**ClassWord 1**

Gestión de redes y comunicaciones

Facultad: Ingeniería de Sistemas

Presentado por:

Orlando Camilo Acosta Vargas

Profesor: Alexander García Pérez, Ingeniería de Sistemas

Universidad EAN

2019

**Índice**

1. **Introducción**
2. **Señal sinusoidal**

**2.1. El período (T) en una sinusoide**.

**2.2. Modelo practico en Matlab**

**2.2.1 ¿Qué es MATLAB?**

**2.2.2 Pasos de desarrollo en Matlab**

**2.2.3 Código de Señal sinusoidal en Matlab**

* + 1. **Pasos de desarrollo interfaz gráfica de Matlab**

1. **Serie Fourier**
   1. **Serie Fourier: generador de señal**
   2. **Código de la señal Fourier en Matlab**
   3. **Código de la señal Fourier generando funciones Sinusoidal, Cuadrada, Triangular, Diente Sierra en Matlab**
2. **Transformada de Fourier.**

**4.1. Transformada de Fourier: analizador de espectro.**

**4.2. Código de la señal Transformada de Fourier en Matlab**

1. **Modulación de simulación: analógica + digital**

**5.1. Modulación**

**5.1.1. Tipos de Modulación**

**5.2. Modulación Análoga**

**5.2.1. Modulación Am**

**5.2.2. Modulación Fm**

**5.2.3. Modulación Pm**

**5.3. Modulación Digital**

1. **Bibliografía** 
   1. **Introducción**
   2. **Señal sinusoidal**

 La **Señal de onda sinusoidal representan el valor de la tención**

**de corriente alterna a través de un tiempo** continuamente variable. Está dada por la función.

Donde:

**2.1. El período (T) en una sinusoide**.

Podemos definirlo como conjunto de valores de t que corresponden a un ciclo completo de valores de la función; en este sentido toda función de una variable que repite sus valores en un ciclo completo es una función periódica.

La frecuencia se mide en Hertz (Hz). Muestra el número de ciclos de señal que tuvieron lugar en un segundo.

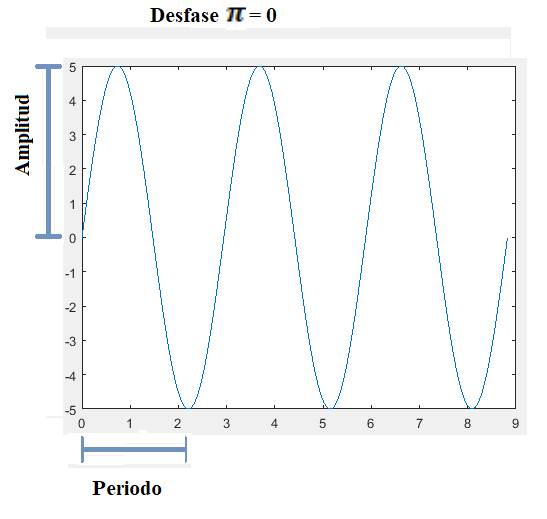
****

Figura (1) Ejemplo de señal sinusoidal

**2.2. Modelo practico en Matlab**

**2.2.1 ¿Qué es MATLAB?**

MATLAB es un lenguaje de alto desempeño diseñado para realizar cálculos técnicos.

Es un ambiente de programación fácil de utilizar donde los problemas y las soluciones se expresan en una notación matemática. MATLAB es un sistema interactivo cuyo elemento básico de datos es el arreglo que no requiere de dimensionamiento previo. Esto permite resolver muchos problemas computacionales, específicamente aquellos que involucren vectores y matrices, en un tiempo mucho menor al requerido para escribir un programa en un lenguaje escalar no interactivo tal como C o Fortran.

MATLAB se utiliza ampliamente en:

• Cálculos numéricos

• Desarrollo de algoritmos

• Modelado, simulación y prueba de prototipos

• Análisis de datos, exploración y visualización

• Graficación de datos con fines científicos o de ingeniería

• Desarrollo de aplicaciones que requieran de una interfaz gráfica de usuario (GUI, Graphical User Interface).

**(**M. C. José Jaime Esqueda Elizondo, Matlab e Interfaces Gráficas, noviembre, 2002**)**

**2.2.2 Pasos de desarrollo en Matlab**

* Inicializamos el entorno de Matlab
* Creamos un nueva Carpeta para el proyecto
* Creamos un nuevo proyecto o un nuevo Script
* Le damos un nombre al archivo y lo guardamos como un archivo .m
* Compilamos el archivo

**2.2.3 Código de Señal sinusoidal en Matlab**

%limpiamos la consola

clc

clear all

close all

%---- DATA IN --

A = 5; %% Amplitud

f = 3400; %% frecuencia

phi = 0; %% desfase

cantper = 3; %%periodo

%--- PROCESS ---

t = linspace(0,cantper\*1/f,1000);%--Aquí se maneja como "T" el Periodo de nuestra señal

xt= A\*sin(2\*pi\*f\*t + phi); %--ecuación de la señal

%--- OUTPUT ---

%-- Graficacion --

figure(1)

plot(t, xt)

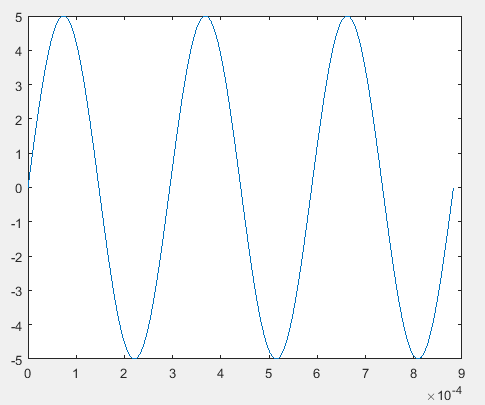


Figura (2) compilación de señal sinusoidal

**2.2.4 Pasos de desarrollo interfaz grafica de Matlab**

* Inicializamos el entorno de Matlab
* Creamos un nueva Carpeta para el proyecto
* En la línea de comandos de Matlab escribimos guide y se nos abrirá la consola de GUI como muestra en la figura 3.

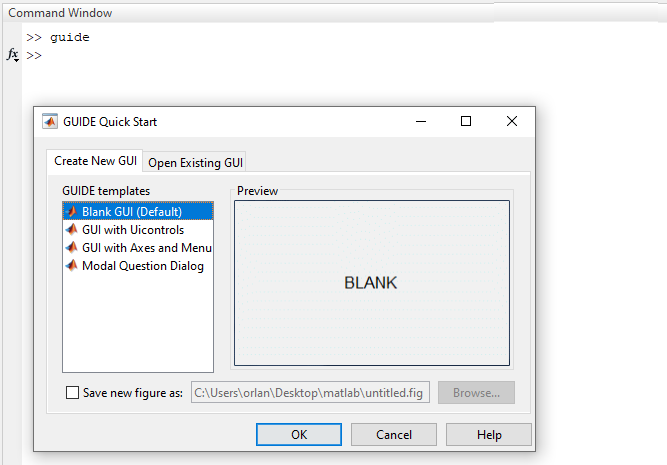


Figura (3) Inicio de interfaz GUI

* Le damos click en ok, y se nos abrirá el área de trabajo GUI como muestra en la figura (4)

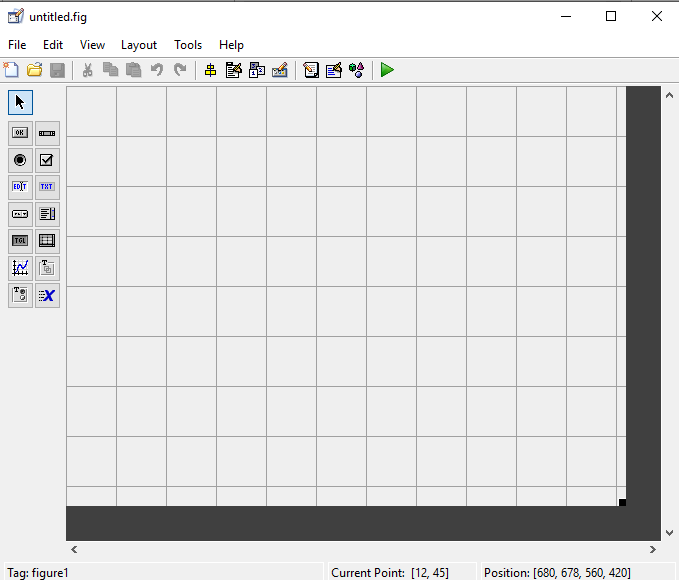


Figura (3) Entorno de trabajo GUI

* Luego le damos click derecho y seleccionamos la propiedad de inspector como muestra en la figura 4

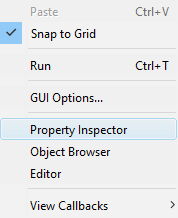


Figura (4) Inspector GUI

* De ahí se nos abrirá la barra de tareas y podremos realizar los cambios al entorno GUI.

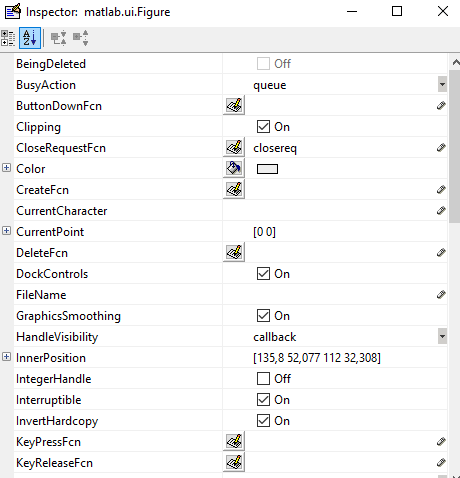


Figura (5) barra de tareas Inspector

* Si queremos le cambiamos el color de nuestra consola

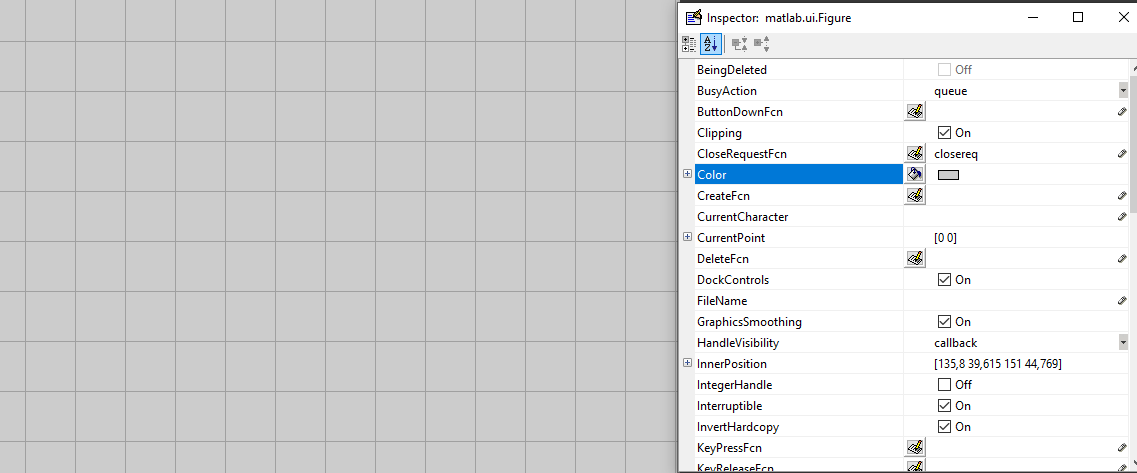


Figura (6) Cambio de color a consola

* Ahora le damos click en la barra derecha en el boton Axes y colocamos 4 de ellos en la consola

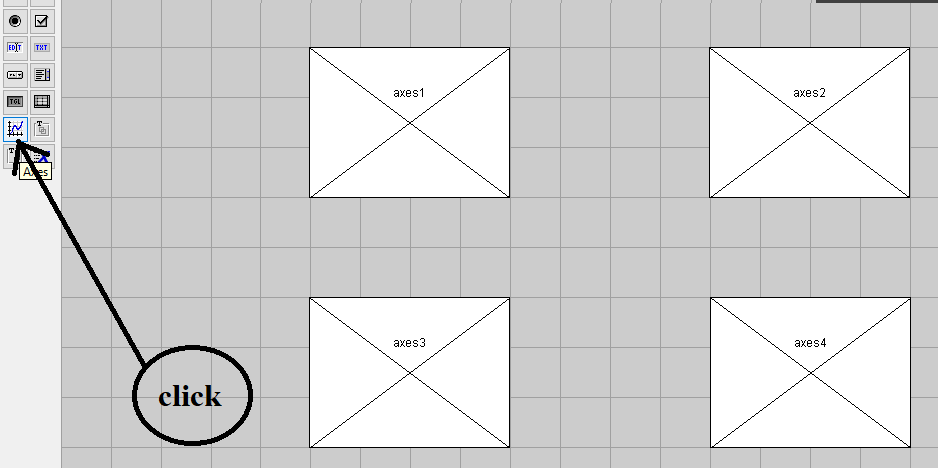


Figura (7) Axes graficador

* Ahora en la misma barra hacemos click en txt y agregamos un texto, y le cambiamos el nombre

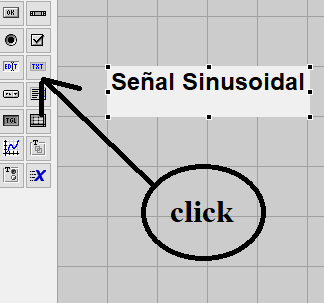


Figura (8) Crear Boton text

* Ahora en la misma barra le agregamos un push button a la consola



Figura (9) Crear Boton Push

* Ahora le damos click derecho al botón, le damos click en la propiedad inspector y le cambiamos los valores String = salir , Tag = salir

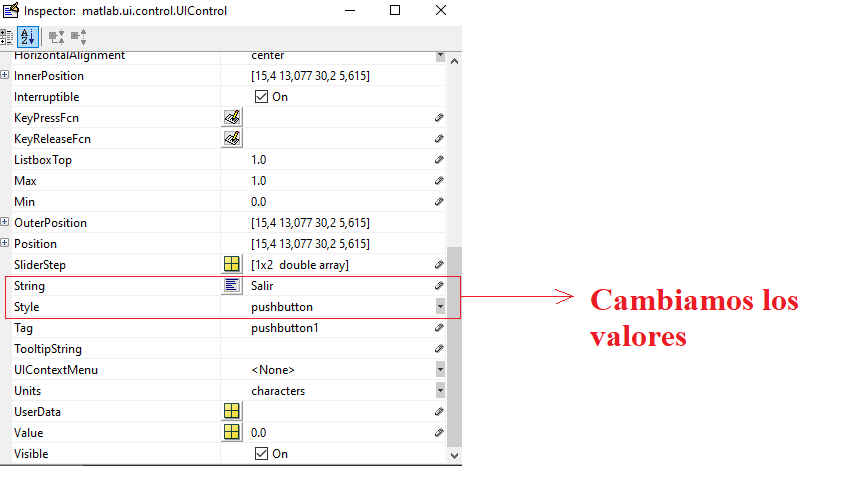


Figura (10) cambio de valores

* Realizados los cambios de damos click derecho a el botón y seleccionamos editor

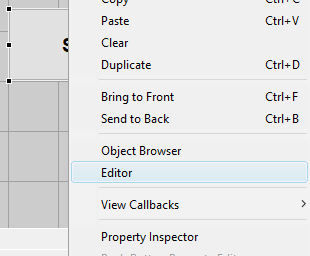


Figura (11) botón editor

* El cual nos mandara a guarguar el archivo como .fig

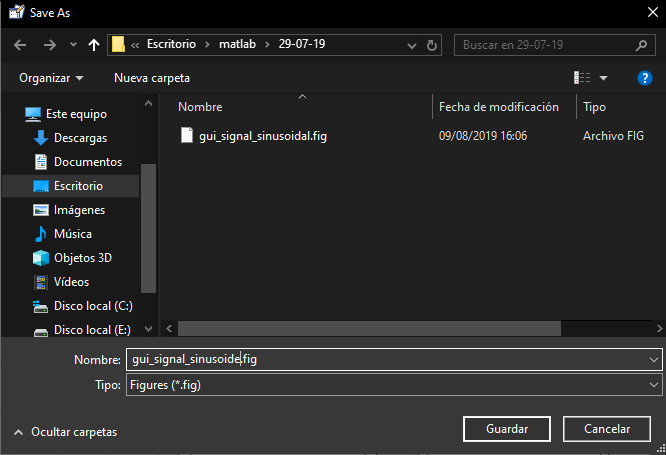


Figura (12) como debemos guardar

* Una ves generado el achivo .fig abriremos el archivo .m

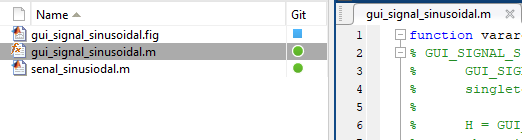
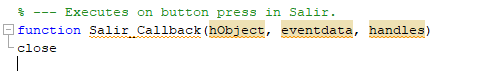


Figura (13) Formato de archivos guardados

* Buscamos el nombre de la función que le dimos a nuestro botón y le colocamos el código que necesita



* Ahora en nuestra consola GUI creamos otro botón llamado generar y realizamos el mismo procedimiento



Figura (14) creando nuevo botón

* Una vez ya hecho colocamos 3 otros formatos de textos en nuestro archivo GUI y los llamados Amplitud, frecuencia y periodo



Figura (15) creando cajas de texto

* Luego procedemos a seleccionar el botón edit text y creamos tres iconos para cada uno de ellos.



Figura (16) creando cajas de botón edit text

* Luego procedemos a cambiarle los valores a este botón texto, dándole click derecho, seleccionamos propiedad inspector y cambiamos los valores de la siguiente manera como muestra en la figura 17

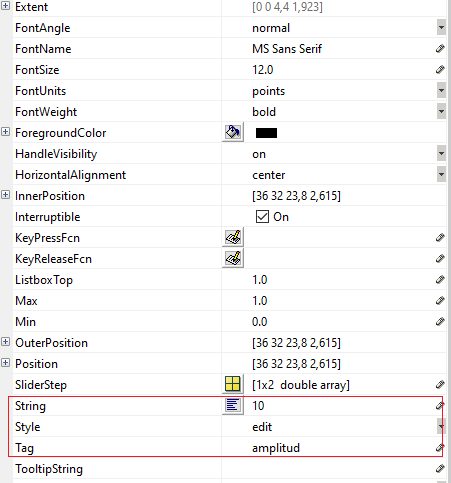


Figura (17) cambiando los valores de edit text

* Esto lo realizamos con cada uno de los botones texto
* Ahora configuramos nuestro archivo en .m en nuestra función

function generar\_Callback(hObject, eventdata, handles)

clc

%------- DATA IN -------%

A= str2num(get(handles.amplitud,'String'))

f = str2num(get(handles.frecuencia,'String'))

cantper = str2num(get(handles.periodo,'String'))

phi = pi;

phi1 = 0;

phi2 = pi/2;

phi3 = pi;

phi4 = 3\*pi/2;

%------- PROCESS -------%

t = linspace(0,cantper\*1/f,500);

xt1 = A\*sin(2\*pi\*f\*t + phi1);

xt2 = A\*sin(2\*pi\*f\*t + phi2);

xt3 = A\*sin(2\*pi\*f\*t + phi3);

xt4 = A\*sin(2\*pi\*f\*t + phi4);

%------- OUTPUT ------%

%--figure(1)--%

axes(handles.axes1)

plot(t,xt1),title('phi=0'),grid on

axes(handles.axes2)

plot(t,xt2),title('phi=pi/2'),grid on

axes(handles.axes3)

plot(t,xt3),title('phi=pi'),grid on

axes(handles.axes4)

plot(t,xt4),title('phi=3\*pi/2'),grid on

* Ahora, compilamos nuestro archivo en GUI como se muestra en la figuras 18 y 19

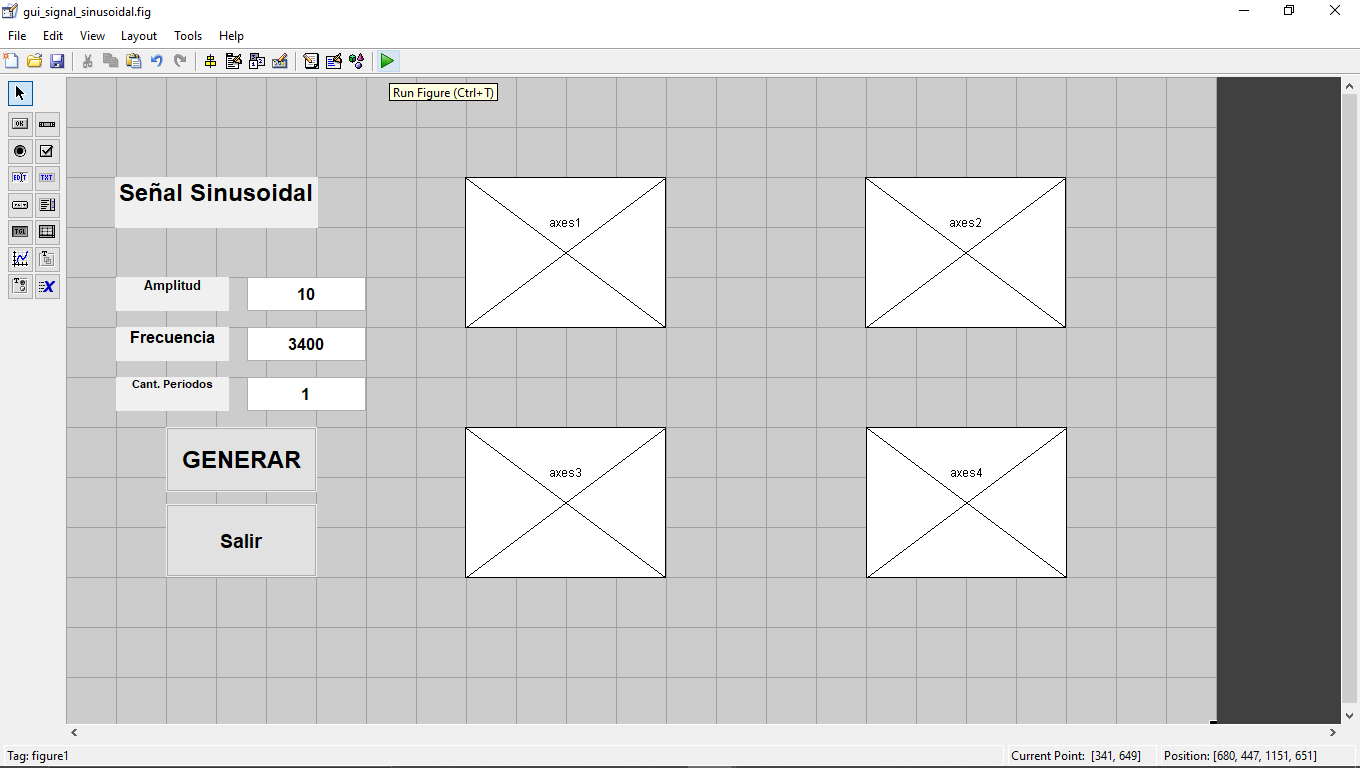


Figura (18) Click run consola GUI

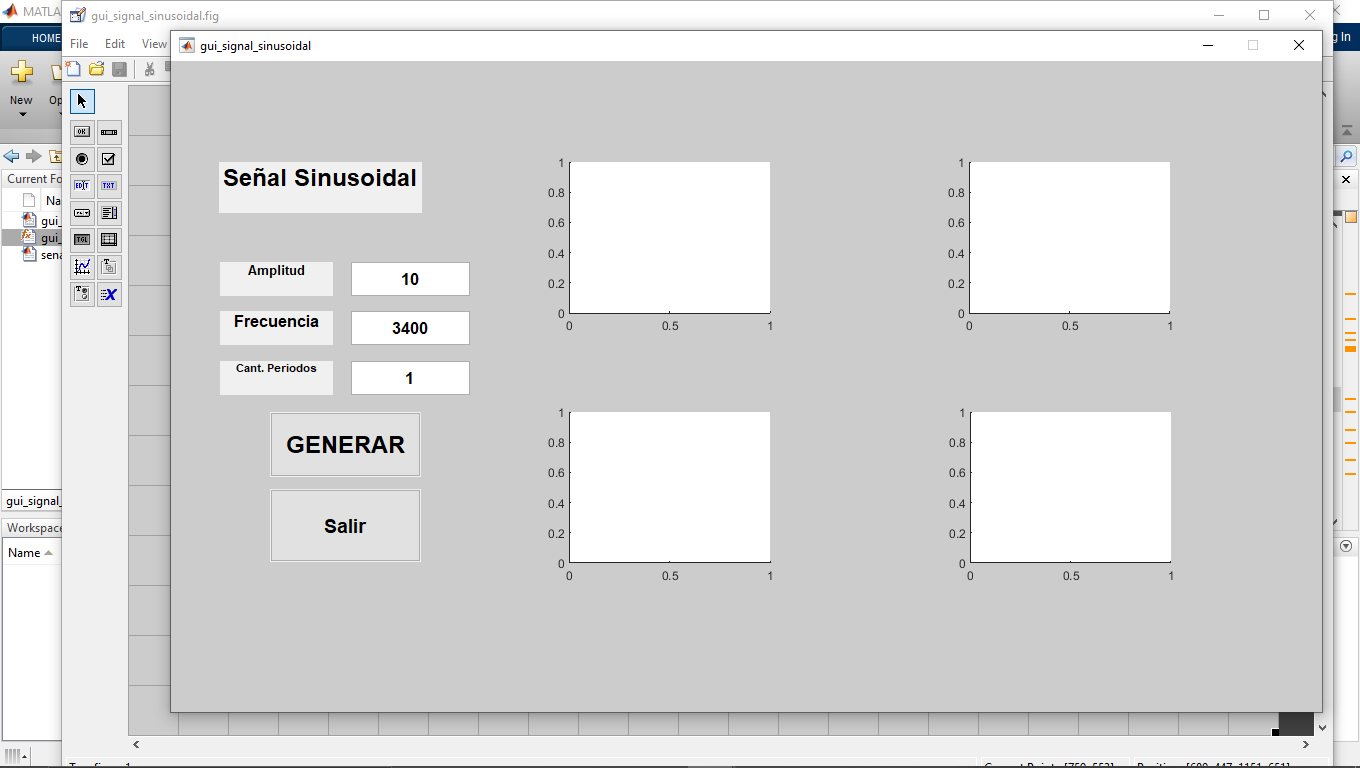


Figura (19) Interfaz gráfica de Matlab

* Una vez realizado esto le damos click en generar y obtendremos nuestra señal sinusoidal con cada una de las gráficas de desfase

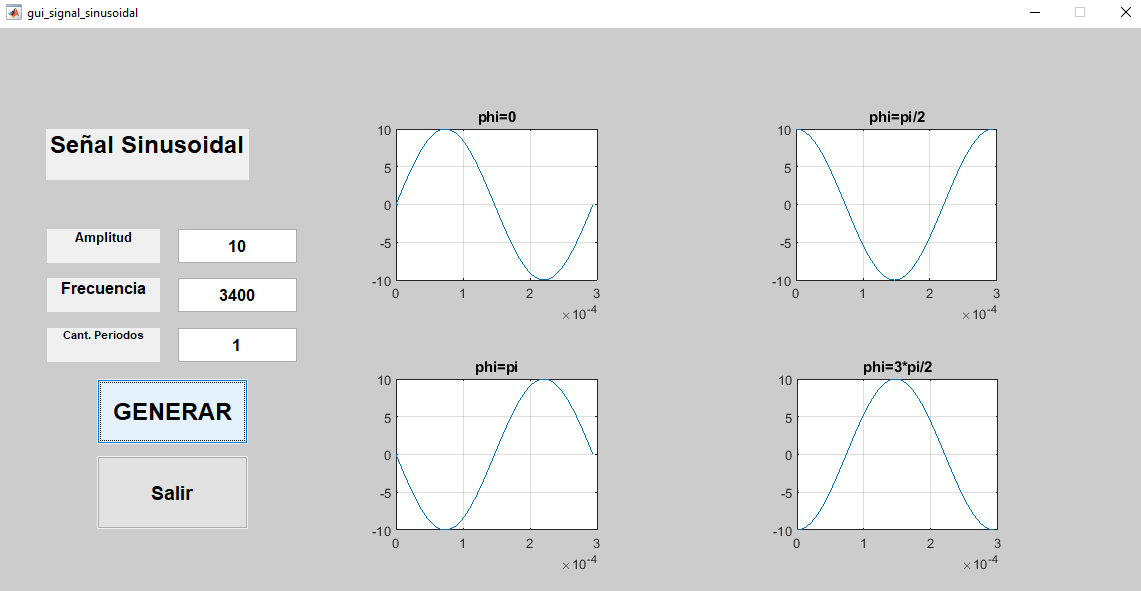


Figura (20) modelo de señal sinusoidal

* Por último, crearemos nuestro botón limpiar el cual se encargará de limpiar nuestras representaciones en axes
* Creamos nuestro botón limpiar en la interfaz GUI

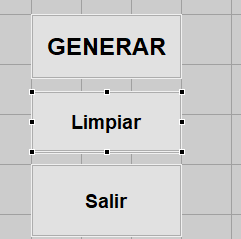


Figura (21) creación de botón limpiar

* Le configuramos los valores en propiedad inspector
* Luego procedemos a editarlo en el código

function Limpiar\_Callback(hObject, eventdata, handles)

axes(handles.axes1)

cla

axes(handles.axes2)

cla

axes(handles.axes3)

cla

axes(handles.axes4)

cla

* Por último, compilamos en nuestra interfaz GUI como muestra las figuras 18, 19, 22, 23

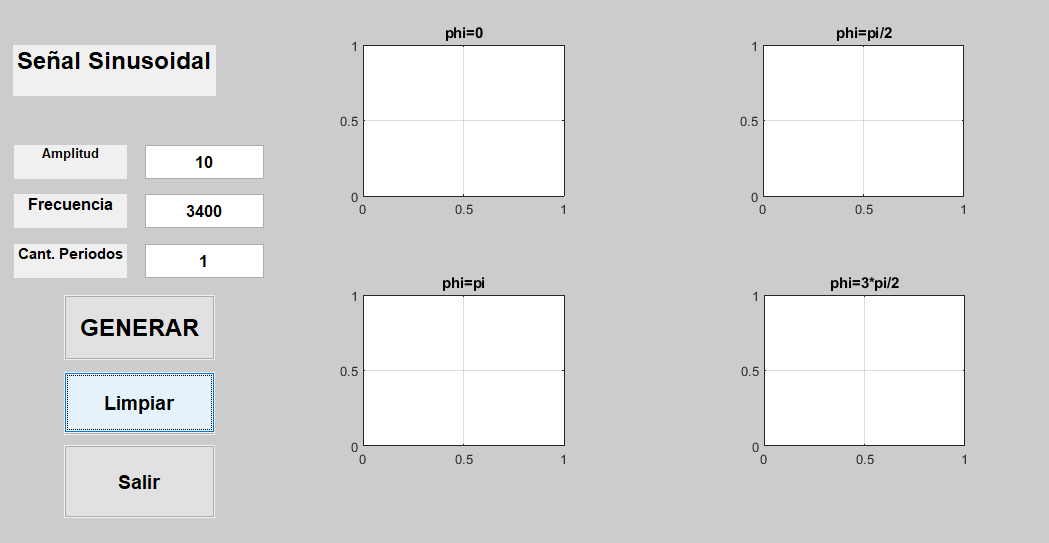


Figura (22) Aplicación botón limpiar

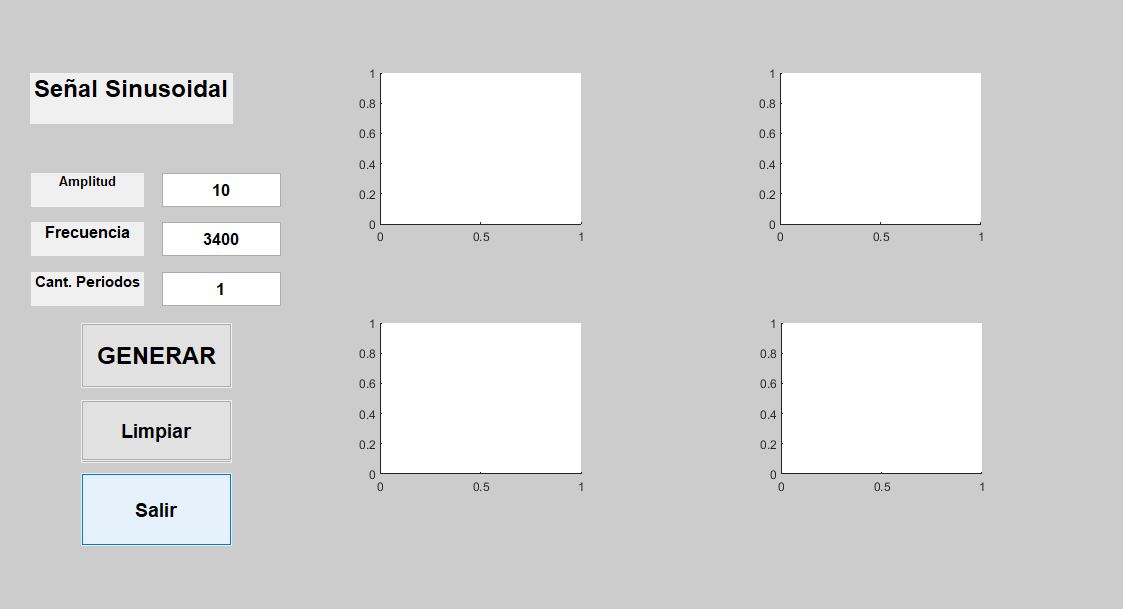


Figura (23) Aplicación botón salir

* 1. **Serie Fourier**

La idea básica de las series de Fourier es que toda función periódica de periodo T puede ser expresada como una suma trigonométrica d}de senos y cosenos del mismo periodo T. El problema aparece naturalmente en astronomía, de hecho, Neugebauer (1952) descubrió que los Babilonios utilizaron una forma primitiva de series de Fourier en la predicción de ciertos eventos celestiales.

(“Series de Fourier, Transformada de Furier y Aplicaciones”, Genaro Gonzalez slideplayer.es. Serie de Fourier. Transformada de Fourier, Retrieved August 7, 2019, from <https://slideplayer.es/slide/1126251/>)

Las funciones periódicas que se presentan en problemas prácticos con frecuencia son bastante complicadas y es deseable representarlas en términos de funciones periódicas simples. Se verá que casi cualquier función periódica f (τ) de periodo 2… que aparezca en las aplicaciones (por ejemplo, con relación a vibraciones) puede representarse por una serie trigonométrica la cual se denominará serie de Fourier de f.

Las series de Fourier surgen de la tarea práctica de representar una función periódica f (τ) dada en términos de funciones coseno y seno. Estas series son trigonométricas cuyos coeficientes se determinan a partir de f (τ) mediante ciertas fórmulas (fórmulas de Euler), las cuales se establecerán primero. Después se considerará la teoría de las series de Fourier.

(“Domingo Alcaraz Candela, Series y transformadas de Fourier, Serie de Fourier”, Retrieved August 7, 2019, from <http://filemon.upct.es/~paredes/am_ti/apuntes/Tema%202.%20Series%20y%20transformadas%20de%20Fourier.pdf>)

* 1. **Serie Fourier: generador de señal**

Este modelo estará basado en la introducción al tema de la transformación de Fourier. Para este modelo practico realizamos un generador de señal el cual producirá una señal dependiente del tiempo con unas características determinadas de frecuencia, amplitud y forma. Algunas veces estas características son externamente controladas a través de señales de control.

Dado la ecuación:

* 1. **Código de la señal Fourier en Matlab**

1. clc
2. clear all
3. close all
5. %---DATA IN---
6. A=1; %--Amplitud
7. f=3400; %--Frecuencia
8. cantper=1; %--periodo
10. %---PROCESS---
11. t = linspace(0,cantper/f,500); );%--Aquí se maneja como "T" el Periodo de nuestra señal
12. xt\_1cf = (A/1)\*sin(2\*pi\*1\*f\*t); %--ecuación de la señal
13. xt\_2cf = xt\_1cf + (A/3)\*sin(2\*pi\*3\*f\*t);
14. xt\_3cf = xt\_2cf + (A/5)\*sin(2\*pi\*5\*f\*t);
15. xt\_4cf = xt\_3cf + (A/7)\*sin(2\*pi\*7\*f\*t);
16. xt\_5cf = xt\_4cf + (A/9)\*sin(2\*pi\*9\*f\*t);
17. xt\_6cf = xt\_5cf + (A/11)\*sin(2\*pi\*11\*f\*t);
19. %---OUTPUT---
20. figure(1)
21. subplot(3,2,1)
22. plot(t,xt\_1cf),title('componentes de frecuencia: 1'), grid on
23. subplot(3,2,2)
24. plot(t,xt\_2cf),title('componentes de frecuencia: 2'), grid on
25. subplot(3,2,3)
26. plot(t,xt\_3cf),title('componentes de frecuencia: 3'), grid on
27. subplot(3,2,4)
28. plot(t,xt\_4cf),title('componentes de frecuencia: 4'), grid on
29. subplot(3,2,5)
30. plot(t,xt\_5cf),title('componentes de frecuencia: 5'), grid on
31. subplot(3,2,6)
32. plot(t,xt\_6cf),title('componentes de frecuencia: 6'), grid on

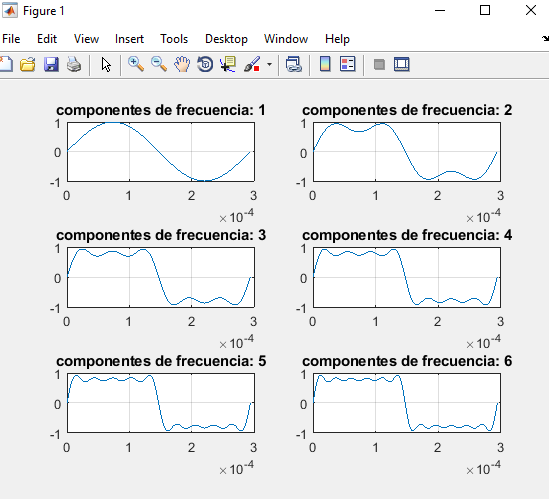
****

Figura (24) Modelo de la Generación de señal Fourier

* 1. **Código de la señal Fourier generando funciones Sinusoidal, Cuadrada, Triangular, Diente Sierra en Matlab**

1. clc
2. clear all
3. close all
4. %-- DATA IN--%
5. A=1; %--Amplitud
6. f=3400; %--Frecuencia
7. cantper=1; %--periodo
8. comp\_freq=6; %--Componente Frecuencia
9. phi=0;
11. %-- PROCESS--%
12. t=linspace(0,cantper/f,500);
14. %señal sinusoidal basica
15. sinusoidal = A\*sin(2\*pi\*f\*t + phi);
17. %Señal cuadrada
18. cuadrada =0;
19. for n=1:2:(2\*comp\_freq-1) %comience en n=1 con pasos de :2:
20. cuadrada = cuadrada +(A/n)\*sin(2\*pi\*f\*n\*t);
21. end
23. %Señal triangular
24. triangular =0;
25. for n=1:2:(2\*comp\_freq-1) %comience en n=1 con pasos de :2:
26. triangular = triangular +(A/n^2)\*cos(2\*pi\*f\*n\*t);
27. end
29. %Señal diente sierra
30. dsierra =0;
31. for n=1:comp\_freq %comience en n=1 con pasos de :1:
32. if(mod(n,2) ==0)
33. signo=-1;
34. else
35. signo=1;
36. end
37. dsierra = dsierra + signo\*(A/n)\*sin(2\*pi\*f\*n\*t);
38. end

41. %-- OUTPUT--%
42. figure(1);
43. subplot(2,2,1)
44. plot(t,sinusoidal),title('Sinusoidal'),grid on
45. subplot(2,2,2)
46. plot(t,cuadrada),title('Cuadrada'),grid on
47. subplot(2,2,3)
48. plot(t,triangular),title('Triangular'),grid on
49. subplot(2,2,4)
50. plot(t,dsierra),title('Diente Sierra'),grid on

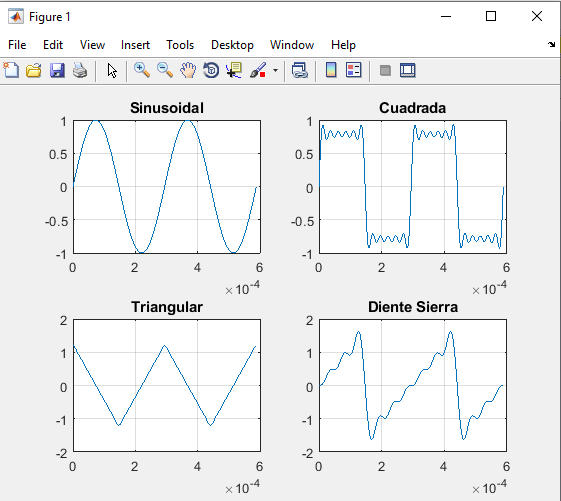
****

Figura (25) Modelo de la Generación de señales Sinusoidal, Cuadrada, Triangular, Diente Sierra.

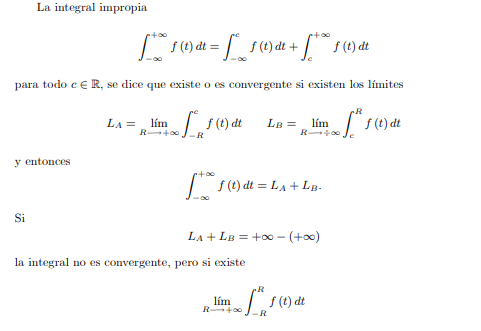
* 1. **Transformada de Fourier:**

Dada una función f: R −→ C llamaremos transformada de Fourier de f a la función compleja

para todo z ∈ R donde la expresión anterior tenga sentido, es decir, donde la integral impropia anterior sea convergente.

Esta convergencia es más difícil de verificar que en el caso de la transformada de Laplace. Supongamos por ejemplo que τ y z son reales, por lo que , que como sabemos tiene módulo 1. Si f (t) es también real, para garantizar la convergencia absoluta de la integral anterior debe satisfacerse que

por lo que las funciones reales que tendrán transformada de Fourier tienen a este límite le llamaremos valor principal de Cauchy de la integral impropia. Este valor principal coincide con el de la integral cuando ésta es convergente, por lo que en la integral impropia de la integral de Fourier se considera su valor principal.



(“Domingo Alcaraz Candela, Series y transformadas de Fourier, Serie de Fourier”, Retrieved August 9, 2019, from <http://filemon.upct.es/~paredes/am_ti/apuntes/Tema%202.%20Series%20y%20transformadas%20de%20Fourier.pdf>)

**4.1. Transformada de Fourier: analizador de espectro.**

Para realizar este modelo practico, tomaremos los datos del generador de señales y procedemos a descomponerlo en pequeñas partes simples en la cual vamos a ser capaces de observar el valor eficaz o la potencia de la cada una de las componentes en frecuencia, que tengamos en cada intervalo de medida de la señal de entrada.

**4.2. Código de la señal Transformada de Fourier en Matlab**

clc

clear all

close all

%-- DATA IN--%

A=1; %--Amplitud

f=3400; %--Frecuencia

cantper=1; %--periodo

%-- PROCESS--%

t=linspace(0,cantper/f,500);

xt\_1cf =(A/1)\*sin(2\*pi\*1\*f\*t);

xt\_2cf = xt\_1cf+(A/3)\*sin(2\*pi\*3\*f\*t);

xt\_3cf = xt\_2cf+(A/5)\*sin(2\*pi\*5\*f\*t);

xt\_4cf = xt\_3cf+(A/7)\*sin(2\*pi\*7\*f\*t);

xt\_5cf = xt\_4cf+(A/9)\*sin(2\*pi\*9\*f\*t);

xt\_6cf = xt\_5cf+(A/11)\*sin(2\*pi\*11\*f\*t);

%--codigo del espectro de la transfoemada

tf\_1cf = abs(fft(xt\_1cf));

tf\_2cf = abs(fft(xt\_2cf));

tf\_3cf = abs(fft(xt\_3cf));

tf\_4cf = abs(fft(xt\_4cf));

tf\_5cf = abs(fft(xt\_5cf));

tf\_6cf = abs(fft(xt\_6cf));

%-- OUTPUT--%

figure(1);

subplot(3,2,1)

plot(t,xt\_1cf),title('componentes de frecuencia: 1'),grid on

subplot(3,2,2)

plot(t,xt\_2cf),title('componentes de frecuencia: 2'),grid on

subplot(3,2,3)

plot(t,xt\_3cf),title('componentes de frecuencia: 3'),grid on

subplot(3,2,4)

plot(t,xt\_4cf),title('componentes de frecuencia: 4'),grid on

subplot(3,2,5)

plot(t,xt\_4cf),title('componentes de frecuencia: 5'),grid on

subplot(3,2,6)

plot(t,xt\_4cf),title('componentes de frecuencia: 6'),grid on

figure(2);

subplot(3,2,1)

plot(tf\_1cf(1:15)),title('Transformada Fourier con 1 componentes de frecuencia'),grid on

subplot(3,2,2)

plot(tf\_2cf(1:15)),title('Transformada Fourier: con 2 componentes de frecuencia'),grid on

subplot(3,2,3)

plot(tf\_3cf(1:15)),title('Transformada Fourier: con 3 componentes de frecuencia'),grid on

subplot(3,2,4)

plot(tf\_4cf(1:15)),title('Transformada Fourier: con 4 componentes de frecuencia'),grid on

subplot(3,2,5)

plot(tf\_5cf(1:15)),title('Transformada Fourier: con 5 componentes de frecuencia'),grid on

subplot(3,2,6)

plot(tf\_6cf(1:15)),title('Transformada Fourier: con 6 componentes de frecuencia'),grid on

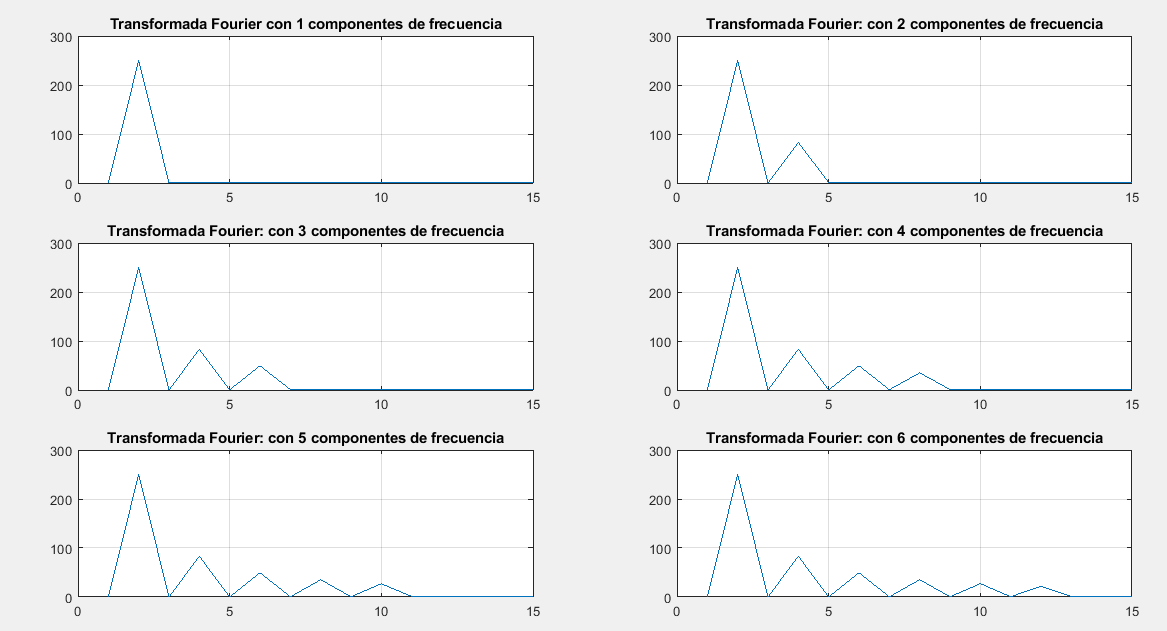


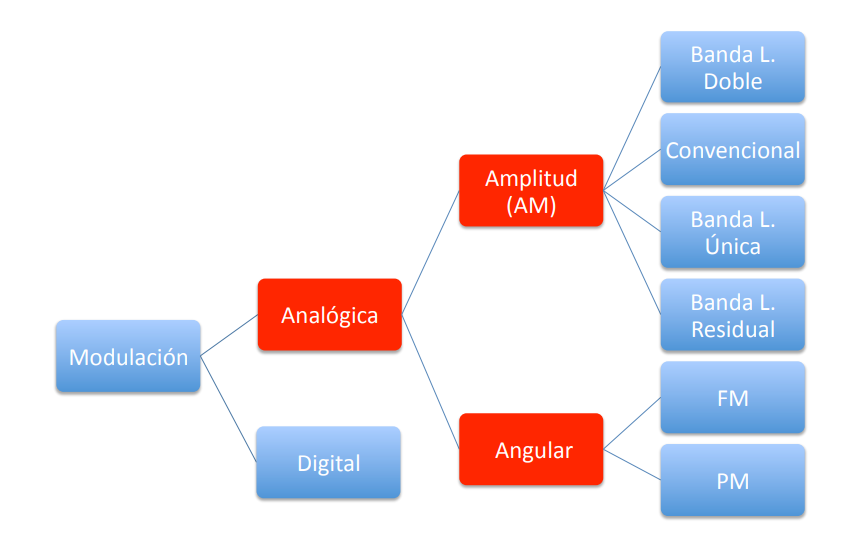
Figura (26) Modelo de la transformada de Fourier.

* 1. **Modulación de simulación: analógica + digital**

**5.1. Modulación**

Podemos definirlo como un conjunto de técnicas que se utilizan para transportar la información a través de un canal de comunicación y recuperar la señal en su forma original en la otra extremidad.

**5.1.1. Tipos de Modulación**



(Licamartino.rog. Tipos de Modulaciones, Retrieved August 11, 2019, from <http://www.lucamartino.altervista.org/Presentation1.pdf>)

**5.2. Modulación Análoga**

La modulación Análoga es el sistema de una onda portadora de acuerdo con el mensaje (señal modulada) y puede ser también una codificación modulación analógica con portadora analógica se utiliza cuando se desea transmitir la señal analógica a una frecuencia diferente o con un ancho de banda menor. La modulación se puede realizar utilizando cambios de amplitud, frecuencia o fase de la señal portadora.

(Ceti. Modulaciones análogas y digitales, Retrieved August 11, 2019, from <https://ceti-y-las-comunicaciones.webnode.mx/parcial-2/modulacion-analogica-y-digital/>)

El las cuales hay tres técnicas de modulación básica son:

* Modulación de la amplitud (AM o amplitud modulada).
* Modulación de la frecuencia (FM o frecuencia modulada).
* Modulación de la portadora (PM o portadora sin modular)

**5.2.1. Modulación Am**

Podemos definirlo como un dispositivo de dos señales de entrada, una es la señal portadora de amplitud y frecuencia constante, y la otra es la señal de información o modulación.

La señal Am es una mezcla de señales de AF y RF, de manera que estas varían según la amplitud de la señal AF (modulación) alteran la amplitud de la señal RF (portadora).

Esta señal está dada por:

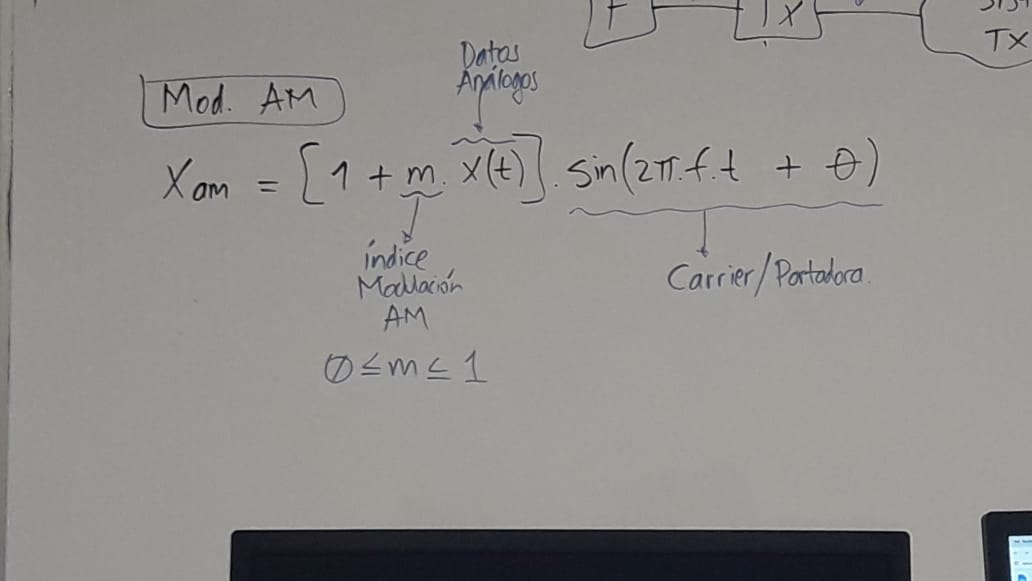


Figura (27) Ecuación de frecuencia Am

**5.2.2. Modulación Fm**

Es una modulación angular que transmite información a través de una onda portadora variando su frecuencia.

Es una señal de banda bastante estrecha de transmitir, se utiliza para comunicaciones de voz en la radio comercial y en las configuraciones de aficionados. El tipo de radiodifusión FM es generalmente llamado amplia-FM o W-FM (de las siglas en inglés "Wide-FM"). En la radio de dos vías, la banda estrecha o N-FM (de las siglas en inglés "Narrow-FM") es utilizada para ahorrar ancho de banda. Además, se utiliza para enviar señales al espacio.

Cuando se desarrolló esta frecuencia estaba modulada por MF, por lo que estaba saturada por transmisiones de radio, por lo que se adjuntó la banda de VHF, espectro que ofrecía grandes posibilidades de transmisión de la señal.

Esta señal está dada por:

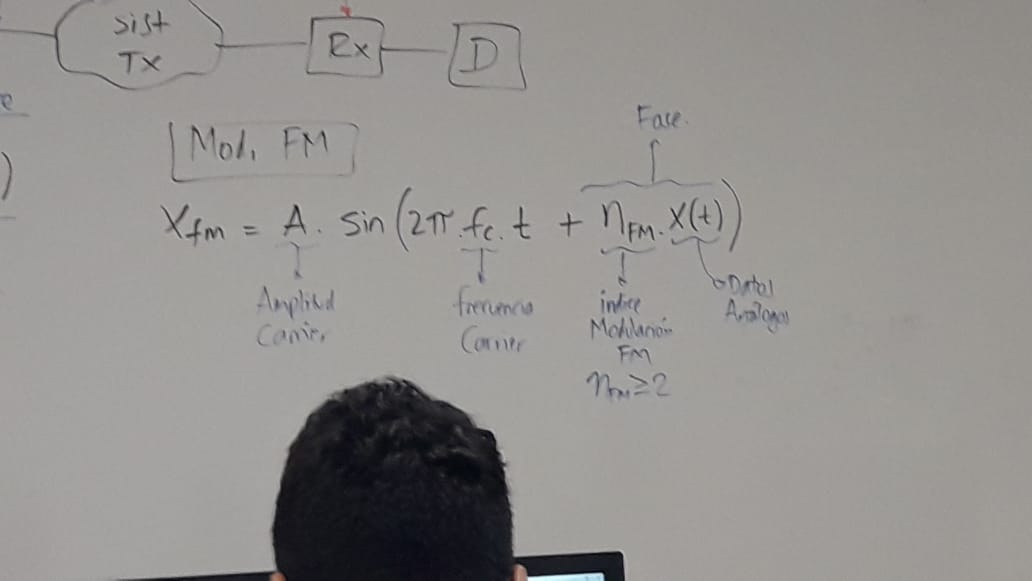


Figura (28) Ecuación de frecuencia Fm

**5.2.3. Modulación Pm**

Las señales de datos PM es un tipo de modulación exponencial al igual que la modulación de frecuencia. En este caso el parámetro de la señal portadora que variará de acuerdo a señal moduladora es la fase.

La modulación de fase (PM) no es muy utilizada principalmente por que se requiere de equipos de recepción más complejos que en FM y puede presentar problemas de ambigüedad para determinar por ejemplo si una señal tiene una fase de 0º o 180º.

Esta señal está dada por:

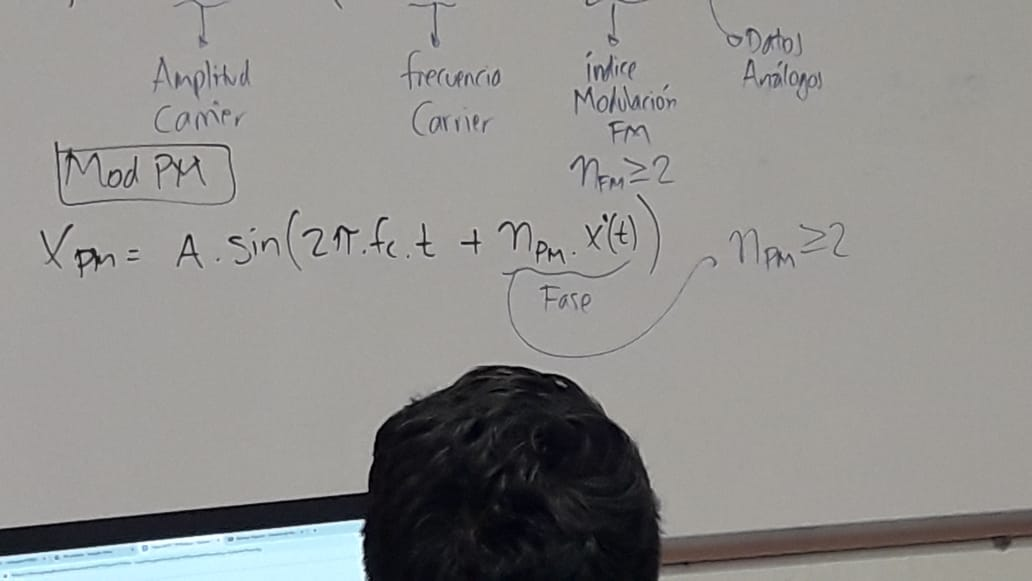


Figura (29) Ecuación de frecuencia Pm

**5.2.4 Código de la señal Modulada AM, PM y Fm en Matlab**

clc

clear all

close all

%---- DATA IN --

% -- carrier

Ac = 10;

fc = 1000e3;

cpc= 18;

%datos analogos

Ax = 2;

fx = 3400;

cpx = 2;

%-- Indice de modulacion AM

m = 0.4;

%-- Indice de modulacion FM

Nfm = 2;

%-- Indice de modulacion PM

Npm = 2;

%--- PROCESS ---

% --- construccion carrier

tc= linspace(0,cpc/fc,500);

carrier = Ac\*sin(2\*pi\*fc\*tc);

%contruccion datos analogos

tx = linspace(0,cpx/fx,500);

xt = Ax\*sin(2\*pi\*fx\*tx);

%modulacion Am

Xam = (1+ m\*xt).\*carrier;

%modulacion Fm

Xfm = Ac\*sin(2\*pi\*fc\*tc + Nfm\*xt);

%modulacion Pm

xt\_diff = Ax\*cos(2\*pi\*fx\*tx);

Xpm = Ac \* sin(2\*pi\*fc\*tc + Npm\*xt\_diff);

%--- OUTPUT ---

figure(1)

subplot(5,1,1),plot(tc,carrier), title('carrier'),grid on

subplot(5,1,2),plot(tx,xt), title('Datos Análogos'),grid on

subplot(5,1,3),plot(tc,Xam), title('Modulacion Am'),grid on

subplot(5,1,4),plot(tc,Xfm), title('Modulacion Fm '),grid on

subplot(5,1,5),plot(tc,Xpm), title('Modulacion Pm '),grid on

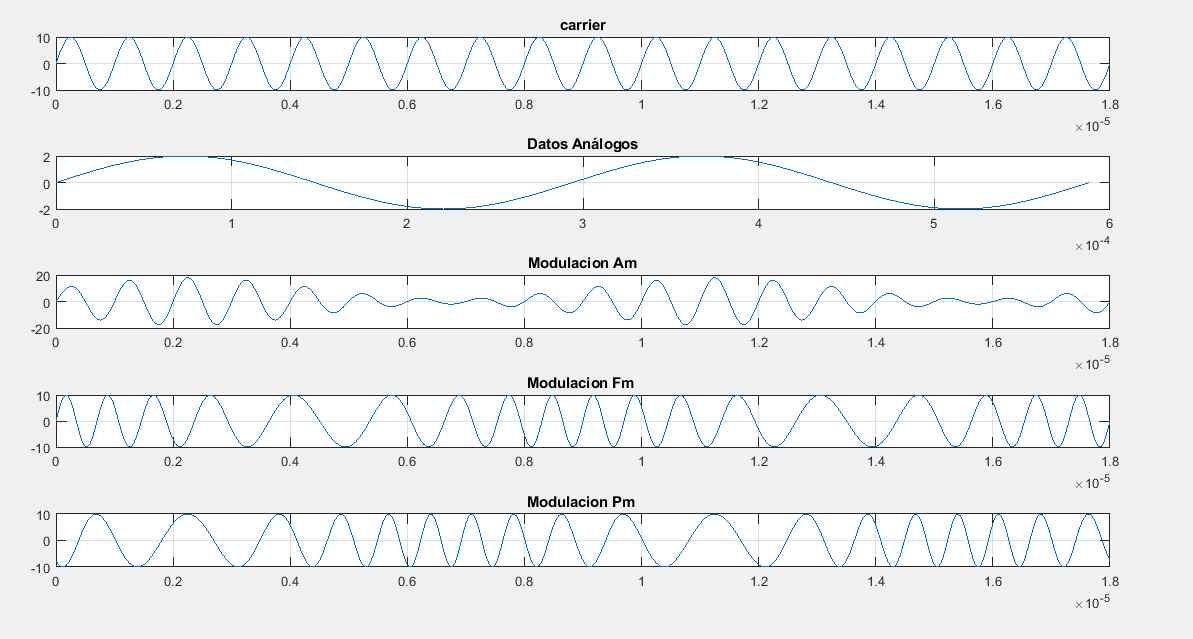


Figura (25) Modelo de señales Análogas

**5.3. Modulación Digital**

Es un proceso mediante el cual se trasforman los símbolos digitales en forma de onda adecuadas para la transmisión sobre un canal de comunicación.

* 1. **Bibliografía**

Stallings,William. Comunicaciones y redes de computadores. Prentice Hall, 2004.Séptima Edición.

Jesús Monzo Espín, Function Sinusoidal, Retrieved August 6, 2019, from <https://www.geogebra.org/m/eAkQEpyP>

Riverglennapts.com. Fundamentos sinusoidal, Retrieved August 6, 2019, from <https://riverglennapts.com/es/fundamentals/432-sinusoidal-wave-signal.html>

M. C. José Jaime Esqueda Elizondo, Matlab e Interfaces Gráficas, noviembre, 2002

Slidershare.net. Teoria de Fourier, Retrieved August 7, 2019, from <https://es.slideshare.net/walter13333/teora-de-fourier>

slideplayer.es. Serie de Fourier. Transformada de Fourier, Retrieved August 7, 2019, from <https://slideplayer.es/slide/1126251/>

Domingo Alcaraz Candela. Teoria de Fourier. Serie de Fourier. Transformada de Fourier, Retrieved August 7, 2019, from <http://filemon.upct.es/~paredes/am_ti/apuntes/Tema%202.%20Series%20y%20transformadas%20de%20Fourier.pdf>

Licamartino.rog. Tipos de Modulaciones, Retrieved August 11, 2019, from <http://www.lucamartino.altervista.org/Presentation1.pdf>

Ceti. Modulaciones análogas y digitales ,Retrieved August 11, 2019, from <https://ceti-y-las-comunicaciones.webnode.mx/parcial-2/modulacion-analogica-y-digital/>