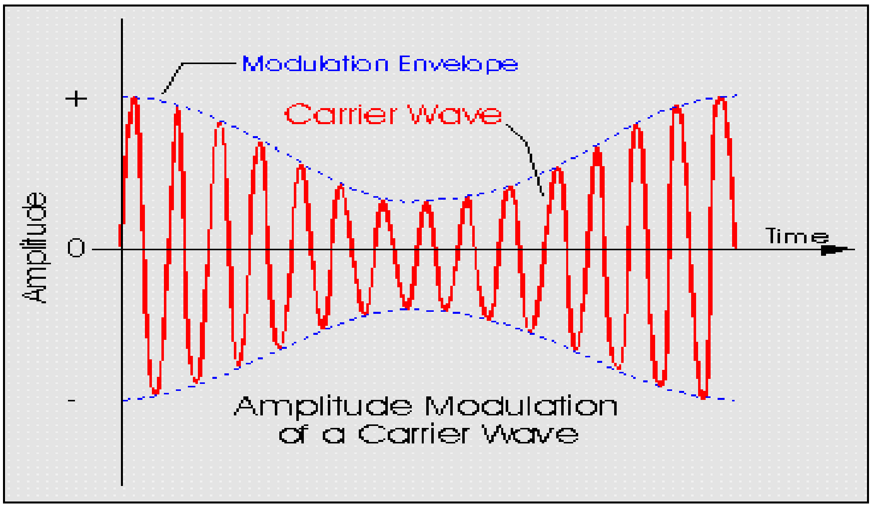
Modulating Signal by Matlab R2010a



amplitude modulation

Amplitude modulation(AM)

*f(t)=(A+m(t))*cos(2*pi*fc)*

Syntax

```
y = ammod(x,Fc,Fs)
y = ammod(x,Fc,Fs,ini_phase)
y = ammod(x,Fc,Fs,ini_phase,carramp)
```

Description

بالنسبة للايعازات فلا ازيد بلشرح عن ماقاله مجموعة من امهر مبرمجي(Mathwork)
واظن ان وصفهم واضح جدا لكل ايعاز

`y = ammod(x,Fc,Fs)` uses the message signal `x` to modulate a carrier signal with frequency `Fc` (Hz) using amplitude modulation. The carrier signal and `x` have sample frequency `Fs` (Hz). The modulated signal has zero initial phase and zero carrier amplitude, so the result is suppressed-carrier modulation.

> **Note** The `x`, `Fc`, and `Fs` input arguments must satisfy `Fs` > 2(`Fc` + `BW`), where `BW` is the bandwidth of the modulating signal `x`.

`y = ammod(x,Fc,Fs,ini_phase)` specifies the initial phase in the modulated signal `y` in radians.

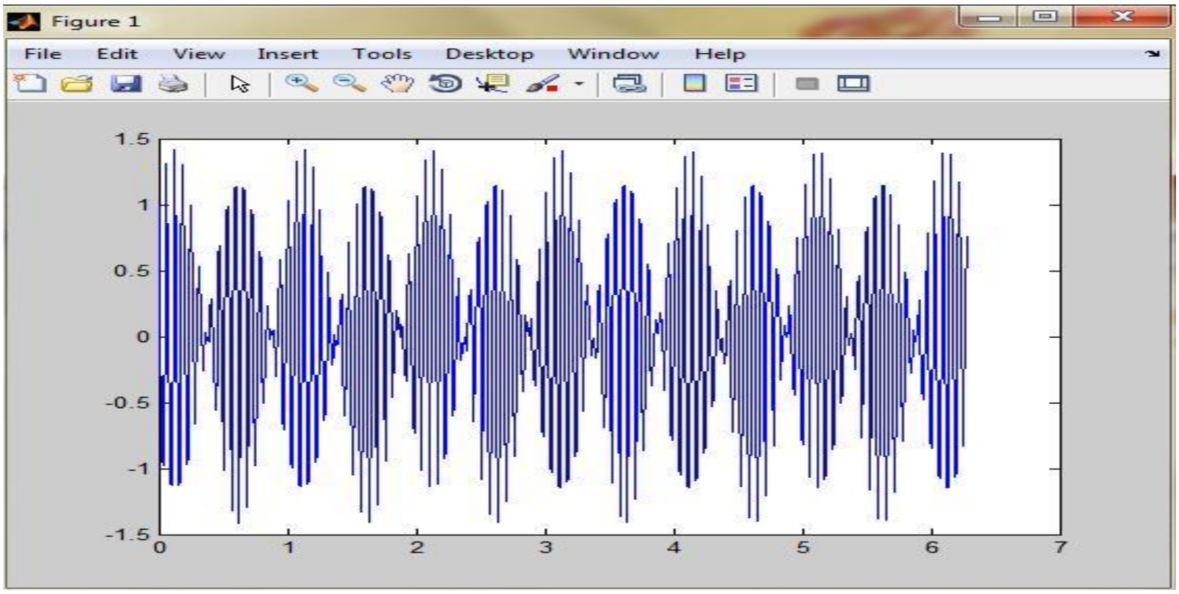
`y = ammod(x,Fc,Fs,ini_phase,carramp)` performs transmitted-carrier modulation instead of suppressed-carrier modulation. The carrier amplitude is `carramp`.

Examples

The input massage is x = sin(2*pi*t*fs)+cos(2*pi*t*fs) and fc=20 ,fs=100 draw the AM for it

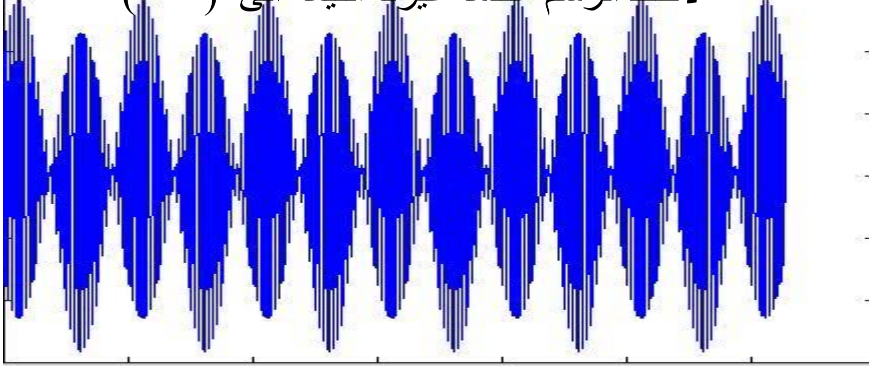
يجب ان تخزن وقت الدالة بمصفوفة عمودية والا لاتحل لان طبيعة دالة (Ammod) تتعامل مع مصفوفة عمودية

```
Fs = 100;
t = [0:pi/250:2*pi]' ;
Fc = 20; % Carrier frequency
x = sin(2*pi*t)+cos(2*pi*t); % Sinusoidal signal
% Modulate x using single- and double-sideband AM.
ydouble = ammod(x,Fc,Fs);
plot(t,ydouble)
```



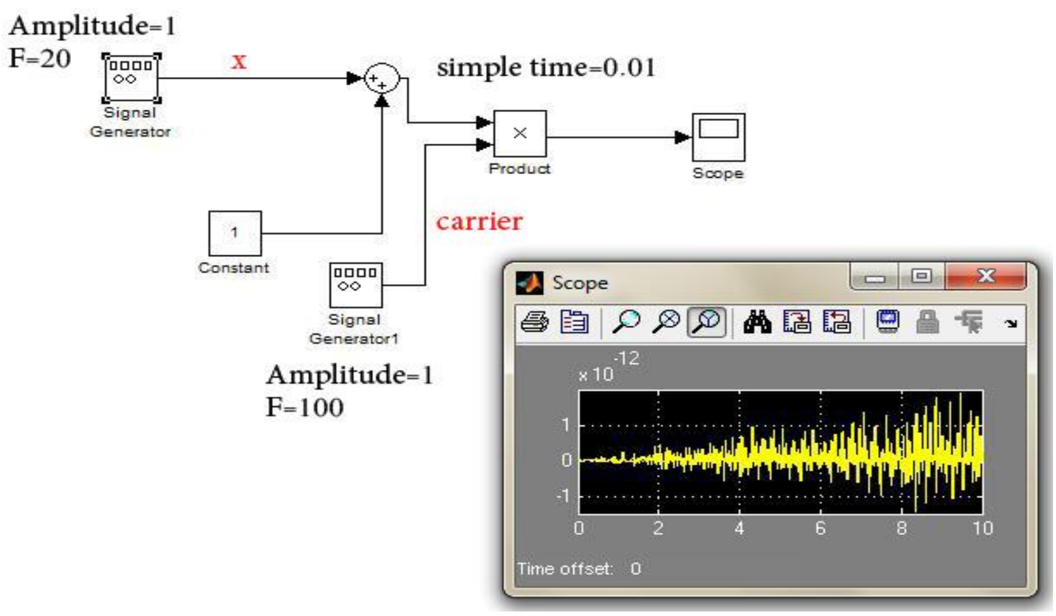
جرب تغير القيمة (٢٥٠)الموجودة بمصفوفة الوقت التغير الى (٣٥٠) ومرة الى (١٥٠) وغيره من القيم وشاهد تغير الرسم

لاحظ الرسم عندما غيرنا القيمة الى (٥٥٠)



سبب تغير الرسم هوا اكثرنا من عدد النقاط ضمن فترة الوقت للدالة المرسلة لذالك يزداد تقارب النقاط

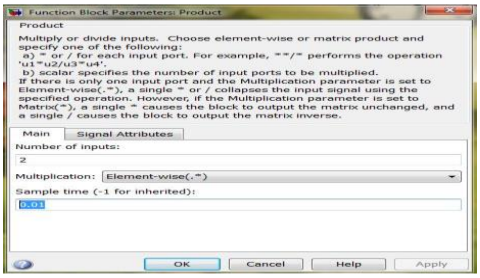
Solve this input signal AM by simulink x=sin(t) (fc=100,fs=20 A=1)



لاحض كيف نغير(simple time) لبلوك (product) ونغير القيمة ماذا تعني هذه القيمة؟

تعني فترة ضرب كل نقطتيين في دالتيي المرسلة والحاملة أي بعد كم نقطة يضرب النقاط المتقابلة في الموقع على المحور (X)

غير القيمه وشاهد تغير الرسم .تلاحظ كلما نقلل القيمة تزداد التداخل حدة



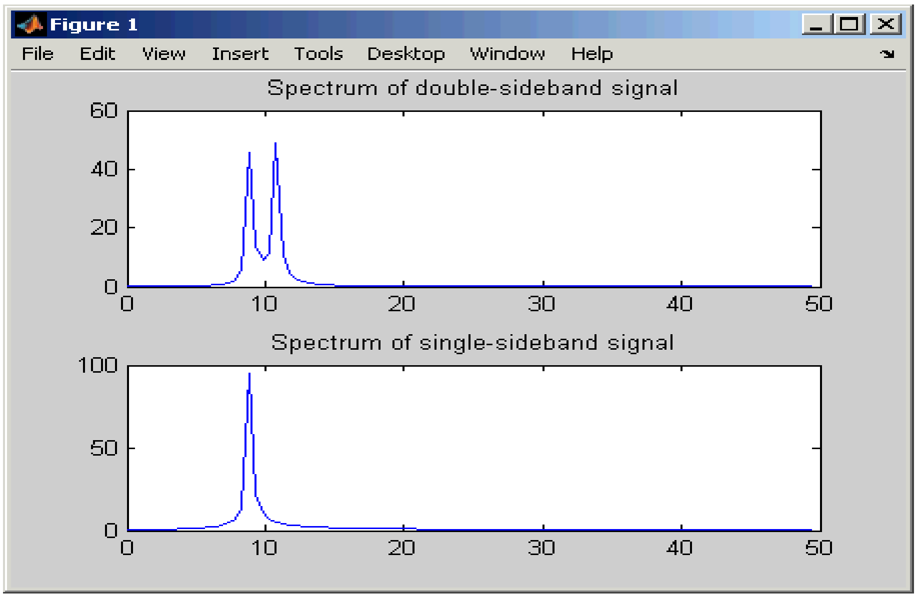
M-file for this function

```
Fs = 100;
t = [0:0.1:2*Fs+1]'/Fs;
Fc = 20; % Carrier frequency
x =  sin(t); % Sinusoidal signal
ydouble = ammod(x,Fc,Fs);
plot(t,ydouble)
```

فسر سبب تغير الشدة في الرسم للعلم لم نغير شيى والرسم نفسه ما الذي تغير اذن؟
الذي تغير هوا اكثرنا من عدد النقط المتداخلة

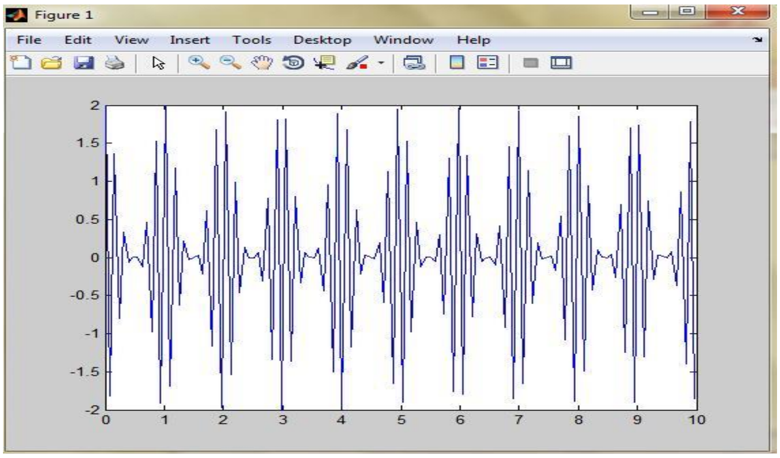
The example below compares double-sideband and single-sideband amplitude modulation.

```
% Sample the signal 100 times per second, for 2 seconds.
Fs = 100;
t = [0:2*Fs+1]'/Fs;
Fc = 10; % Carrier frequency
x = sin(2*pi*t); % Sinusoidal signal

% Modulate x using single- and double-sideband AM.
ydouble = ammod(x,Fc,Fs);
ysingle = ssbmod(x,Fc,Fs);

% Compute spectra of both modulated signals.
zdouble = fft(ydouble);
zdouble = abs(zdouble(1:length(zdouble)/2+1));
frqdouble = [0:length(zdouble)-1]*Fs/length(zdouble)/2;
zsingle = fft(ysingle);
zsingle = abs(zsingle(1:length(zsingle)/2+1));
frqsingle = [0:length(zsingle)-1]*Fs/length(zsingle)/2;

% Plot spectra of both modulated signals.
figure;
subplot(2,1,1); plot(frqdouble,zdouble);
title('Spectrum of double-sideband signal');
subplot(2,1,2); plot(frqsingle,zsingle);
title('Spectrum of single-sideband signal');
```

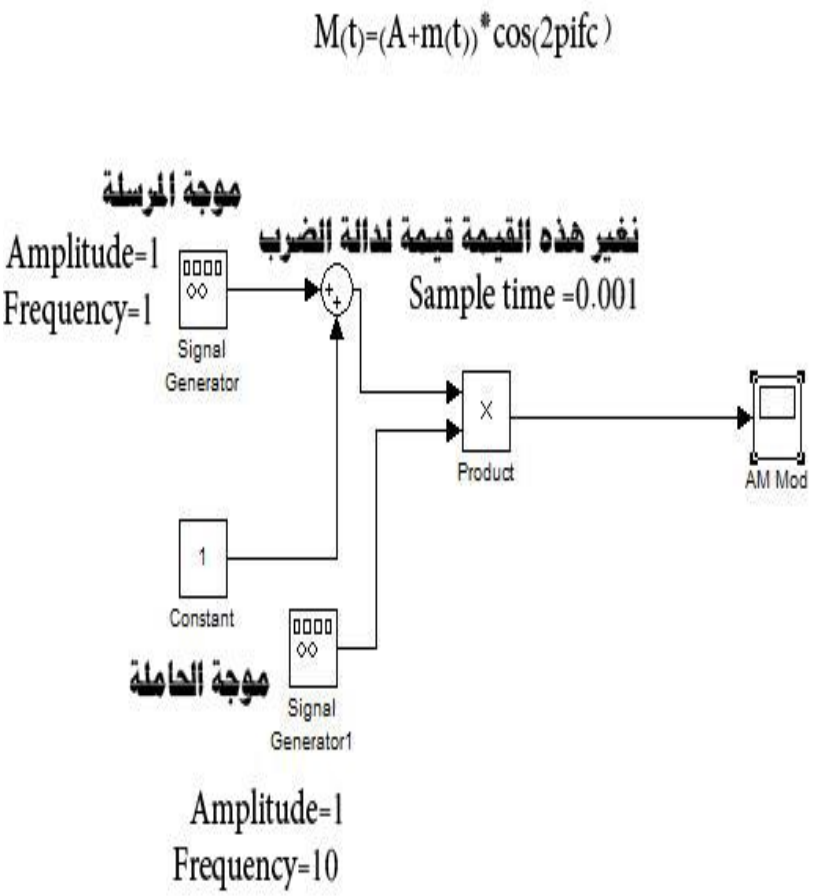


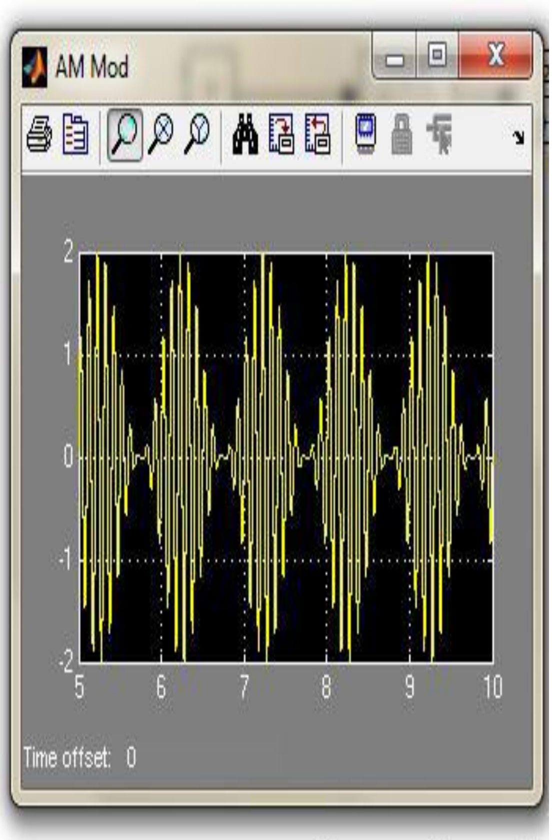
بناء تضمين AM بواسطة الكود

```
clc
clear
fm=1;
fc=1000;
am=1;
t=0:pi/40:10;
mt=cos(2*pi*fm*t)
mc=cos(2*pi*fc*t)
Fsend=(am+mt).* mc
plot(t, Fsend)
```



تمثيله بل simulink





Hussien Ahmmed

Amplitude demodulation

Hussien Ahmmed

Syntax

```
z = amdemod(y,Fc,Fs)
z = amdemod(y,Fc,Fs,ini_phase)
z = amdemod(y,Fc,Fs,ini_phase,carramp)
z = amdemod(y,Fc,Fs,ini_phase,carramp,num,den)
```

Description

`z = amdemod(y,Fc,Fs)` demodulates the amplitude modulated signal `y` from a carrier signal with frequency `Fc` (Hz). The carrier signal and `y` have sample frequency `Fs` (Hz). The modulated signal `y` has zero initial phase and zero carrier amplitude, so it represents suppressed carrier modulation. The demodulation process uses the lowpass filter specified by `[num,den] = butter(5,Fc*2/Fs)`.

> **Note** The `Fc` and `Fs` arguments must satisfy `Fs` > 2(`Fc` + `BW`), where `BW` is the bandwidth of the original signal that was modulated.

`z = amdemod(y,Fc,Fs,ini_phase)` specifies the initial phase of the modulated signal in radians.

`z = amdemod(y,Fc,Fs,ini_phase,carramp)` demodulates a signal that was created via transmitted carrier modulation instead of suppressed carrier modulation. `carramp` is the carrier amplitude of the modulated signal.

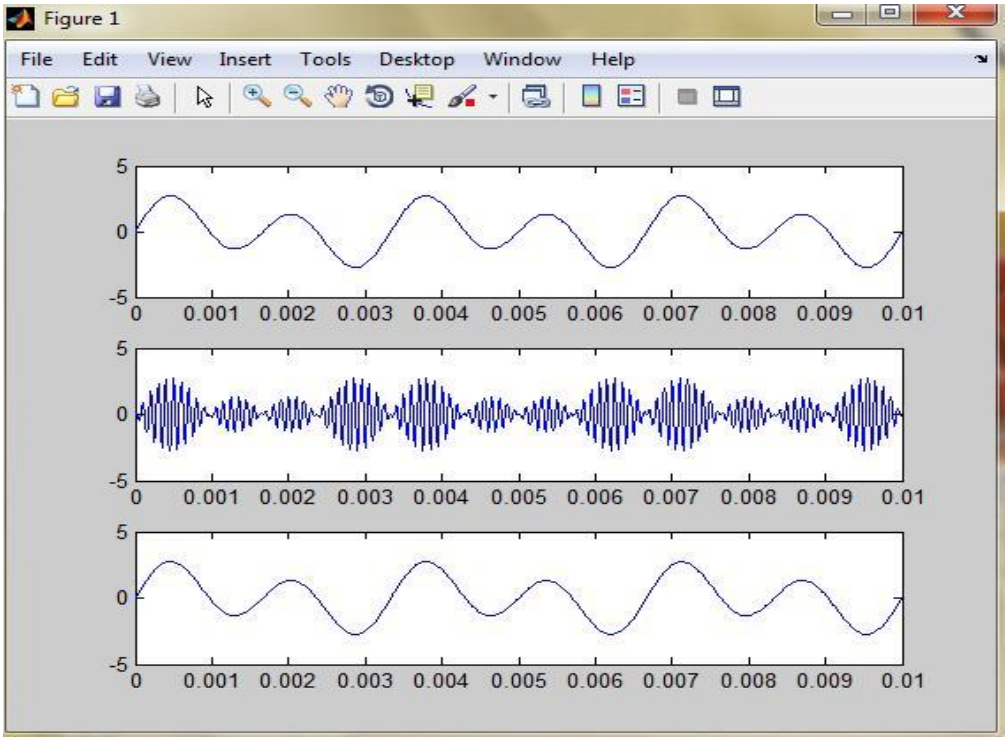
`z = amdemod(y,Fc,Fs,ini_phase,carramp,num,den)` specifies the numerator and denominator of the lowpass filter used in the demodulation.

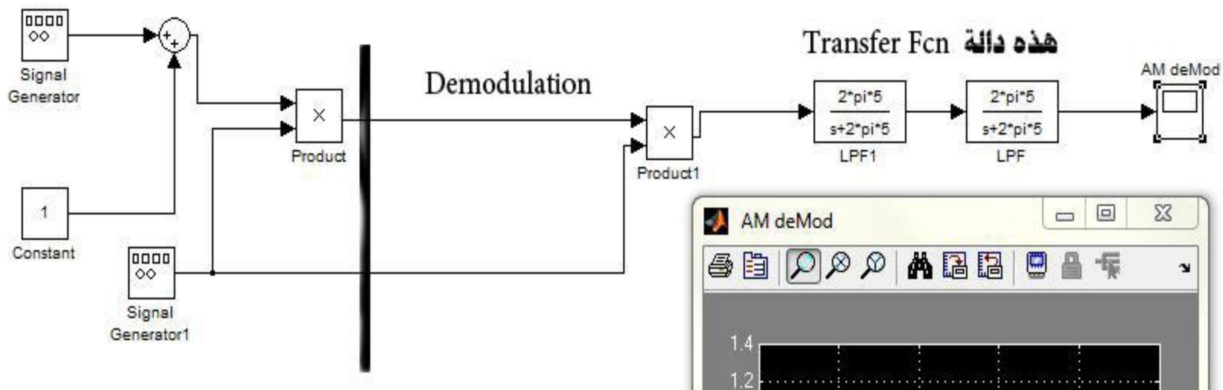
Examples

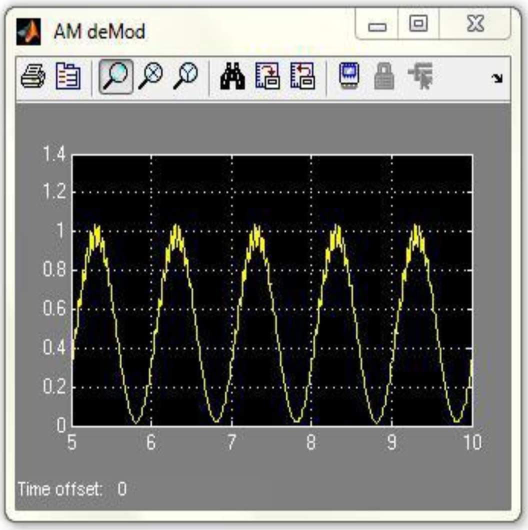
The code below illustrates the use of a nondefault filter.

```
t = .01;
Fc = 10000; Fs = 80000;
t = [0:1/Fs:0.01]';
s = sin(2*pi*300*t)+2*sin(2*pi*600*t); % Original signal
 [num,den] = butter(10,Fc*2/Fs); % Lowpass filter
y1 = ammod(s,Fc,Fs); % Modulate.
s1 = amdemod(y1,Fc,Fs,0,0,num,den); % Demodulate.
subplot(3,1,1)
plot(t,s) %  befor modlating
 subplot(3,1,2)
plot(t,y1)  %  after modlating
subplot(3,1,3)
plot(t,s1)  %  after De modlating
```

لاحظ الرسم كيف يضمنها وعند استلامها يعيد فكها فتعود كما ارسلت قبل التحميل







لاتنسى تغير قيم فترة ضرب كل نقطتيين في بلوك الضرب الى 0.001

Double side band-sc

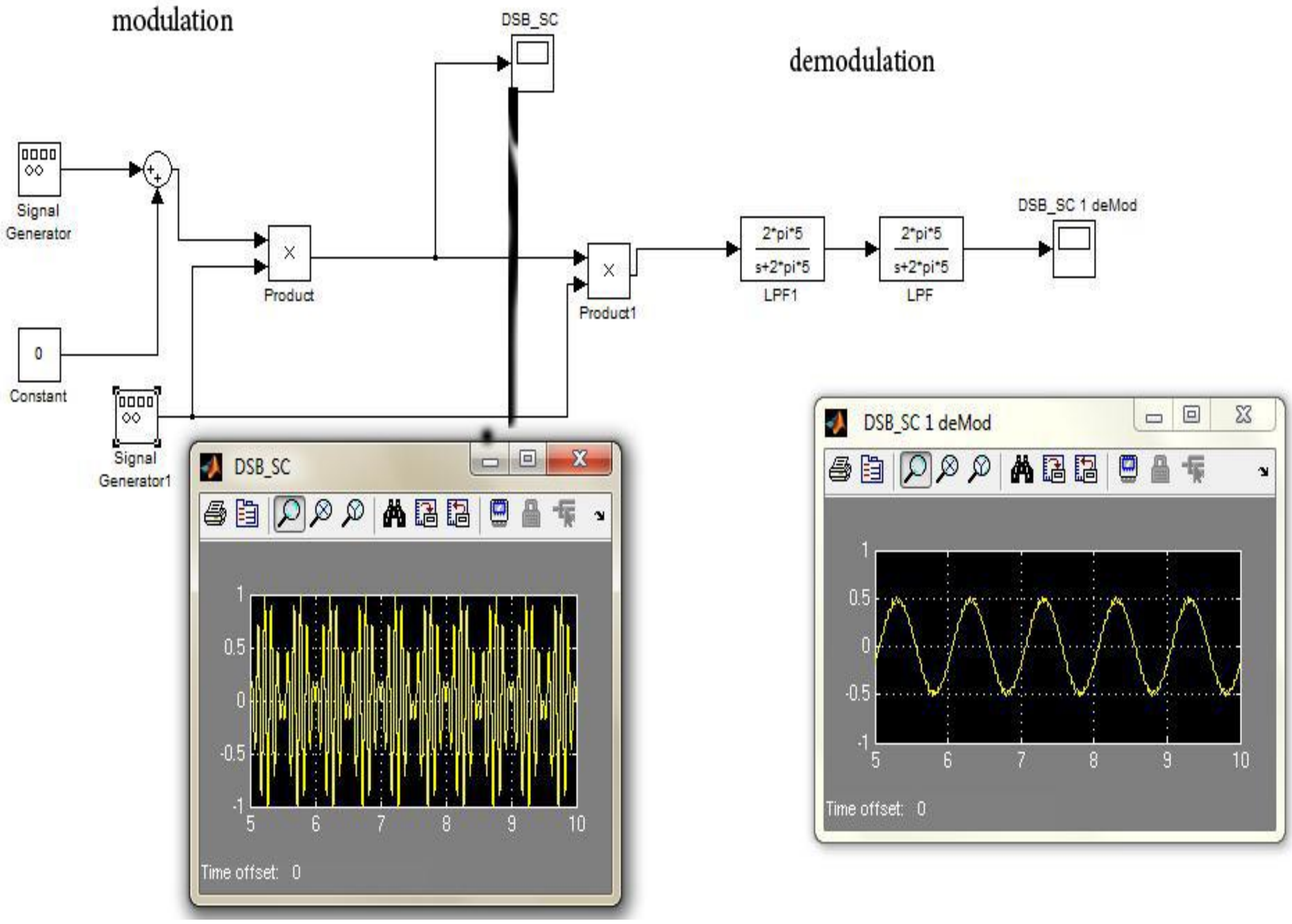
*f(t)=m(t)*cos(2*pi*fc)*

يختلف عن سابقة فقط A=0

بناء عملية بواسطة الكود DSB_sc

```
clc
clear
fm=1;
fc=1000;
A=0;
t=0:pi/40:10;
mt=cos(2*pi*fm*t)
mc=cos(2*pi*fc*t)
Fsend=(A+mt).*mc
plot(t, Fsend)
```

تمثيله بل simulink



Frequency modulation

Syntax

```
y = fmmod(x,Fc,Fs,freqdev)
y = fmmod(x,Fc,Fs,freqdev,ini_phase)
```

Description

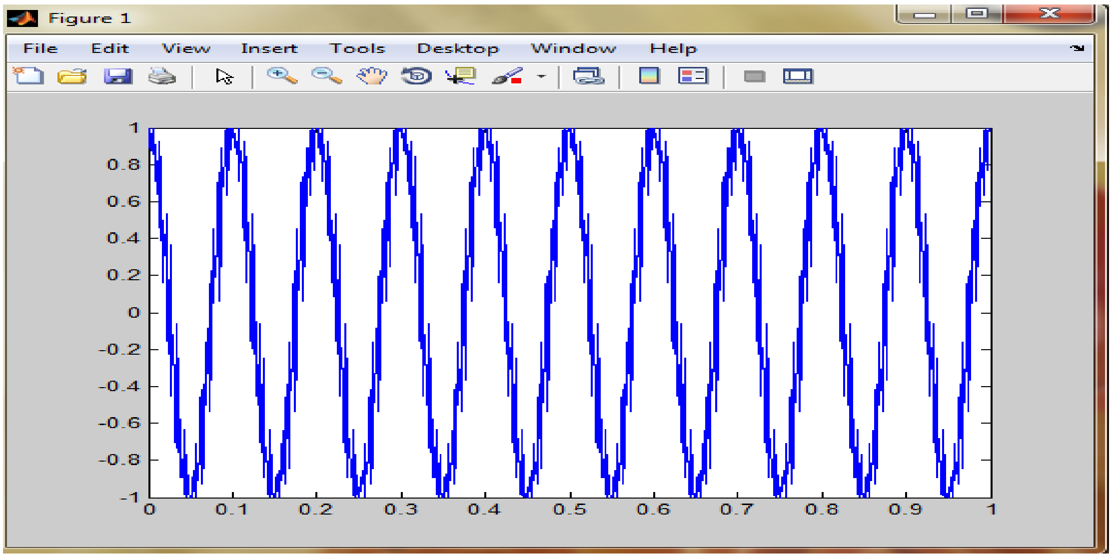
`y = fmmod(x,Fc,Fs,freqdev)` modulates the message signal `x` using frequency modulation. The carrier signal has frequency `Fc` (Hz) and sampling rate `Fs` (Hz), where `Fs` must be at least `2*Fc`. The `freqdev` argument is the frequency deviation constant (Hz) of the modulated signal.

`y = fmmod(x,Fc,Fs,freqdev,ini_phase)` specifies the initial phase of the modulated signal, in radians.

Examples

Example: let input signal is
x = sin(2*pi*300*t)+2*sin(2*pi*600 *t)
and (fc=10,fs=100 , Frequency deviation=50) find FM modulated

```
Fs = 1000; % Sampling rate of signal
Fc = 10; % Carrier frequency
t = [0:Fs]'/Fs; % Sampling times
x = sin(2*pi*300*t)+2*sin(2*pi*600*t);
dev = 50; % Frequency deviation in modulated signal
y = fmmod(x,Fc,Fs,dev); % Modulate both channels.
plot(t,y)
```



The code below modulates a multichannel signal using `fmmod` and demodulates it

Fs = 8000; % Sampling rate of signal

```
Fc = 3000; % Carrier frequency
t = [0:Fs]'/Fs; % Sampling times
s1 = sin(2*pi*300*t)+2*sin(2*pi*600*t); % Channel 1
s2 = sin(2*pi*150*t)+2*sin(2*pi*900*t); % Channel 2
x = [s1,s2]; % Two-channel signal
dev = 50; % Frequency deviation in modulated signal
y = fmmod(x,Fc,Fs,dev); % Modulate both channels.
z = fmdemod(y,Fc,Fs,dev); % Demodulate both channels.
```

Frequency demodulation

Syntax

```
z = fmdemod(y,Fc,Fs,freqdev)
z = fmdemod(y,Fc,Fs,freqdev,ini_phase)
```

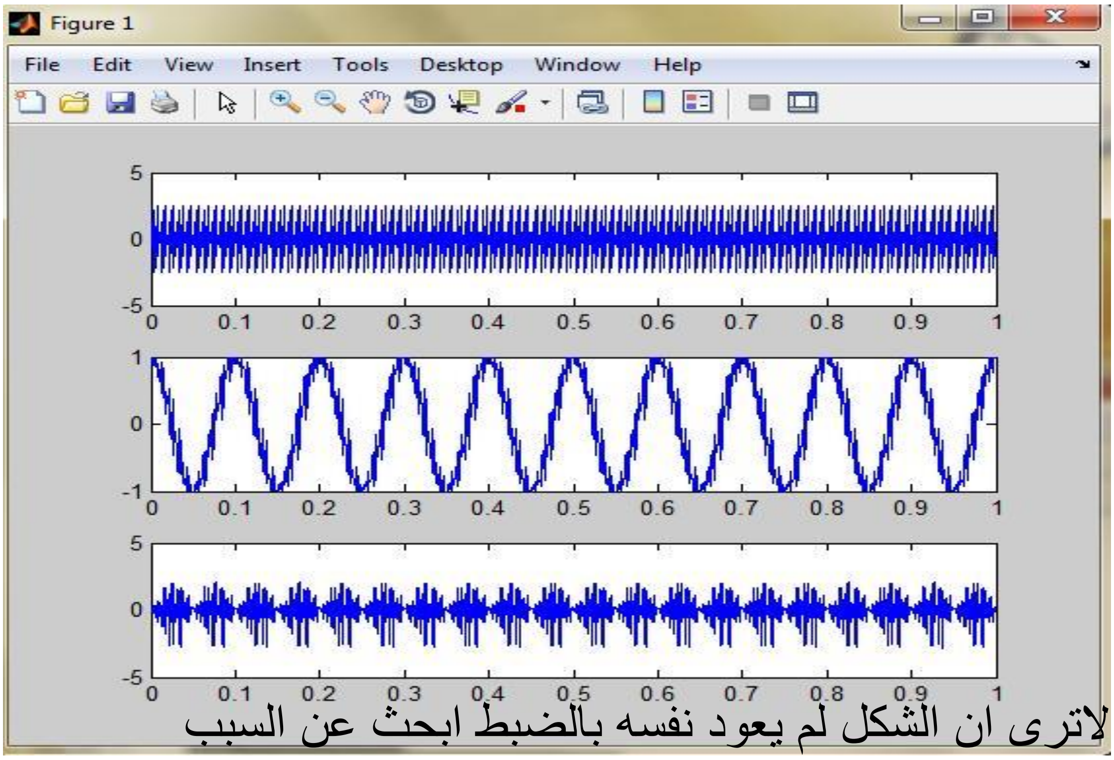
Description

`z = fmdemod(y,Fc,Fs,freqdev)` demodulates the modulating signal `z` from the carrier signal using frequency demodulation. The carrier signal has frequency `Fc` (Hz) and sampling rate `Fs` (Hz), where `Fs` must be at least `2*Fc`. The `freqdev` argument is the frequency deviation (Hz) of the modulated signal `y`.

`z = fmdemod(y,Fc,Fs,freqdev,ini_phase)` specifies the initial phase of the modulated signal, in radians.

Example: signal that input to system is
x = sin(2*pi*300*t)+2*sin(2*pi*600*t) that (fc=10,fs=1000)
find Fm mod and Demod

```
Fs = 1000; % Sampling rate of signal
Fc = 10; % Carrier frequency
t = [0:Fs]'/Fs; % Sampling times
x = sin(2*pi*300*t)+2*sin(2*pi*600*t); % Channel 1
dev = 50; % Frequency deviation in modulated signal
y = fmmod(x,Fc,Fs,dev); % Modulate both channels.
yd =  fmdemod(y,Fc,Fs,dev  ); % Modulate both channels.
subplot(3,1,1)
plot(t,x) %  befor modlating
 subplot(3,1,2)
plot(t,y)  %  after modlating
subplot(3,1,3)
plot(t,yd)  %  after De modulating
```



الاترى ان الشكل لم يعود نفسه بالضبط ابحث عن السبب

Phase modulation

Syntax

```
y = pmmod(x,Fc,Fs,phasedev)
y = pmmod(x,Fc,Fs,phasedev,ini_phase)
```

Description

`y = pmmod(x,Fc,Fs,phasedev)` modulates the message signal `x` using phase modulation. The carrier signal has frequency `Fc` (hertz) and sampling rate `Fs` (hertz), where `Fs` must be at least `2*Fc`. The `phasedev` argument is the phase deviation of the modulated signal in radians.

`y = pmmod(x,Fc,Fs,phasedev,ini_phase)` specifies the initial phase of the modulated signal in radians.

Phase demodulation

Syntax

```
z = pmdemod(y,Fc,Fs,phasedev)
z = pmdemod(y,Fc,Fs,phasedev,ini_phase)
```

Description

`z = pmdemod(y,Fc,Fs,phasedev)` demodulates the phase-modulated signal `y` at the carrier frequency `Fc` (hertz). `z` and the carrier signal have sampling rate `Fs` (hertz), where `Fs` must be at least `2*Fc`. The `phasedev` argument is the phase deviation of the modulated signal, in radians.

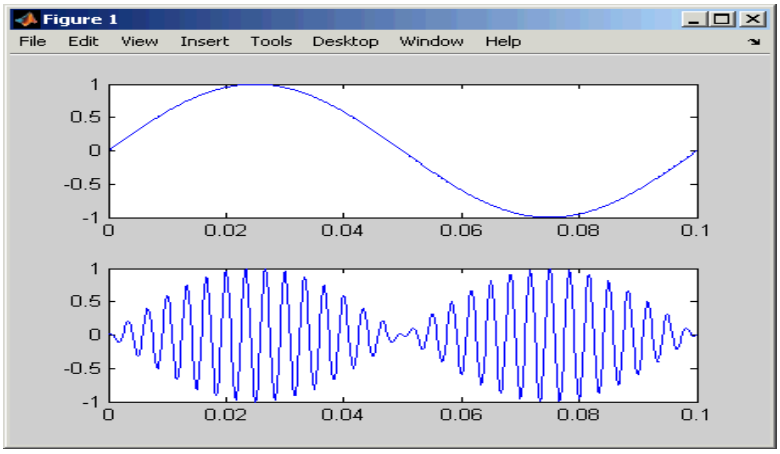
`z = pmdemod(y,Fc,Fs,phasedev,ini_phase)` specifies the initial phase of the modulated signal, in radians.

Representing Analog Signals

To modulate an analog signal using this toolbox, start with a real message signal and a sampling rate `Fs` in hertz. Represent the signal using a vector `x`, the entries of which give the signal's values in time increments of 1/`Fs`. Alternatively, you can use a matrix to represent a multichannel signal, where each column of the matrix represents one channel.

For example, if `t` measures time in seconds, then the vector `x` below is the result of sampling a sine wave 8000 times per second for 0.1 seconds. The vector `y` represents the modulated signal.

```
Fs = 8000; % Sampling rate is 8000 samples per second.
Fc = 300; % Carrier frequency in Hz
t = [0:.1*Fs]'/Fs; % Sampling times for .1 second
x = sin(20*pi*t); % Representation of the signal
y = ammod(x,Fc,Fs); % Modulate x to produce y.
figure;
subplot(2,1,1); plot(t,x); % Plot x on top.
subplot(2,1,2); plot(t,y)% Plot y below.
```



As a multichannel example, the code below defines a two-channel signal in which one channel is a sinusoid with zero initial phase and the second channel is a sinusoid with an initial phase of `pi/8`.

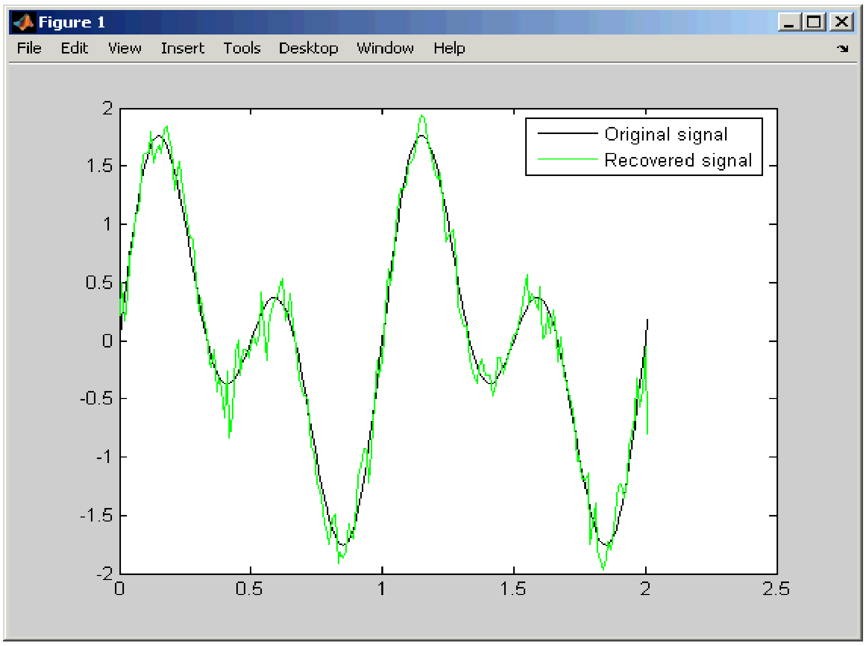
```
Fs = 8000;
t = [0:.1*Fs]'/Fs;
x = [sin(20*pi*t), sin(20*pi*t+pi/8)];
```

Analog Modulation Example

This example illustrates the basic format of the analog modulation and demodulation functions. Although the example uses phase modulation, most elements of this example apply to other analog modulation techniques as well.

The example samples an analog signal and modulates it. Then it simulates an additive white Gaussian noise (AWGN) channel, demodulates the received signal, and plots the original and demodulated signals.

```
% Prepare to sample a signal for two seconds,
% at a rate of 100 samples per second.
Fs = 100; % Sampling rate
t = [0:2*Fs+1]'/Fs; % Time points for sampling

% Create the signal, a sum of sinusoids.
x = sin(2*pi*t) + sin(4*pi*t);

Fc = 10; % Carrier frequency in modulation
phasedev = pi/2; % Phase deviation for phase modulation

y = pmmod(x,Fc,Fs,phasedev); % Modulate.
y = awgn(y,10,'measured',103); % Add noise.
z = pmdemod(y,Fc,Fs,phasedev); % Demodulate.

% Plot the original and recovered signals.
figure; plot(t,x,'k-',t,z,'g-');
legend('Original signal','Recovered signal');
```



ssbmod - Single sideband amplitude modulation

Syntax

```
y = ssbmod(x,Fc,Fs)
y = ssbmod(x,Fc,Fs,ini_phase)
y = ssbmod(x,fc,fs,ini_phase,'upper')
```

Description

`y = ssbmod(x,Fc,Fs)` uses the message signal `x` to modulate a carrier signal with frequency `Fc` (Hz) using single sideband amplitude modulation in which the lower sideband is the desired sideband. The carrier signal and `x` have sample frequency `Fs` (Hz). The modulated signal has zero initial phase.

`y = ssbmod(x,Fc,Fs,ini_phase)` specifies the initial phase of the modulated signal in radians.

`y = ssbmod(x,fc,fs,ini_phase,'upper')` uses the upper sideband as the desired sideband.

ssbdemod - Single sideband amplitude demodulation

Syntax

```
z = ssbdemod(y,Fc,Fs)
z = ssbdemod(y,Fc,Fs,ini_phase)
z = ssbdemod(y,Fc,Fs,ini_phase,num,den)
```

Description

For All Syntaxes

`z = ssbdemod(y,Fc,Fs)` demodulates the single sideband amplitude modulated signal `y` from the carrier signal having frequency `Fc` (Hz). The carrier signal and `y` have sampling rate `Fs` (Hz). The modulated signal has zero initial phase, and can be an upper- or lower-sideband signal. The demodulation process uses the lowpass filter specified by `[num,den] = butter(5,Fc*2/Fs)`.

> **Note** The `Fc` and `Fs` arguments must satisfy `Fs` > 2(`Fc` + `BW`), where `BW` is the bandwidth of the original signal that was modulated.

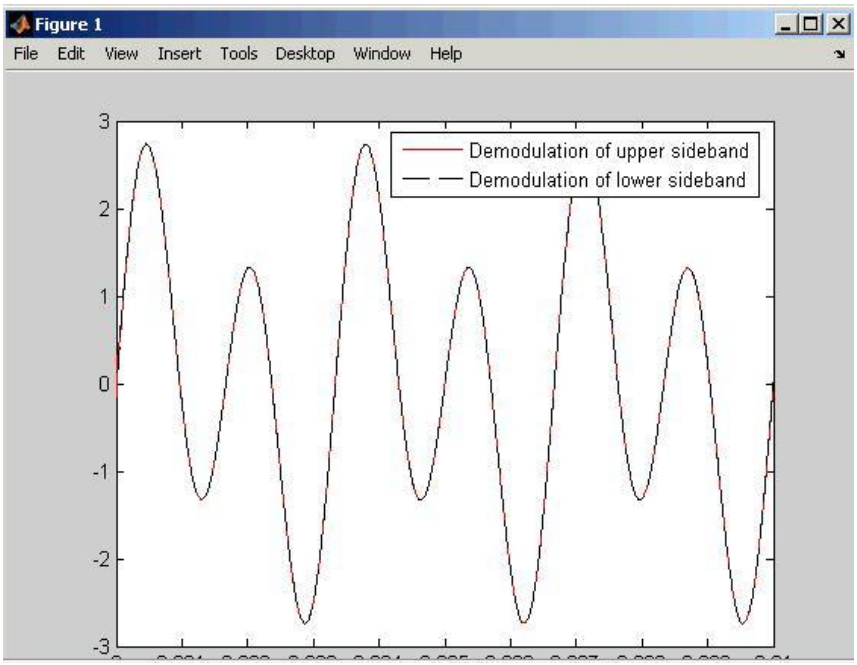
`z = ssbdemod(y,Fc,Fs,ini_phase)` specifies the initial phase of the modulated signal in radians.

`z = ssbdemod(y,Fc,Fs,ini_phase,num,den)` specifies the numerator and denominator of the lowpass filter used in the demodulation.

Examples

The code below shows that `ssbdemod` can demodulate an upper-sideband or lower-sideband signal.

```
Fc = 12000; Fs = 270000;
t = [0:1/Fs:0.01]';
s = sin(2*pi*300*t)+2*sin(2*pi*600*t);
y1 = ssbmod(s,Fc,Fs,0); % Lower-sideband modulated signal
y2 = ssbmod(s,Fc,Fs,0,'upper'); % Upper-sideband modulated signal
s1 = ssbdemod(y1,Fc,Fs); % Demodulate lower sideband
s2 = ssbdemod(y2,Fc,Fs); % Demodulate upper sideband
% Plot results to show that the curves overlap.
figure; plot(t,s1,'r-',t,s2,'k--');
legend('Demodulation of upper sideband','Demodulation of lower
sideband')
```



حسين احمد طالب الربيعي

هندسة البرامجيات

المرحلة الثالثة

Hussien Ahhmed