

Maestría en Ingeniería Eléctrica especialización Telecomunicaciones

Comunicaciones Digitales

**Tarea #2**

***Luis Emilio Tonix Gleason***

***Fernando Alberto Madera Torres***

*26/03/2022*

***Dr. Ramon Michel Parra***

Tabla de contenido

[**Ejercicio 1 3**](#_Toc99536334)

[Modulación AM Convencional 4](#_Toc99536335)

[Modulación AM Doble Banda Lateral con Portadora Suprimida (DBLPS) 5](#_Toc99536336)

[Modulación BLU 7](#_Toc99536337)

[Demodulación Coherente Vs No-Coherente 9](#_Toc99536338)

[Demodulación Envolvente Señal AM 9](#_Toc99536339)

[Demodulación Coherente Señal AM con Filtrado 12](#_Toc99536340)

[Pasa banda Carrier (Fase Cerrada de Costas) 12](#_Toc99536341)

[Demodulación Coherente DBLPS con Filtrado pasa bajas (Fase Cerrada de Costas) 12](#_Toc99536342)

[**Ejercicio 2 14**](#_Toc99536343)

[Modulación FM 14](#_Toc99536344)

[Definiciones 14](#_Toc99536345)

[Modulación por tono 14](#_Toc99536346)

[Demodulacion 16](#_Toc99536347)

# Ejercicio 1

Genere una señal senoidal portadora de 20Hz y una señal de información cosenoidal moduladora de 2 Hz.

a. Utilice ambas señales para generar una señal modulada en DBLPS, AM convencional con índice de modulación= .85, y BLU mediante filtrado. Obtenga las señales resultantes en tiempo y frecuencia de manera analítica y verifique sus resultados mediante gráficas obtenidas por programas en Matlab (gráficas con ejes temporales y frecuenciales correctos). Comente sus observaciones.

b. Explique la selección del tiempo de observación e intervalo de muestreo para poder graficar de manera adecuada las señales moduladas.

c. Realice decodificación coherente para todos los casos, de manera analítica y por simulación y compare los resultados obtenidos. Comente sus observaciones.

d. Realice decodificación por envolvente de la señal AM convencional de manera analítica y mediante simulación obtenida por programa de Matlab y compare los resultados obtenidos. Comente sus observaciones.

e. Realice decodificación coherente de la señal AM mediante filtrado pasa banda de la portadora y retroalimentada al demodulador, de manera analítica y mediante simulaciones de programas de Matlab. Compare los resultados y comente sus observaciones.

f. Para la señal de DBLPS, utilice un receptor con frecuencia de oscilación de 20.1Hz. y 20.5Hz, y realice un filtrado pasa bajas de la señal recibida de manera analítica y por simulación. Comente sobre la señal resultante.

%Parametros

kps\_64 = 120; %Muestreo Nyquist 4 veces la freq de la portadora

time = 0:1/kps\_64:1; %Tiempo de Muestreo

time\_am = -.5:1/120:.5;

%señal Portadora

ap = 2;

freq\_port = 20; %Frecuencia de Portadora

wp = 2\*pi\*freq\_port;

port\_sig = ap\*cos(wp\*time);

%señal moduladora

am = 1.7;

freq\_mod = 2; %Frecuencia de Moduladora

wm = 2\*pi\*freq\_mod;

mod\_sig = am\*cos(wm\*time); %porque + tiempo y no \*tiempo?

% Señales en tiempo

figure(1);

plot(time, mod\_sig)

hold on

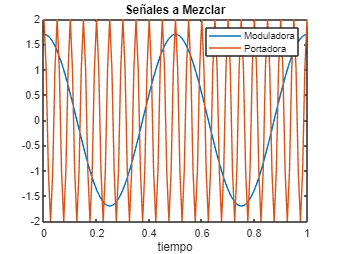
plot(time, port\_sig)

title('Señales a Mezclar');

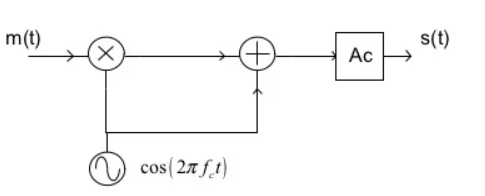
xlabel('tiempo')

legend({'Moduladora','Portadora'},'Location','northeast')

hold off



# Modulación AM Convencional



El índice de modulación  esta dado por la siguiente relación



donde **Am** es la amplitud de la señal moduladora y **Ap** es la amplitud de la señal portadora.

index\_mod = am/ap;

% Multiplicador de Señal Portadora y Modulada

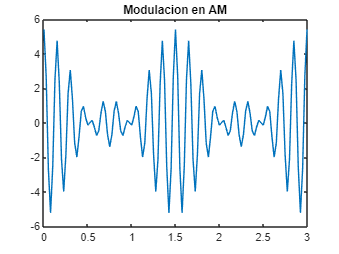
%am\_sig = am\*mod\_sig.\*port\_sig + ap\*port\_sig;

am\_sig = port\_sig.\*mod\_sig + port\_sig;

figure(2);

plot(time\*3, am\_sig);

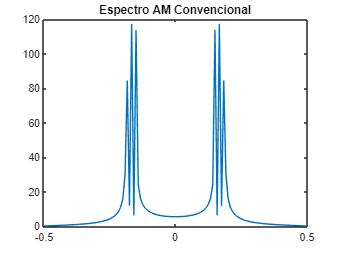
title('Modulacion en AM');



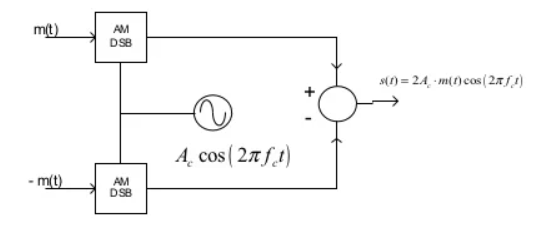
figure(3);

plot(time\_am,abs(ttof(am\_sig)));

title('Espectro AM Convencional');



# Modulación AM Doble Banda Lateral con Portadora Suprimida (DBLPS)



Primero necesitamos desfasar la señal moduladora 180° para tenerla invertida, entonces la multiplicamos una la portadora y al final sumamos las diferencias de las señales para tener la señal deseada

Texto

Descripción generada automáticamente

time\_dsb = -.5:1/120:.5; %Tiempo de Muestreo

% Inversion de Moduladora 180°

wm = (pi).\*freq\_mod;

mod\_sig\_180 = am.\*-cos(wm\*time);

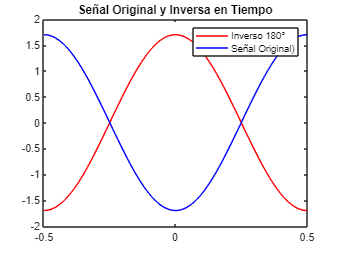
mod\_sig\_0 = am.\*cos((wm)\*time);

figure(4)

plot(time\_dsb,mod\_sig\_180,'r',time\_dsb,mod\_sig\_0,'b');

legend('Inverso 180°','Señal Original)');

title('Señal Original y Inversa en Tiempo');



% Modulador Balanceado 1

mod\_bal\_01 = mod\_sig\_0 .\* port\_sig;

%Modulador Balanceado 2

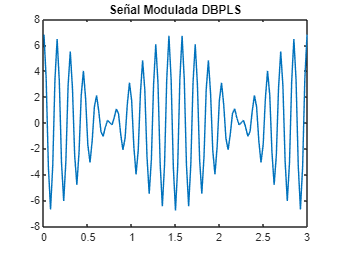
mod\_bal\_02 = mod\_sig\_180 .\* port\_sig;

% Diferencia de Señales

sig\_am\_dsbsc = mod\_bal\_01 - mod\_bal\_02;

plot(time\*3,sig\_am\_dsbsc);

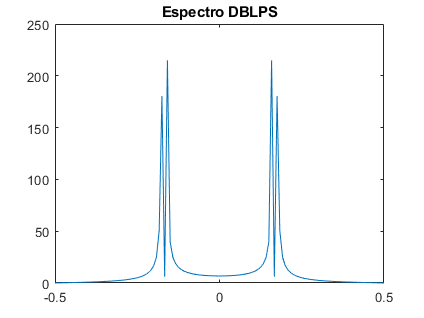
title('Señal Modulada DBPLS');



figure(8)

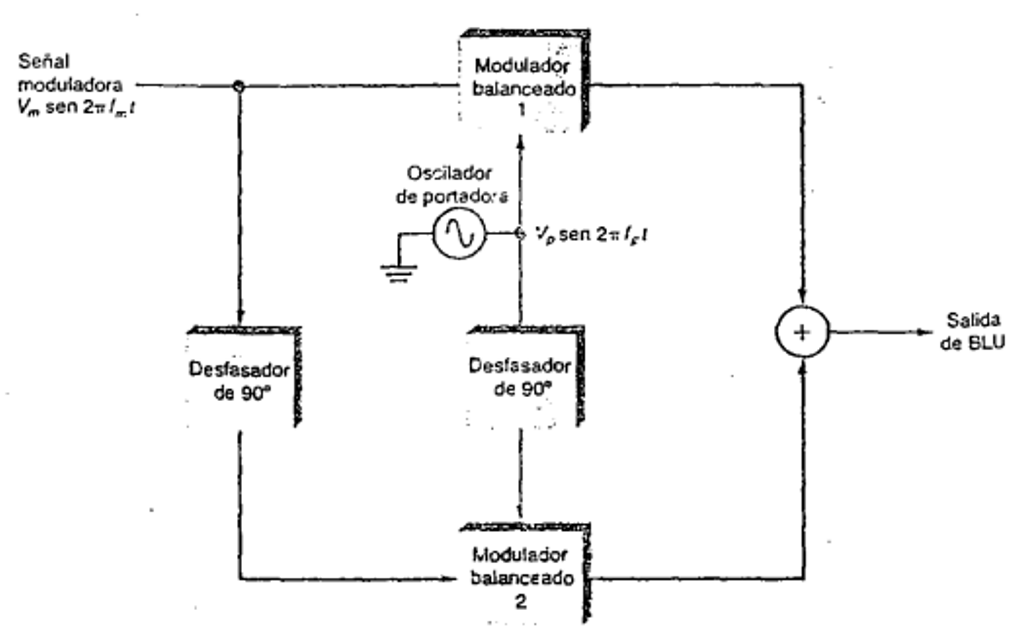
plot(time\_dsb,abs(ttof(sig\_am\_dsbsc)));

title('Espectro DBLPS');



# Modulación BLU

Para desfasar una señal antes de que entre al modulador que solo es un multiplicador de señal hay que utilizar por simplicidad la función de Matlab delayseq, para adelantar o atrasar 90° en tiempo la señal modulada original.



Si bien se puede derivar la señal cos para tener un -seno ósea atrasar la señal 90°, si aplico un integrador el coseno me adelanta la señal 90°

% Desface de Moduladora 90°

wm = 2\*pi\*freq\_mod;

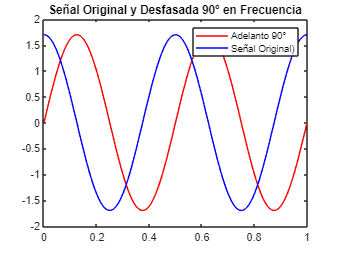
mod\_sig\_90 = am\*sin(wm\*time); % señal adelantada 90°---->porque + tiempo y no \*tiempo?

figure(4)

plot(time,mod\_sig\_90,'r',time,mod\_sig,'b');

legend('Adelanto 90°','Señal Original)');

title('Señal Original y Desfasada 90° en Frecuencia');



% Desface de Portadora 90°

wp = 2\*pi.\*freq\_port;

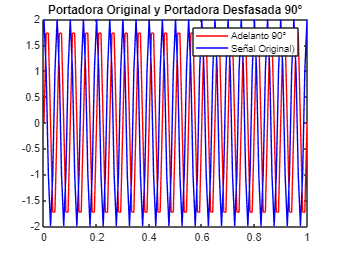
port\_sig\_90 = ap.\*sin(wp.\*time); % señal adelantada 90°---->porque + tiempo y no \*tiempo?

figure (5);

plot(time,port\_sig\_90,'r',time,port\_sig,'b');

legend('Adelanto 90°','Señal Original)');

title('Portadora Original y Portadora Desfasada 90°');



El primer modulador balanceado lo único que hace es multiplicar la señal moduladora con la señal de la portadora original.

time\_blu = -.5:1/120:.5; %Tiempo de Muestreo

% Modulador balanceado 1

mod\_bal\_1 = am\_sig;

% Modulador Balanceado 2

mod\_bal\_2 = mod\_sig\_90.\*port\_sig\_90;

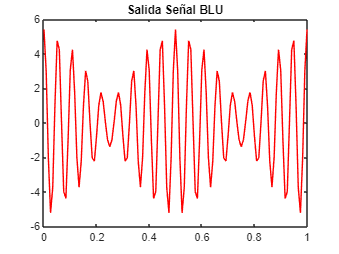
% Sumador resultante BLU

BLU\_out = mod\_bal\_1 + mod\_bal\_2;

figure (6);

plot(time,BLU\_out,'r');

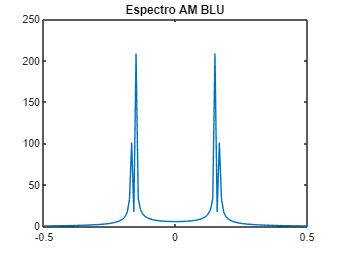
title('Salida Señal BLU');



figure(7);

plot(time\_blu,abs(ttof(BLU\_out)));

title('Espectro AM BLU');



# Demodulación Coherente Vs No-Coherente

Un receptor **Coherente o Síncrono**, es aquel que utiliza frecuencias generadas en el receptor que se usan para la demodulación, se sincronizan con las frecuencias de un Oscilador generadas en el transmisor.

Un Receptor **No Coherente o de Detección de envolvente** porque la información se recupera a partir de la onda recibida detectando la envolvente.

# Demodulación Envolvente Señal AM

Diagrama

Descripción generada automáticamente

Forma

Descripción generada automáticamente con confianza media

Lo que se realiza para la detección no Coherente de una señal es, multiplicarla por si misma para elevar el coseno al cuadrado y así solo se cancele mediante identidades trigonométricas, quedándonos por si sola la Señal con la Información transmitida.

% Señal de entrada Señal AM Convencional Normal

yt = am\_sig .\* am\_sig;

% Filtrado LPF Normal

y\_fil\_sig = lowpass(yt, 4, freq\_port );

plot(time\*3,y\_fil\_sig);

title('señal AM Recuperada por Demodulador No Coherente');

Gráfico, Gráfico de líneas, Histograma

Descripción generada automáticamente

plot(time\_am,abs(ttof(y\_fil\_sig)));

title('Espectro AM con Demodulador No Coherente');

Gráfico, Histograma

Descripción generada automáticamente

# Demodulación Coherente Señal AM con Filtrado

# Pasa banda Carrier

Diagrama

Descripción generada automáticamente con confianza media

1. La señal de recepción es 
2. La señal demodulada es 
3. Existe una propiedad trigonométrica Forma

   Descripción generada automáticamente con confianza media
4. Por lo tanto la señal queda expandida de esta forma Forma

   Descripción generada automáticamente con confianza media
5. Su transformada de Fourier Forma

   Descripción generada automáticamente con confianza media

Forma

Descripción generada automáticamente con confianza media

Imagen que contiene Gráfico

Descripción generada automáticamente

sig\_recover = am\_sig.\*port\_sig;

figure (8);

plot(time,sig\_recover);

title('Señal AM en Receptor');

Gráfico, Histograma

Descripción generada automáticamente

Forma

Descripción generada automáticamente con confianza media

plot(time\_am,abs(ttof(sig\_recover)))

title('Recepcion AM espectro');

Gráfico, Histograma

Descripción generada automáticamente

% Filtrado de Señal Recuperada

filtrof = zeros(1,length(sig\_recover));

%filtrof(find(time\_am==-20):find(time\_am==20))=1; %AQUI QUE HACE?

% multiplicamos por el filtro en el dominio de la frecuencia

sig\_filtered = ttof(sig\_recover).\*filtrof;

%regresamos la señal al dominio del tiempo

%sig\_filtered = real(ttof(sig\_filtered));

plot(time\_am,sig\_filtered);

title('Señal AM Filtrada');

# Demodulación Coherente DBLPS con Filtrado pasa bajas (Fase Cerrada de Costas)

Diagrama

Descripción generada automáticamente

Texto, Carta

Descripción generada automáticamente

Texto

Descripción generada automáticamente

% Señal de entrada Señal DBLPS Normal

rt = sig\_am\_dsbsc;

%Local Oscilator

A\_lo = 2;

wc\_lo = 2\*pi\*freq\_mod;

%vco = filt\_tot\_sig;

l\_osc = A\_lo .\* cos(wc\_lo+time);

%l\_osc = vco;

l\_osc\_90 = A\_lo .\* sin(wc\_lo+time);

%l\_osc\_90 algo que me derive 90° la señal entrante, por lo muentras esto

% Multiplicador Entrada Normal y Señal Retro Normal

if1\_signal\_am = rt .\* l\_osc;

if2\_signal\_am = rt .\* l\_osc\_90;

% Filtrado LPF en frecuencia

filt\_sig1 = lowpass(if1\_signal\_am, 4, freq\_port );

filt\_sig2 = lowpass(if2\_signal\_am, 4, freq\_port );

tot\_sig = filt\_sig1 .\* filt\_sig2;

filt\_tot\_sig = lowpass(tot\_sig, 4, freq\_port );

%regresamos la señal al dominio del tiempo

plot(time,filt\_sig1,'r',time, filt\_sig2,'b', time, filt\_tot\_sig, 'g' );

legend('LPF\_Sig1','LPF\_Sig2', 'Out\_Signal');

title('Señal DBLPS Filtrada');

Gráfico, Histograma

Descripción generada automáticamente

filt\_sig\_tot = real(ftot(filt\_sig1));

plot(time,filt\_sig\_tot);

title('Señal DBLPS Filtrada en Frecuencia');

Gráfico, Histograma

Descripción generada automáticamente

Texto

Descripción generada automáticamente

% Salida Demodulada es la Salida del Primer Filtro LPF

y\_e = filt\_tot\_sig;

plot(time,y\_e );

title('Señal DBLPS Demodulada');

Gráfico, Gráfico de líneas

Descripción generada automáticamente

# Ejercicio 2

Para el mismo caso de las señales utilizadas en la pregunta anterior, repita el procedimiento considerando modulación de frecuencia con índice de modulación β=1.

a. Exprese las ecuaciones que presentan las señales moduladas y presente sólo resultados de simulación. Compare los resultados entre ambos métodos de modulación y comente sus observaciones.

b. Realice recepción de la señal modulada en frecuencia con una de las técnicas de demodulación frecuencial vistas en clase, compare y comente sus resultados.

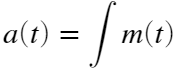
# Modulación FM

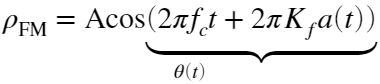
# Definiciones



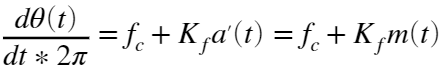








Fase instantánea

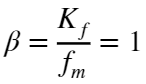


Índice de modulación  para el caso de portadora con amplitud unitaria queda en términos de 

 donde  es la frecuencia máxima de 

# Modulación por tono

Se desea una frecuencia en un rango de  . 

Con la señal moduladora . . Por lo tanto 

%definicion de ejes

time = -2:1/256:2;

freq = -32:1/16:32;

%

fmod = 2; % frecuencia moduladora maxima a 2 hz

kf = 2; % variación

fc = 20; % frecuencia central portadora

wm = 2\*pi\*fmod; % moduladora en rad

%m(t) -> cos(wm\*time)

a\_t = 1/wm\*sin(wm\*time); %a(t) = integral m(t)

theta = 2\*pi\*(fc\*time+kf\*a\_t); %

fm1=cos(theta);

Código para su grafica con puntos máximos en frecuencia

%plot stuff

figure(1);

fm1 = abs(ttof(fm1))/(.5\*length(freq)); % normalizing with observation windows

TF = islocalmax(fm1);

key\_freq = freq(TF);

value\_freq = fm1(TF);

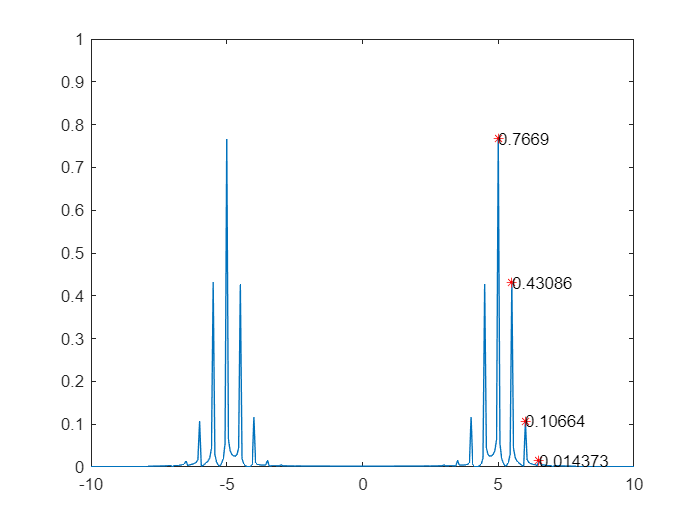
key\_freq = key\_freq(21:24);

value\_freq = value\_freq(21:24);

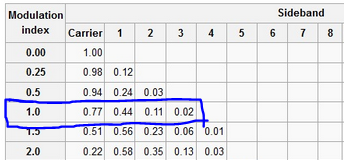
plot(freq,fm1,key\_freq,value\_freq,'r\*')

text(key\_freq,value\_freq,string(value\_freq))

axis([-10 10 0 1])

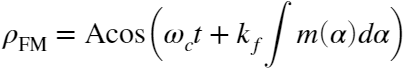


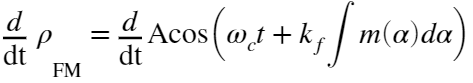
Dicho comportamiento se debe al ancho de banda en FM que esta relacionado con  la cual es la función de bessel y el ancho de banda esta dado por  . Las funciones de bessel pesan a las frecuencias.

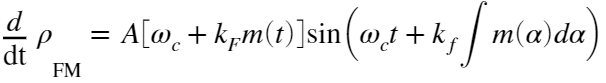


# Demodulación

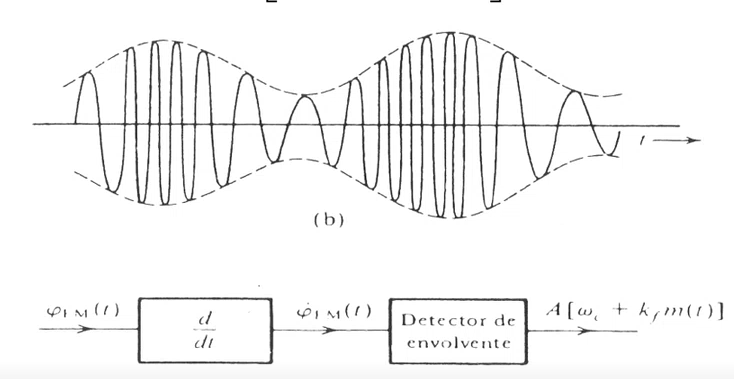
Tenemos que derivar la señal de FM







La señal queda de FM y AM, entonces la paso por un valor al cuadrado o abs para obtener puros valores positivos, la señal quedaría solamente como un detector de envolvente, lo filtro y recupero la señal original.



Para poder derivar de manera mas sencilla y sin tanto gasto de memoria, podemos recordar que la derivación en frecuencia es  por lo que solo queda aplicar un filtro derivador a la señal para que surge el efecto deseado.

filtro\_derivador = zeros(1,length(freq));

filtro\_derivador = 1i\*2\*pi\*freq;

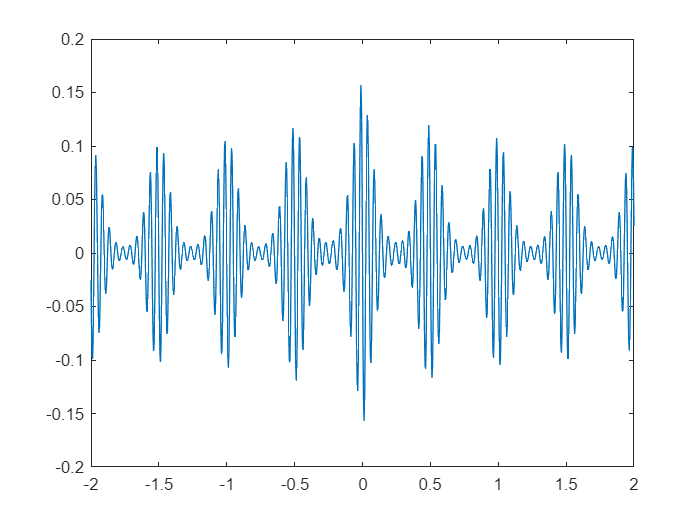
fm1\_df = fm1.\*filtro\_derivador; % señal deriva

fm1\_dt = real(ftot(fm1\_df)); % Señal derivada en tiempo conversion FM-AM

fm1\_dt\_abs = abs(fm1\_dt);

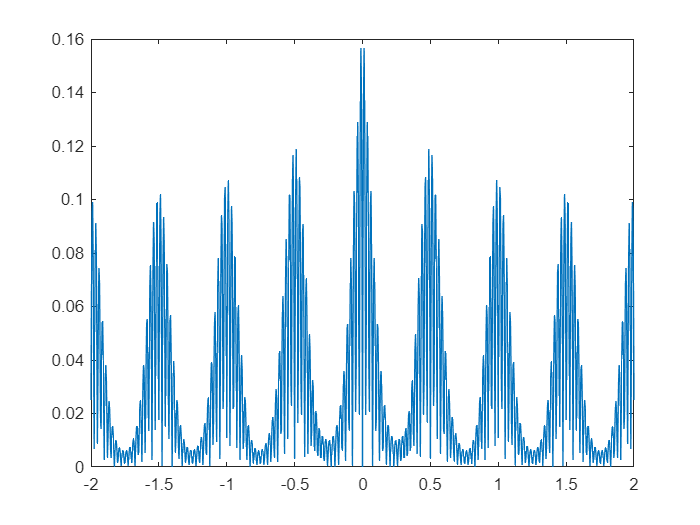
figure(2);

plot(time,fm1\_dt); % señal derivada



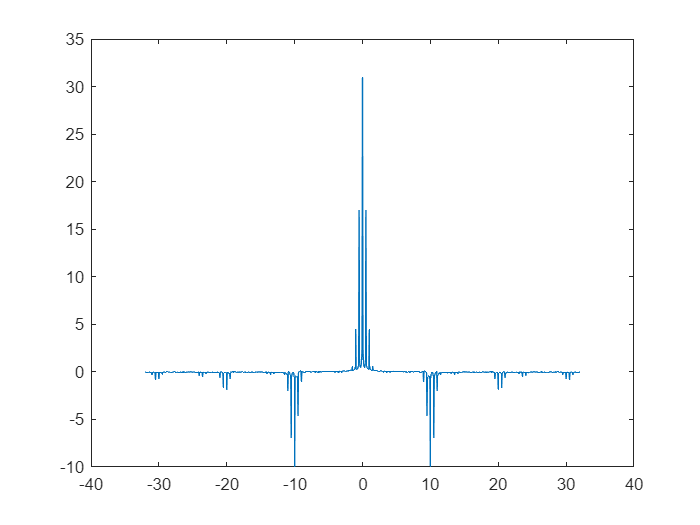
figure(3);

plot(time,fm1\_dt\_abs);



fm1\_dt\_abs\_f = real(ttof(fm1\_dt\_abs));

plot(freq,fm1\_dt\_abs\_f);



lowpassFilter = zeros(1,length(freq));

%make 1 the range of values that wanna be filtered

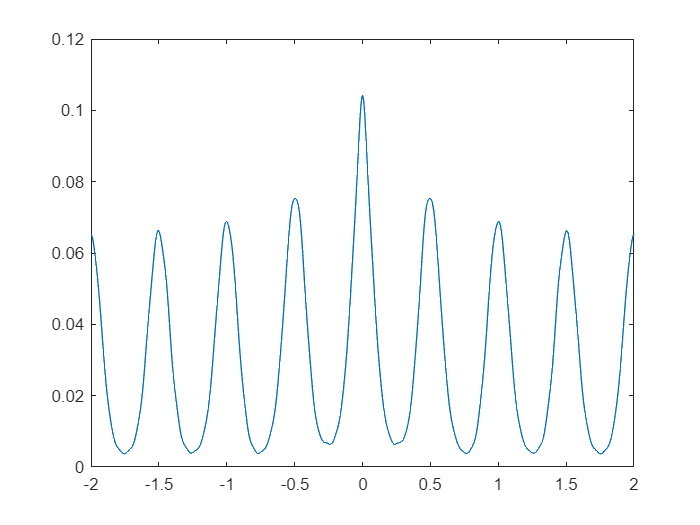
lowpassFilter(find(freq==interp1(freq,freq,-4,'nearest')):find(freq==interp1(freq,freq,4,'nearest'))) = 1;

Demodule\_f = lowpassFilter.\*fm1\_dt\_abs\_f; % applying filter

Demodule\_t = real(ftot(Demodule\_f)); % convert to time

figure(5);

plot(time,Demodule\_t);



figure(6);

Demode\_FM = Demodule\_t-mean(Demodule\_t); % remove DC constant

plot(time,Demode\_FM);

