

Funciones y su significado

<code>title</code>	Añadir título a ejes o leyenda
<code>xlabel</code>	Etiquetar eje x
<code>ylabel</code>	Etiquetar el eje y
<code>zlabel</code>	Etiquetar el eje z
<code>clabel</code>	Etiquetar la altura del diagrama de contorno
<code>legend</code>	Añadir leyenda a ejes
<code>colorbar</code>	Barra de color que muestra la escala de color
<code>text</code>	Añadir descripciones de texto a los puntos de datos
<code>textlabel</code>	Formatear texto con caracteres TeX
<code>gtext</code>	Añadir texto a la figura usando el mouse
<code>gtext</code>	Crear objeto de línea
<code>rectangle</code>	Crear rectángulo con esquinas agudas o curvas
<code>annotation</code>	Crear anotaciones

Subplot

La subtrama (m, n, p) divide la figura actual en una cuadrícula m -by- n y crea un eje para una subtrama en la posición especificada por p . MATLAB® numera sus subparcelas por fila, de manera que la primera subtrama es la primera columna de la primera fila, la segunda subtrama es la segunda columna de la primera fila, y así sucesivamente. Si los ejes ya existen, entonces la subtrama de comando (m, n, p) hace que la subtrama en la posición p sea el eje actual.

Linkaxes

Utilice los linkaxes para sincronizar los límites de eje individuales en diferentes subparcelas dentro de una figura. Llamar a las barras de enlace hará que todos los ejes de entrada tengan los mismos límites. Esto es útil cuando se desea ampliar o desplazar en una subtrama y mostrar el mismo rango de datos en otra subtrama.

Linkaxes (`axes_handles`) vincula los límites de los ejes x e y de los ejes especificados en el vector `axes_handles`.

Linkaxes (axes_handles, 'option') Enlaza los ejes axes_handles según la opción especificada. El argumento de opción puede ser una de las siguientes cadenas:

Varargin

Varargin es una variable de entrada en una sentencia de definición de función que permite a la función aceptar cualquier número de argumentos de entrada. Especifique varargin usando caracteres en minúsculas e incluya como el último argumento de entrada después de cualquier entrada declarada explícitamente. Cuando la función se ejecuta, varargin es una matriz de celdas 1 por N, donde N es el número de entradas que recibe la función después de las entradas declaradas explícitamente. Sin embargo, si la función no recibe entradas después de las entradas declaradas explícitamente, varargin es una matriz de celdas vacías.

LineWidth

establece el ancho de los objetos de línea, como gráficos de funciones 2D, curvas en 2D y 3D, flechas, líneas de parámetros en superficies, etc.

El valor debe especificarse como una longitud física absoluta incluyendo una unidad de longitud tal como LineWidth = 1.5 * unit :: mm. Los números sin una unidad física dan el tamaño en mm.

Tenga en cuenta que los gráficos no siempre pueden reaccionar a pequeños cambios del ancho de línea debido a la discretización en píxeles.

No se pueden hacer líneas invisibles estableciendo su ancho en 0. Utilice LinesVisible = FALSE en su lugar.

LineWidth no tiene un efecto en el ancho de línea de los ejes y las líneas de cuadrícula de coordenadas. Utilice los atributos AxesLineWidth y GridLineWidth para manipular los ejes y la cuadrícula de coordenadas, respectivamente.

Find

Find (X) devuelve un vector que contiene los índices lineales de cada elemento no nulo en el array X.

Si X es un vector, entonces find devuelve un vector con la misma orientación que X.

Si X es una matriz multidimensional, entonces find devuelve un vector de columna de los índices lineales del resultado.

Si X no contiene elementos distintos de cero o está vacío, entonces find devuelve una matriz vacía.

Fplot

Fplot (f) traza la curva definida por la función $y = f(x)$ sobre el intervalo por defecto $[-5\ 5]$ para x.

ejemplo

Fplot (f, xinterval) en el intervalo especificado. Especifique el intervalo como un vector de dos elementos de la forma $[xmin\ xmax]$.

ejemplo

Fplot (funx, funy) traza la curva definida por $x = funx(t)$ y $y = funy(t)$ sobre el intervalo por defecto $[-5\ 5]$ para t.

Fplot (funx, funy, tinterval) durante el intervalo especificado. Especifique el intervalo como un vector de dos elementos de la forma $[tmin\ tmax]$.

Fplot (____, LineSpec) especifica el estilo de línea, el símbolo de marcador y el color de línea. Por ejemplo, '-r' traza una línea roja. Utilice esta opción después de cualquiera de las combinaciones de argumentos de entrada en las sintaxis anteriores.

ejemplo

Fplot (____, Name, Value) especifica propiedades de línea usando uno o más argumentos de par de nombre-valor. Por ejemplo, 'LineWidth', 2 especifica un ancho de línea de 2 puntos.

Fplot (ax, ____) traza en los ejes especificados byax en lugar de los ejes actuales (gca). Especifique los ejes como el primer argumento de entrada.

ejemplo

Fp = fplot (____) devuelve un objeto de línea de función o un objeto de línea de función parametrizada, dependiendo de las entradas. Utilice fp para consultar y modificar las propiedades de una línea específica. Para obtener una lista de propiedades, consulte Propiedades de la línea de función o Propiedades de la línea de función parametrizada.

[X, y] = fplot (____) devuelve las abscisas y las ordenadas de la función sin crear una trama. Esta sintaxis se eliminará en una versión futura. Utilice las propiedades XData y YData del objeto de línea de función, fp, en su lugar.

Nota: fplot ya no soporta argumentos de entrada para especificar la tolerancia de error o el número de puntos de evaluación. Para especificar el número de puntos de evaluación, utilice la propiedad MeshDensity.

Amdemod

Amdemod (y, Fc, Fs) demodula la señal modulada en amplitud y de una señal portadora con frecuencia Fc (Hz). La señal portadora y y tienen frecuencia de muestreo Fs (Hz). La señal modulada y tiene cero fase inicial y cero portadora amplitud, por lo que representa suprimido portador modulación. El proceso de demodulación utiliza el filtro de paso bajo especificado por [num, den] = butter (5, Fc * 2 / Fs).

Butter

devuelve los coeficientes de la función de transferencia de un filtro digital Butterworth de paso inferior de orden n con una frecuencia de corte normalizada Wn.

Square

Genera una onda cuadrada con el periodo 2π para los elementos del vector temporal t. Cuadrado (t) es similar a sin (t), pero crea una onda cuadrada con picos de ± 1 en lugar de una onda sinusoidal.

stem

La función stem traza datos de secuencia discreta

Traza la secuencia de datos, Y, a los valores especificados por X. Las entradas X e Y deben ser vectores

o matrices del mismo tamaño. Además, X puede ser un vector de fila o columna e Y debe ser una matriz con

filas de longitud (X).

Si X e Y son ambos vectores, entonces el tallo traza entradas en Y contra entradas correspondientes en X.

Si X es un vector e Y es una matriz, entonces el tallo traza cada columna de Y contra el conjunto de valores especificados por X, de modo que todos los elementos de una fila de Y se trazan contra el mismo valor.

Si X e Y son ambas matrices, entonces el tallo traza columnas de Y contra columnas correspondientes de X.

abs(func)

Valor absoluto de los elementos de “func” o módulo en el caso de ser complejos.

A continuación el código en matlab

Funcion furier

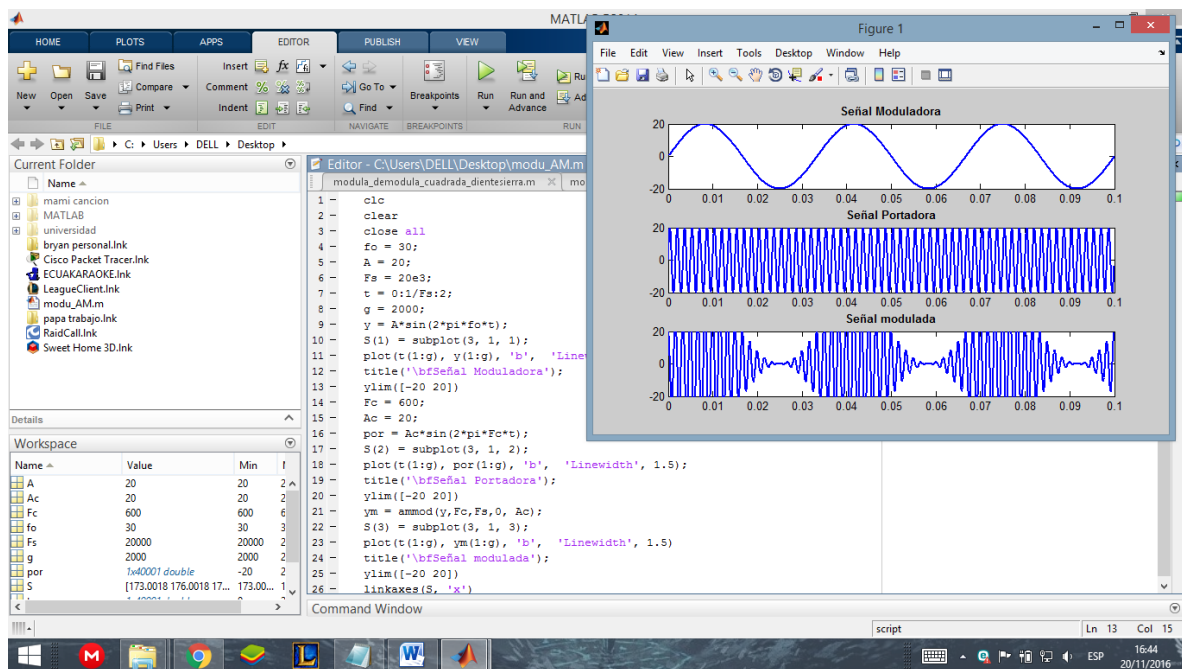
```
function varargout = fou_rier(varargin)
% El comando fou_rier calcula el vector de
% frecuencias y el vector del modulo
% de la transformada de fourier, para graficarlo
% de forma directa
% Parámetros:
% [f esp] = fou_rier(y, Fs) o bien
% [f esp] = fou_rier(y, t)
% donde y es el vector de informacion
% Fs la frecuencia de muestreo y t
% el vector de tiempo.

if nargin == 2 && nargsout == 2
    y = varargin{1};
    if length(varargin{2}) == 1
        Fs = varargin{2};
    else
        t = varargin{2};
        Fs = 1/(t(2)-t(1));
    end
    L = length(y);
    NFFT = 2^nextpow2(L); % Próxima potencia de 2 desde la longitud de y
    Y = fft(y,NFFT)/L;
    f = Fs/2*linspace(0,1,NFFT/2+1);
    modulo = 2*abs(Y(1:NFFT/2+1));

    varargout{1} = f;
    varargout{2} = modulo;
else
    error('Los argumentos de entrada son el vector de informacion junto
    con la Fs o el vector de tiempo. Los de salida son el vector de
    frecuencias y de informacion');;
end
```

Modulacion AM

```
clc
clear
close all
fo = 30;
A = 20;
Fs = 20e3;
t = 0:1/Fs:2;
g = 2000;
y = A*sin(2*pi*fo*t);
S(1) = subplot(3, 1, 1);
plot(t(1:g), y(1:g), 'b', 'Linewidth', 1.5);
title('\bfSeñal Moduladora');
ylim([-20 20])
Fc = 600;
Ac = 20;
por = Ac*sin(2*pi*Fc*t);
S(2) = subplot(3, 1, 2);
plot(t(1:g), por(1:g), 'b', 'Linewidth', 1.5);
title('\bfSeñal Portadora');
ylim([-20 20])
ym = ammod(y,Fc,Fs,0, Ac);
S(3) = subplot(3, 1, 3);
plot(t(1:g), ym(1:g), 'b', 'Linewidth', 1.5)
title('\bfSeñal modulada');
ylim([-20 20])
linkaxes(S, 'x')
```



Modulación, demmodulacion, seno

```
clc
clear
close all

% Características de la señal moduladora (datos)
fo = 100;
A = 10;

Fs = 20e3;
t = 0:1/Fs:2;
g = 2000;
% Generación de la señal moduladora
y = A*sin(2*pi*fo*t);
S(1) = subplot(3, 1, 1);
plot(t(1:g), y(1:g), 'b', 'Linewidth', 1.5);
title('\bfSeñal Moduladora');
ylim([-20 20])

% Generación de la señal portadora
Fc = 300;
Ac = 10;
por = Ac*sin(2*pi*Fc*t);
S(2) = subplot(3, 1, 2);
plot(t(1:g), por(1:g), 'b', 'Linewidth', 1.5);
title('\bfSeñal Portadora');
ylim([-20 20])

% Modulación de la señal
ym = ammod(y, Fc, Fs, 0, Ac);
S(3) = subplot(3, 1, 3);
plot(t(1:g), ym(1:g), 'b', 'Linewidth', 1.5);
title('\bfSeñal modulada');
ylim([-20 20])

linkaxes(S, 'x');

% Espectro de frecuencia de la señal moduladora
pause; figure
T(1) = subplot(2, 2, 1);
[f, esp] = fourier(y, Fs);
plot(f, esp, 'r', 'Linewidth', 2);
title('\bfESPECTRO DE FRECUENCIA DE LA SEÑAL ORIGINAL EN BANDA BASE')
d = find(esp > max(esp) * 0.1);
xlim([0 f(d*2)])

T(2) = subplot(2, 2, 2);
[f, esp] = fourier(por, t);
```

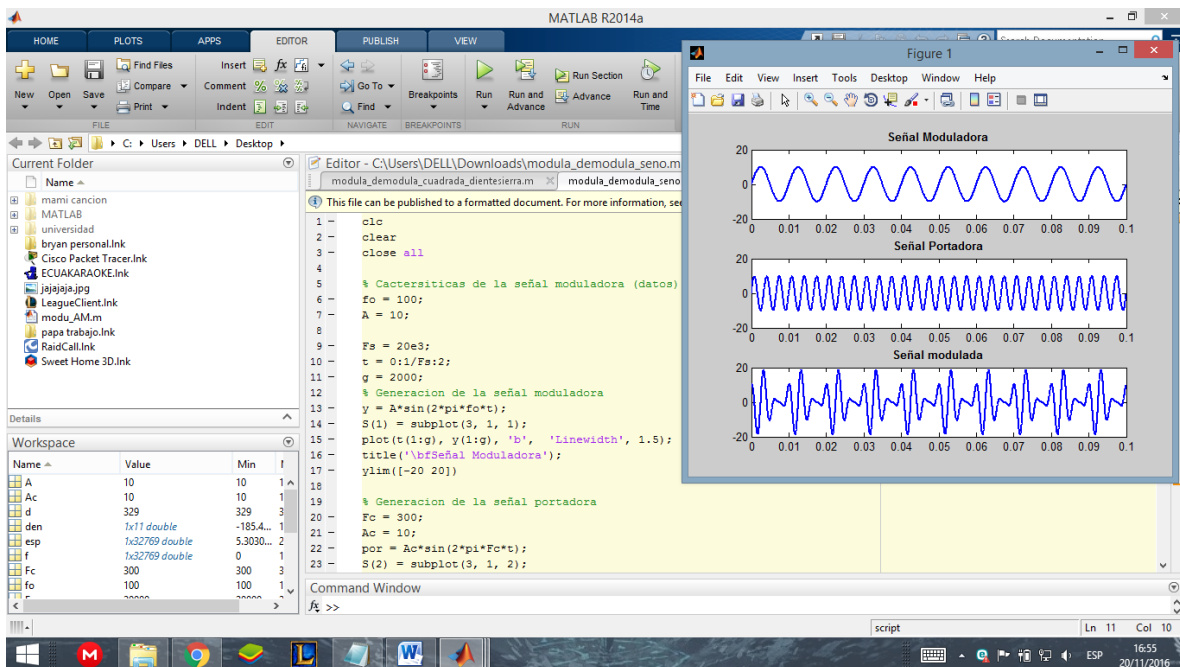
```

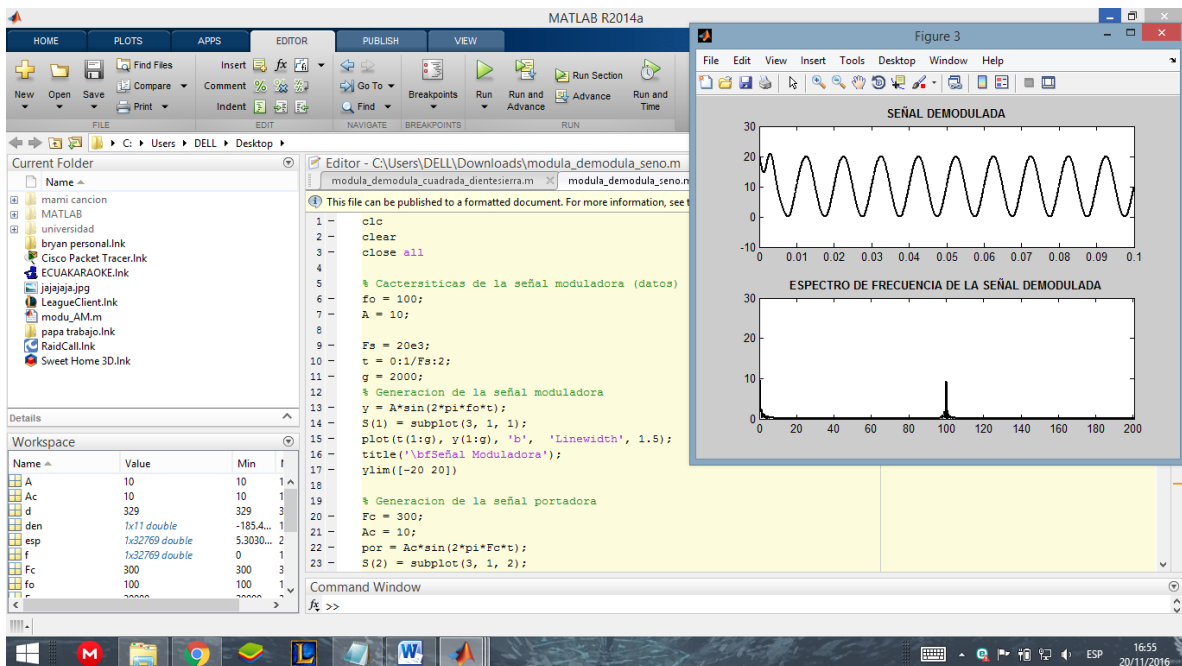
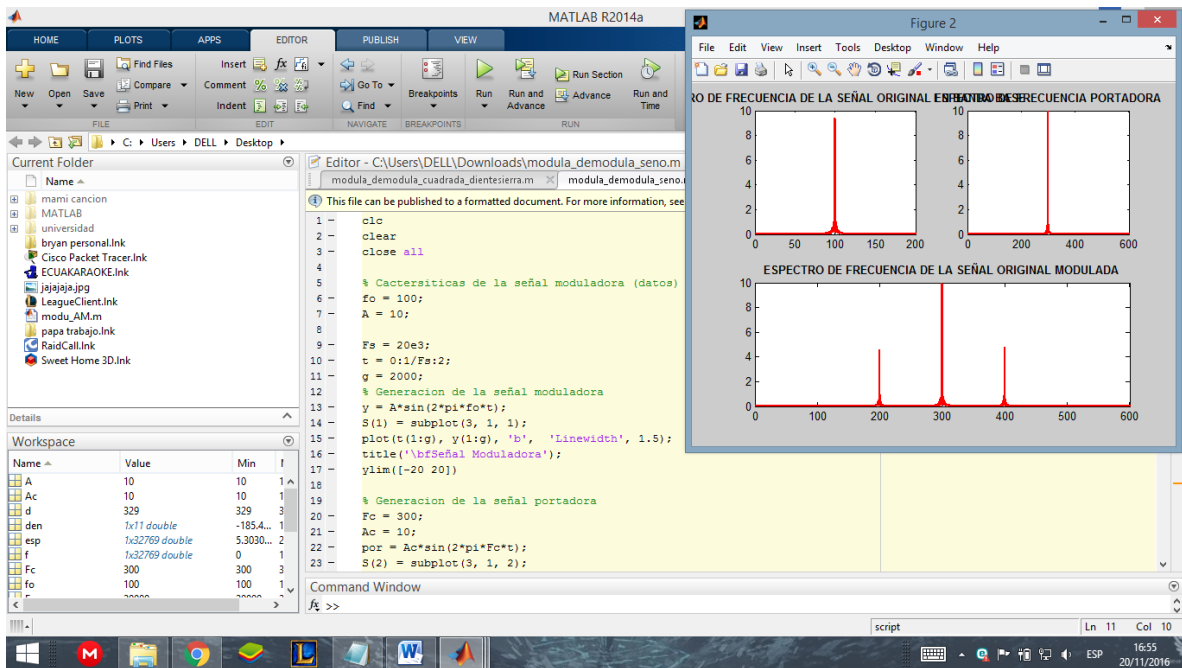
plot(f, esp, 'r', 'Linewidth', 2);
title('\bfESPECTRO DE FRECUENCIA PORTADORA')
d = find(esp>max(esp)-max(esp)*0.1);
xlim([0 f(d*2)])

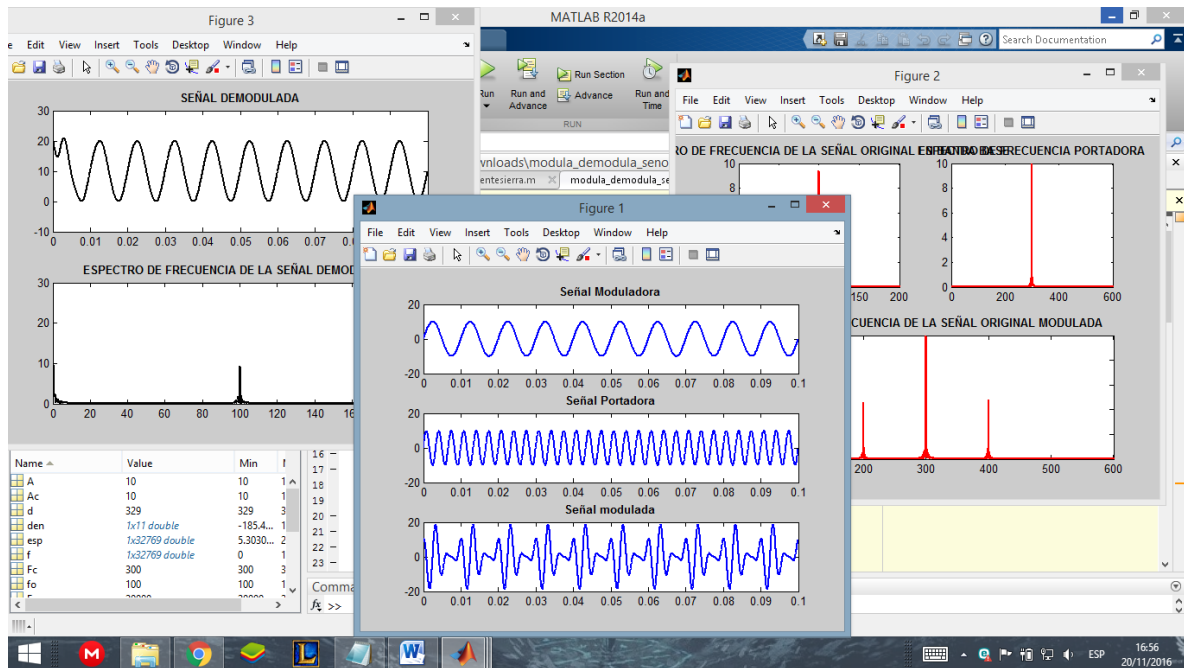
% Espectro de frecuencia de la señal modulada
T(3) =subplot(2, 1, 2);
[f, esp] = fou_rier(ym, Fs);
plot(f, esp, 'r', 'Linewidth', 2);
title('\bfESPECTRO DE FRECUENCIA DE LA SEÑAL ORIGINAL MODULADA')
d = find(esp> (max(esp)-max(esp)*0.1));
xlim([0 f(d*2)])

pause; figure
subplot(2, 1, 1);
[num,den] = butter(10,Fc*2/Fs);
ydem = amdemod(ym,Fc,Fs,0,0,num,den);
plot(t(1:g),ydem(1:g), 'k', 'Linewidth', 2);
title('\bfSEÑAL DEMODULADA')
subplot(2, 1, 2);
[f esp] = fou_rier(ydem, t);
plot(f, esp, 'k', 'Linewidth', 2);
title('\bfESPECTRO DE FRECUENCIA DE LA SEÑAL DEMODULADA')
d = find(esp> (max(esp)-max(esp)*0.8));
d = d(end);
xlim([0 f(d*2)])

```







Modulada, demodulada y cuadrada

```
clc
clear
close all

% Características de la señal moduladora (datos)
fo = 100;
A = 10;

Fs = 20e3;
t = 0:1/Fs:2;
g = 2000;
% Generación de la señal moduladora
y = A*square(2*pi*fo*t);
S(1) = subplot(3, 1, 1);
plot(t(1:g), y(1:g), 'b', 'Linewidth', 1.5);
title('\bfSeñal Moduladora');
ylim([-20 20])

% Generación de la señal portadora
Fc = 1500;
```

```

Ac = 10;
por = Ac*sin(2*pi*Fc*t);
S(2) = subplot(3, 1, 2);
plot(t(1:g), por(1:g), 'b', 'Linewidth', 1.5);
title('\bfSeñal Portadora');
ylim([-20 20])

% Modulacion de la señal
ym = ammod(y,Fc,Fs,0, Ac);
S(3) = subplot(3, 1, 3);
plot(t(1:g), ym(1:g), 'b', 'Linewidth', 1.5)
title('\bfSeñal modulada');
ylim([-20 20])

linkaxes(S, 'x');

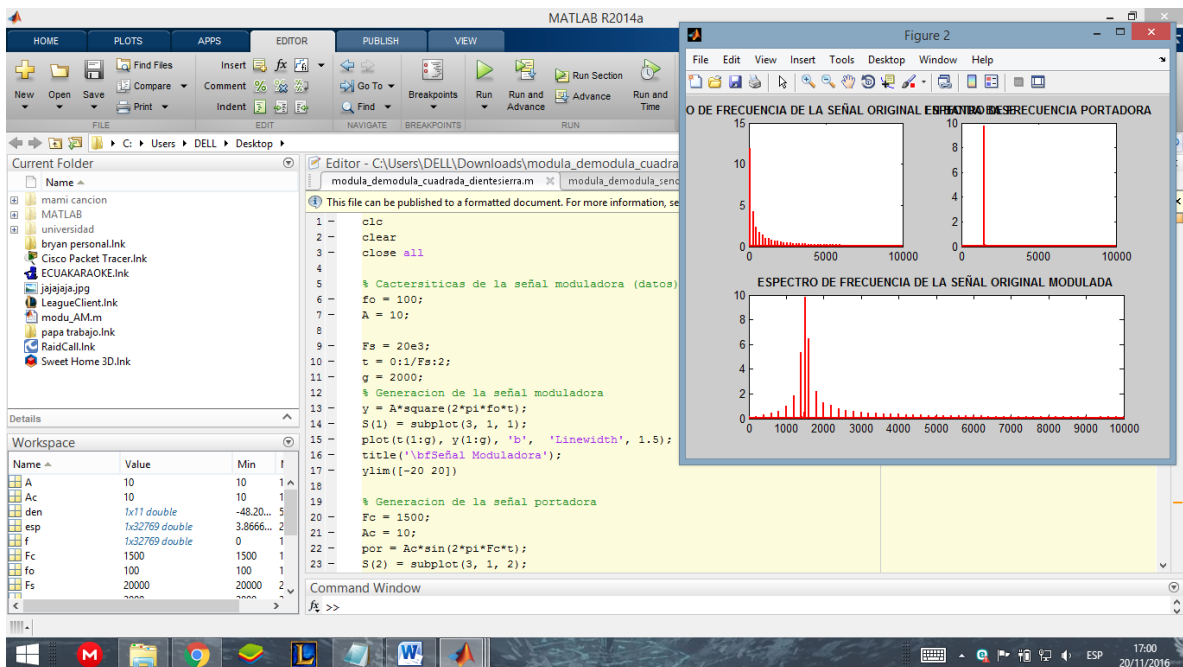
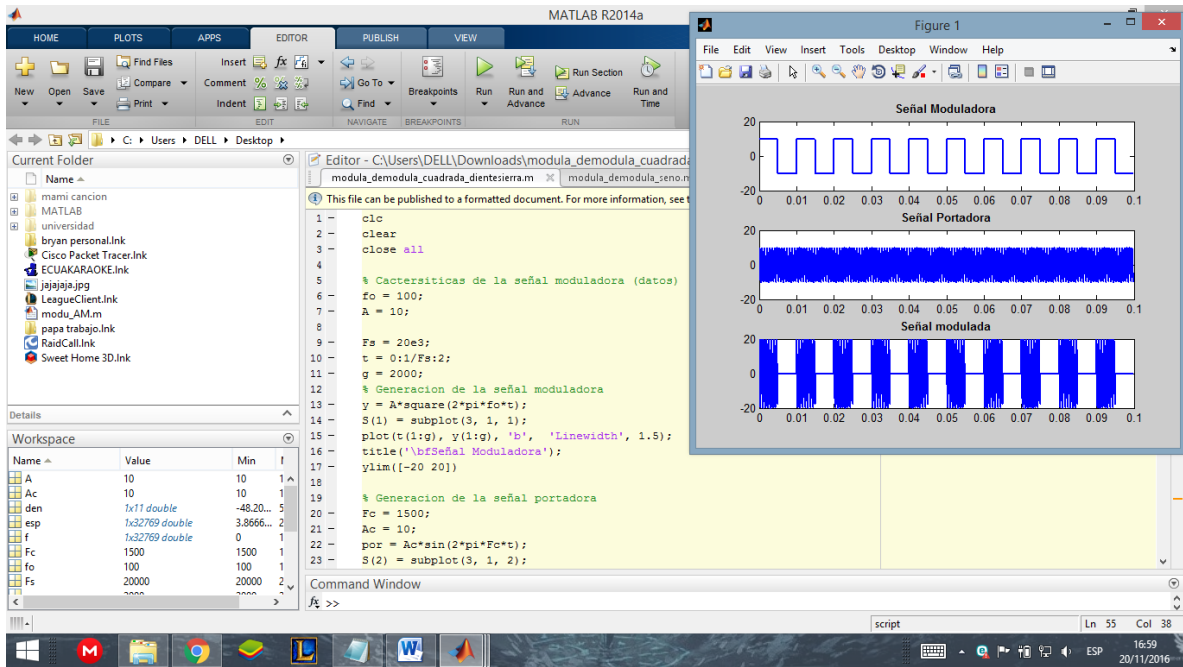
% Espectro de frecuencia de la señal moduladora
pause; figure
T(1) =subplot(2, 2, 1);
[f, esp] = fou_rier(y, Fs);
plot(f, esp, 'r', 'Linewidth', 2);
title('\bfESPECTRO DE FRECUENCIA DE LA SEÑAL ORIGINAL EN BANDA BASE')

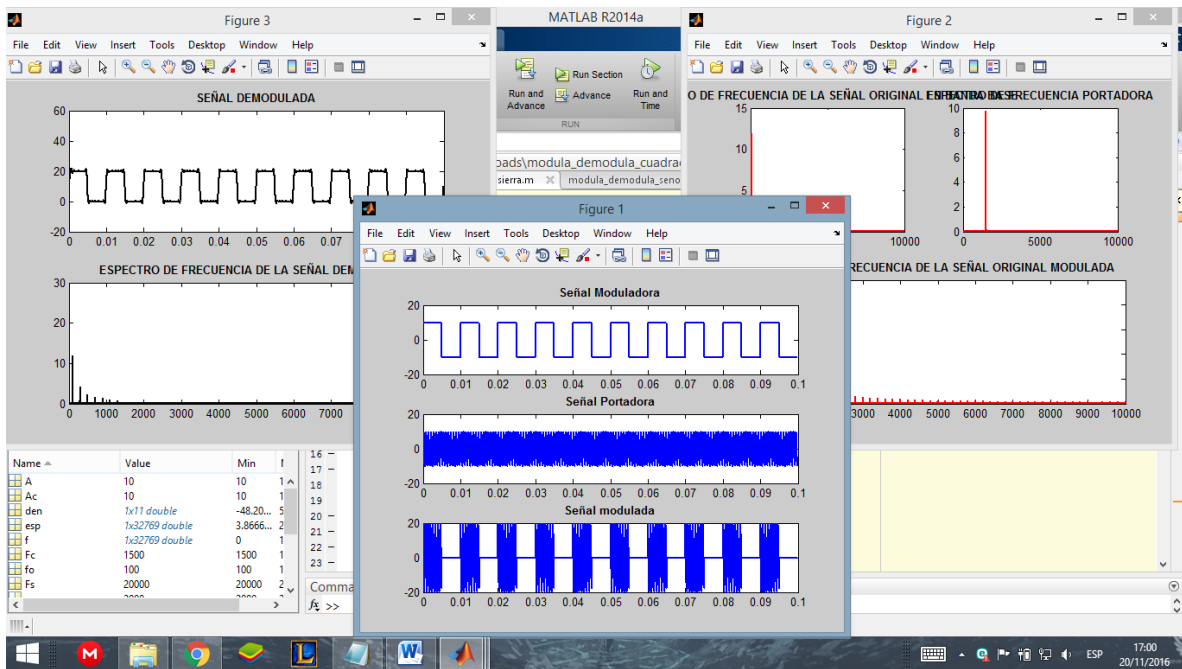
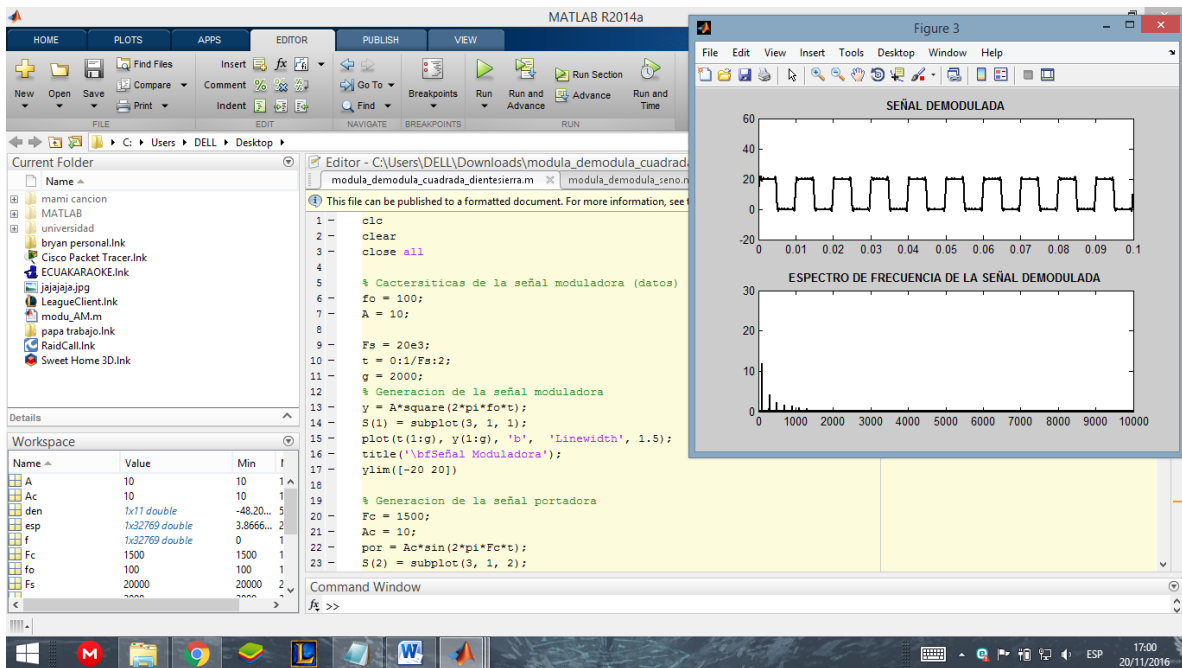
T(2) = subplot(2, 2, 2);
[f, esp] = fou_rier(por, Fs);
plot(f, esp, 'r', 'Linewidth', 2);
title('\bfESPECTRO DE FRECUENCIA PORTADORA')

% Espectro de frecuencia de la señal modulada
T(3) =subplot(2, 1, 2);
[f, esp] = fou_rier(ym, Fs);
plot(f, esp, 'r', 'Linewidth', 2);
title('\bfESPECTRO DE FRECUENCIA DE LA SEÑAL ORIGINAL MODULADA')

pause; figure
subplot(2, 1, 1);
[num,den] = butter(10,Fc*2/Fs);
ydem = amdemod(ym,Fc,Fs,0,0,num,den);
plot(t(1:g),ydem(1:g), 'k', 'Linewidth', 2);
title('\bfSEÑAL DEMODULADA')
subplot(2, 1, 2);
[f esp] = fou_rier(ydem, t);
plot(f, esp, 'k', 'Linewidth', 2);
title('\bfESPECTRO DE FRECUENCIA DE LA SEÑAL DEMODULADA')

```





BIBLIOGRAFIA:

<https://www.youtube.com/watch?v=0bZHVmwET1M&t=92s>

<https://www.youtube.com/watch?v=ivrDFJbUZRM&t=130s>

http://artico.lma.fi.upm.es/numerico/asigs/tds/practicas/pr_mod_2010.pdf

<https://es.scribd.com/doc/26448464/Modulacion-AM-en-Matlab>

<https://es.scribd.com/doc/116525573/76605854-Modulacion-en-Am-y-Fm-en-Matlab-Telecomunicaciones>

<https://www.youtube.com/watch?v=0bZHvmwET1M&t=92s>

<https://www.youtube.com/watch?v=CYmjxdIU-s>

<https://www.mathworks.com/help/matlab/ref/find.html>

<https://www.mathworks.com/help/signal/ref/butter.html>

<https://www.mathworks.com/help/signal/ref/square.html>

<https://www.mathworks.com/help/comm/ref/amdemod.html>