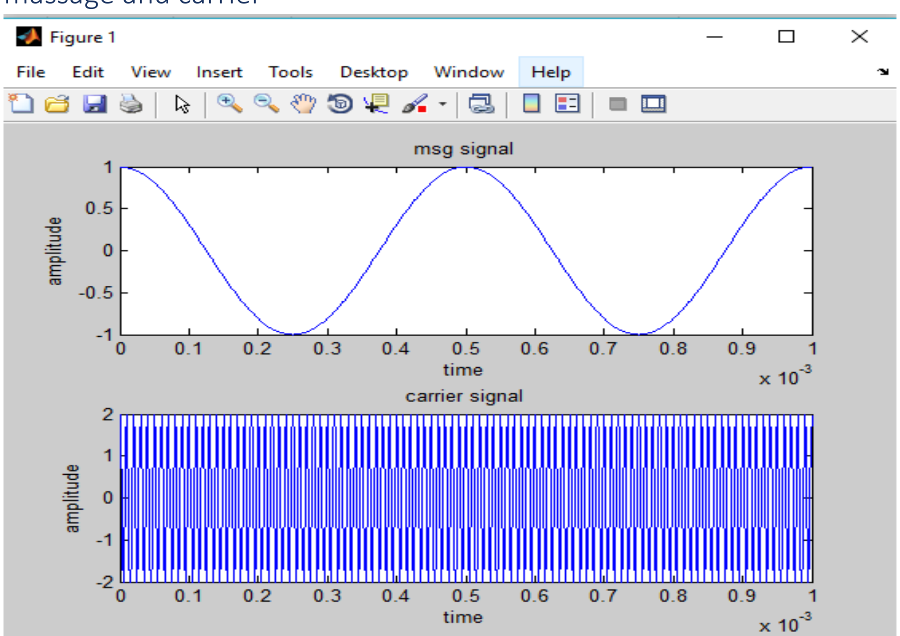

Figure 17 DCB-SC massage and carrier

Modulation

```
s = m.*c;                   %DSB-SC modulated signal
```

Figure 18 DSB-SC Modulation equation

Demodulation

```
%demodulation of DSB-SC
x = s.*c;
Wn = Fc/(Fs/2);             % Normalozed cutoff frequency
[b a] = butter(5,Wn,'low'); %5th order low-pass filter
y = filter(b,a,x);
```

Figure 19 DSB-SC Demodulation equation

Plotting modulated and demodulated signals

```
figure(2);
%DSB-SC modulated signal
subplot(2,1,1);
plot(t,s); hold on;
plot (t,Ac*m,'r:'); hold on;
plot (t,-Ac*m,'r:'); hold on;
xlabel ('time');
ylabel ('amplitude');
title('DSB-SC modulated signal');


%DSB-SC demodulated signal
subplot(2,1,2);
plot(t,y);
xlabel ('time');
ylabel ('amplitude');
title('DSB-SC demodulated signal');

```

Figure 20 DSB-SC Plotting modulated and demodulated signals modulated and demodulated signals

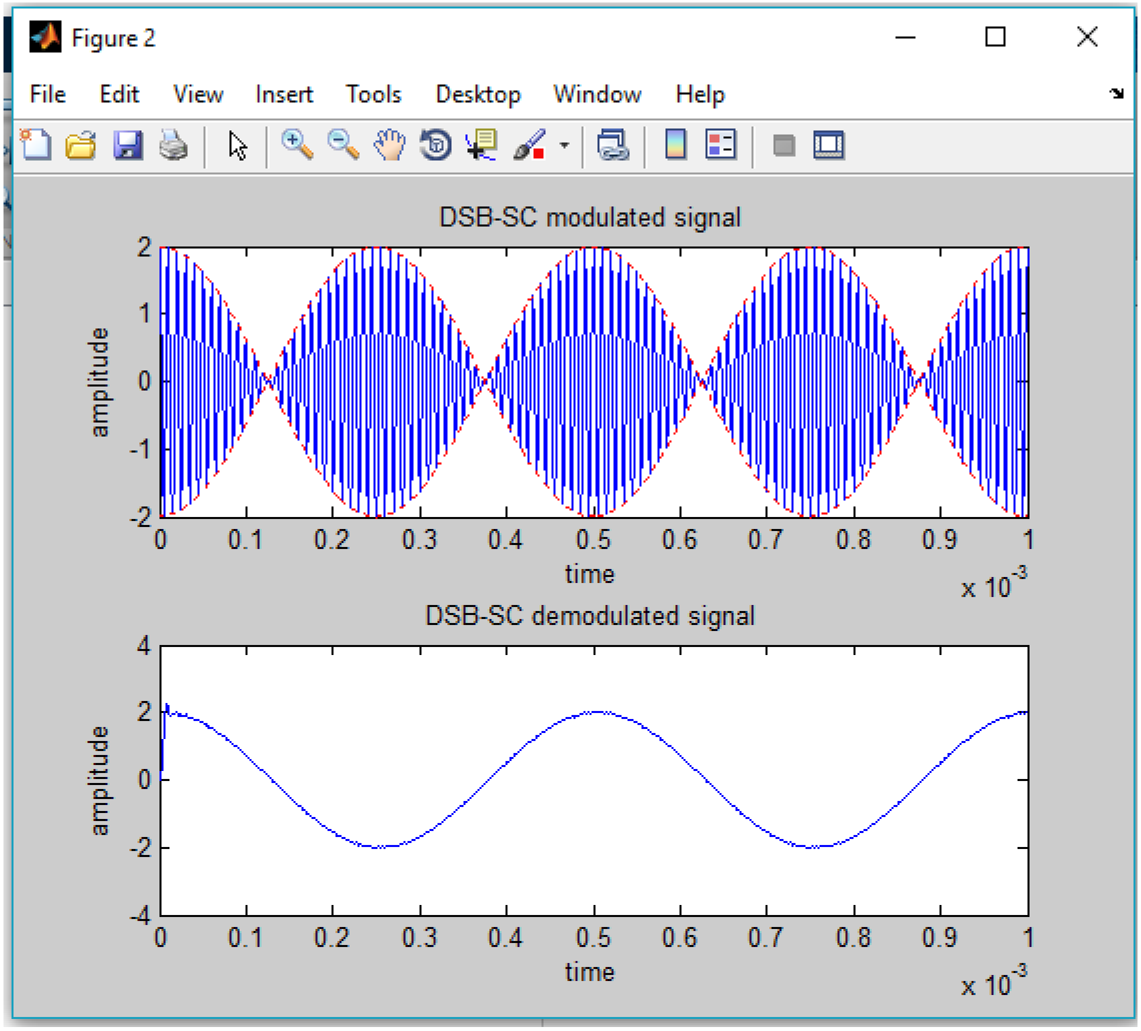

Figure 21 DSB-SC modulated and demodulated signals

3 – SSB (Single-sideband modulation)

3.1 – introduction

is a type of modulation, used to transmit information, such as an audio signal, by radio waves. A refinement of amplitude modulation, it uses transmitter power and bandwidth more efficiently. Amplitude modulation produces an output signal that has twice the bandwidth of the original baseband signal. Single-sideband modulation avoids this bandwidth doubling, and the power wasted on a carrier, at the cost of increased device complexity and more difficult tuning at the receiver.

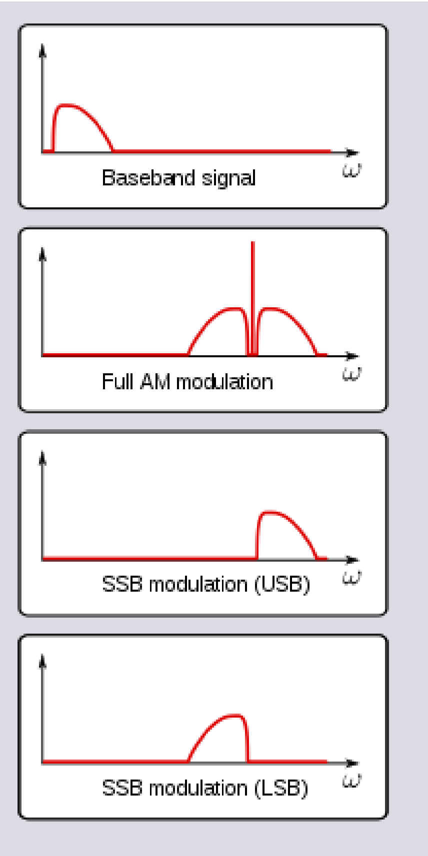

Figure 22 SSB signal

3.2 – time domain modulation equation

S(t) = m(t)*cos(Wc*t) – Mh(t)*sin(Wc*t)   for USB (upper side band)

S(t) = m(t)*cos(Wc*t) + Mh(t)*sin(Wc*t)   for LSB (lower side band)

Where

S(t) = DSB-SC modulated signal .

M(t) = massage signal .

Mh(t) = hilbert transform of m(t).

3.3 – modulation method

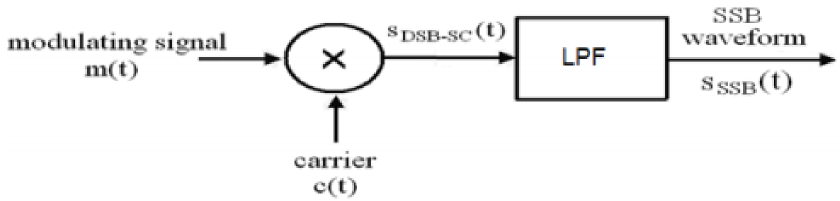

Figure 23 SSB modulation method 1

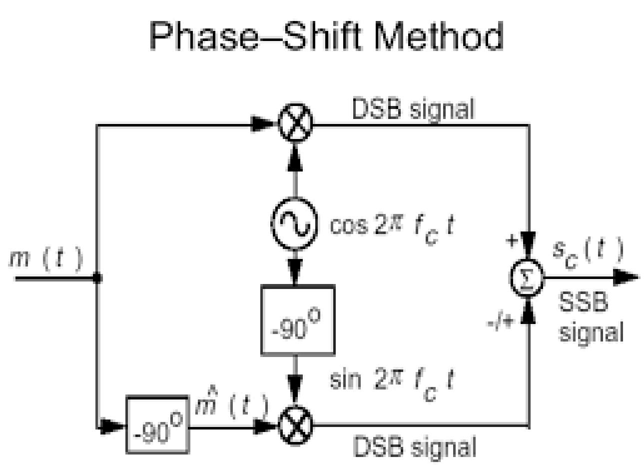

Figure 24 SSB modulation method 2

3.4 – de-modulation method

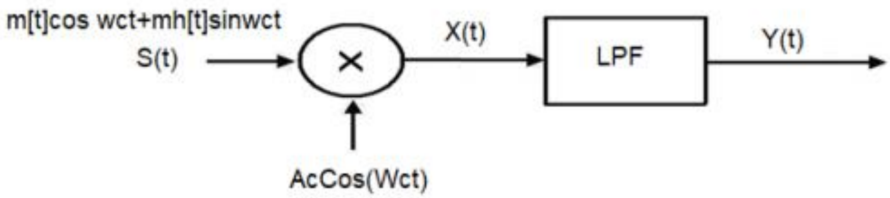

Figure 25 SSB de-modulation method

3.5 – MATLAB code

Set the code parameters

```
%time setting
Fs = 800000;            %sample freq
tmin = 0;               %initial time
tmax = 0.001;           %max time
step = 1/Fs;            %sample time
t = tmin:step:tmax;     %time periode of signal
%amplitude setting
Am = 1;                 %msg amplitude
Ac = 2;                 %carrier amplitude
Fm = 2000;
Fc = 100000;

```

Figure 26 SSB Set the code parameters

Generate massage and carrier

```
m  = Am*cos(2*pi*Fm*t);        %msg
mh = Am*cos(2*pi*Fm*t+90);     %helbert transform of msg
c1 = Ac*cos(2*pi*Fc*t);        %carrier (cos)
c2 = Ac*sin(2*pi*Fc*t);        %carrier (sin)
```

Figure 27 SSB Generate massage and carrier

Plotting the massage and carrier

```
figure(1);
%msg
subplot(2,1,1);
plot(t,m);
xlabel ('time');
ylabel ('amplitude');
title('msg signal');

%helbert transform of msg
subplot(2,1,2);
plot(t,mh);
xlabel ('time');
ylabel ('amplitude');
title('helbert transform of msg');

figure(2);
%carrier
subplot(2,1,1);
plot(t,c1);
xlabel ('time');
ylabel ('amplitude');
title('carrier signal');

%carrier
subplot(2,1,2);
plot(t,c2);
xlabel ('time');
ylabel ('amplitude');
title('carrier of helbert signal');
```

Figure 28 SSB Plotting the massage and carrier

massage and carrier

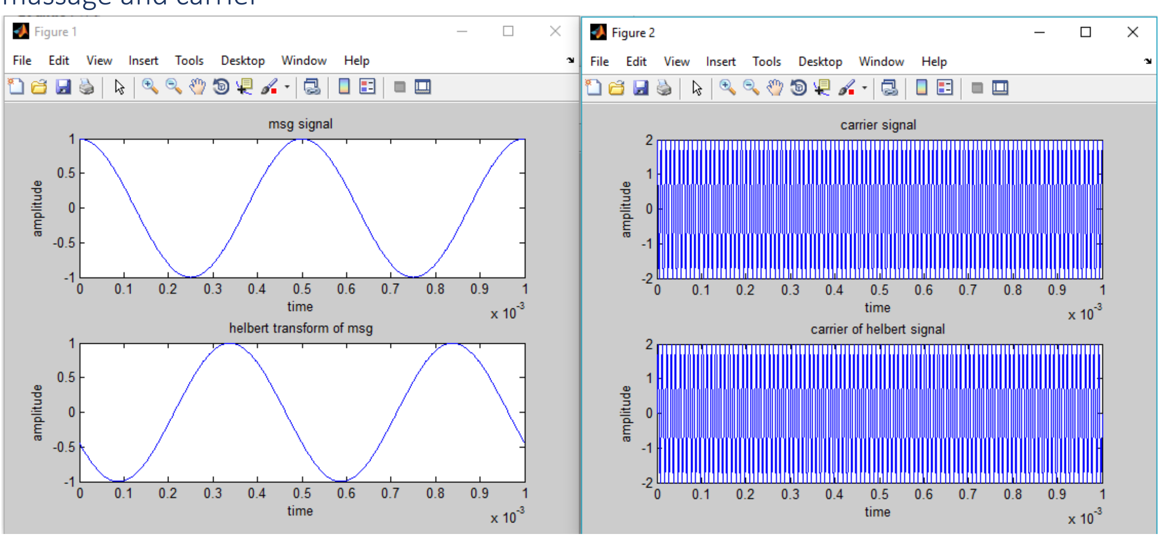

Figure 29 SSB massage and carrier

Modulation

```
%SSB(LSB) modulated signal
s1 = m.*c1;
s2 = mh.*c2;
s = s1+s2;
```

Figure 30 SSB(LSB) Modulation equation

Demodulation

```
%demodulation of SSB(LSB)
x = c1.*s;
Wn = Fc/(Fs/2);              % Normalozed cutoff frequency
[b a] = butter(5,Wn,'low'); %5th order low-pass filter
y = filter(b,a,x);
```

Figure 31 SSB(LSB) Demodulation equation

Plotting modulated and demodulated signals

```
figure(3);
%SSB(LSB) modulated signal
subplot(2,1,1);
plot(t,s);
xlabel ('time');
ylabel ('amplitude');
title('SSB(LSB)-modulated signal');


%SSB(LSB) demodulated signal
subplot(2,1,2);
plot(t,y);
xlabel ('time');
ylabel ('amplitude');
title('SSB(LSB)-demodulated signal');
```

Figure 32 SSB(LSB) Plotting modulated and demodulated signals modulated and demodulated signals

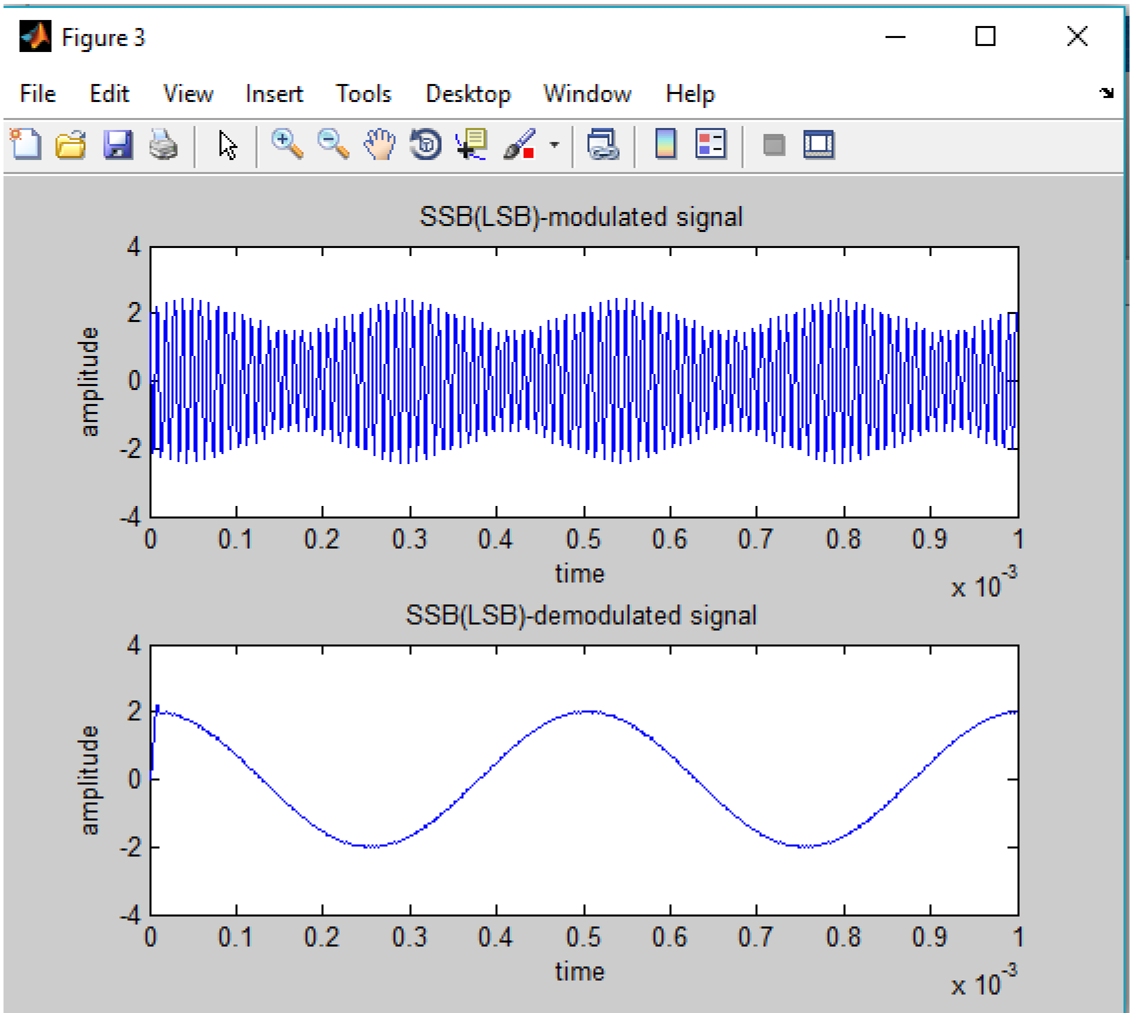

Figure 33 SSB(LSB) modulated and demodulated signals

4 – MATLAB used functions

4.1 amdemod

```
>> help amdemod
 amdemod Amplitude demodulation.
    Z = amdemod(Y,Fc,Fs) demodulates the amplitude modulated signal Y from
    the carrier frequency Fc (Hz). Y and Fc have sample frequency Fs (Hz).
    The modulated signal Y has zero initial phase, and zero carrier
    amplitude, for suppressed carrier modulation. A lowpass filter is used
    in the demodulation.  The default filter is: [NUM,DEN] =
    butter(5,Fc*2/Fs).

    Z = amdemod(Y,Fc,Fs,INI_PHASE) specifies the initial phase (rad) of the
    modulated signal.

    Z = amdemod(Y,Fc,Fs,INI_PHASE,CARRAMP) specifies the carrier amplitude
    of the modulated signal for transmitted carrier modulation.

    Z = amdemod(Y,Fc,Fs,INI_PHASE,CARRAMP,NUM,DEN) specifies the filter to
    be used in the demodulation.
```

Figure 34 amdemod function

4.2 butter

```
>> help butter
 butter Butterworth digital and analog filter design.
    [B,A] = butter(N,Wn) designs an Nth order lowpass digital
    Butterworth filter and returns the filter coefficients in length
    N+1 vectors B (numerator) and A (denominator). The coefficients
    are listed in descending powers of z. The cutoff frequency
    Wn must be 0.0 < Wn < 1.0, with 1.0 corresponding to
    half the sample rate.

    If Wn is a two-element vector, Wn = [W1 W2], butter returns an
    order 2N bandpass filter with passband  W1 < W < W2.
    [B,A] = butter(N,Wn,'high') designs a highpass filter.
    [B,A] = butter(N,Wn,'low') designs a lowpass filter.
    [B,A] = butter(N,Wn,'stop') is a bandstop filter if Wn = [W1 W2].

    When used with three left-hand arguments, as in
    [Z,P,K] = butter(...), the zeros and poles are returned in
    length N column vectors Z and P, and the gain in scalar K.

    When used with four left-hand arguments, as in
    [A,B,C,D] = butter(...), state-space matrices are returned.

    butter(N,Wn,'s'), butter(N,Wn,'high','s') and butter(N,Wn,'stop','s')
    design analog Butterworth filters.  In this case, Wn is in [rad/s]
    and it can be greater than 1.0.
```

Figure 35 butter function