

PRACTICA No. 1

TEMA: “INTRODUCCIÓN A MATLAB Y SIMULINK”

1. OBJETIVOS

- Familiarizar al estudiante con el uso de Matlab.
- Familiarizar al estudiante con el paquete de SIMULINK.
- Aplicar Matlab y simulink para analizar las señales en el dominio del tiempo.

2. INTRODUCCIÓN

2.1. MATLAB

Es un programa interactivo para computación numérica y visualización de datos. Este integra cálculos, programación y visualización en una herramienta flexible, con arquitectura abierta, diseñada para el manejo de proyectos a gran escala en investigación y la industria. MATLAB proporciona tanto a ingenieros como científicos un lenguaje intuitivo para presentar problemas y sus soluciones en forma matemática y gráfica; como ejemplo de ello se tiene el gráfico de la Figura 1 donde se representa el gráfico de una función de dos variables en tres dimensiones, de la misma manera que se pueden representar funciones, se pueden representar datos y realizar mediciones de magnitudes físicas.

MATLAB permite el desarrollo de aplicaciones con la ayuda de una interfaz de construcción gráfica (GUI: *Graphical User Interface*).

Esta herramienta posee una variedad de soluciones para aplicaciones específicas, denominada toolboxes, que son una colección de archivos escritos en lenguaje de MATLAB (archivos.m) los cuales realizan operaciones orientadas a solucionar problemas específicos. Las áreas que cubren estos archivos.m son por ejemplo: procesamiento digital de señales, sistemas de control, simulación y muchos otros.

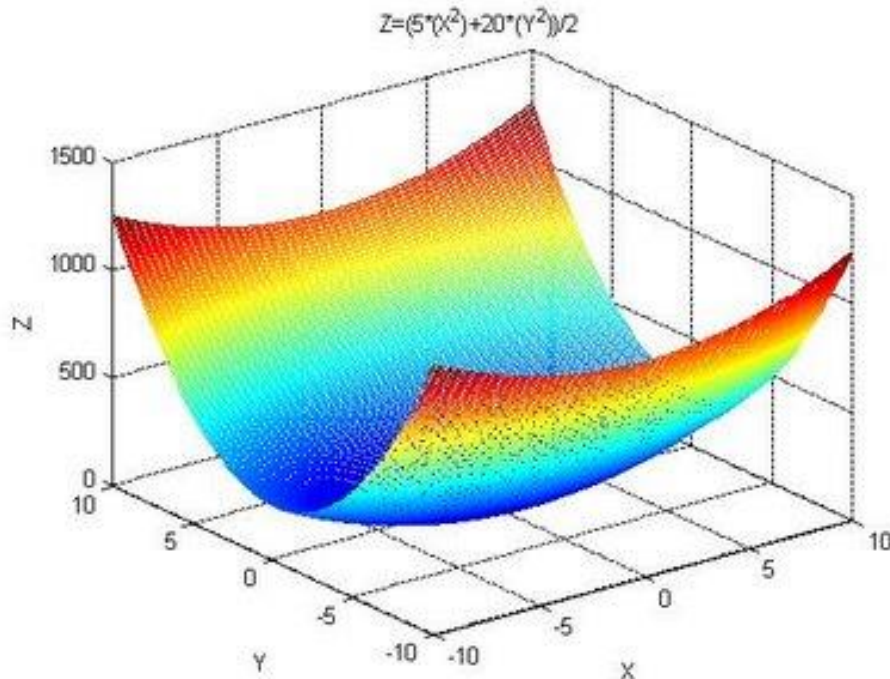


Figura 1: Representación en tres dimensiones de una función $Z = (5x^2 + 20y^2)/2$

2.1.1. INTRODUCCIÓN AL USO DE MATLAB

El programa consiste básicamente, de un área de memoria (workspace) de datos y una línea de comandos (prompt). Se almacena en la memoria con variables literales cualquier valor en forma de matriz, para luego ser utilizado en funciones que pueden ser matemáticas o de un fin específico, como por ejemplo el comando *plot*, el cual presenta un gráfico de los datos ingresados.

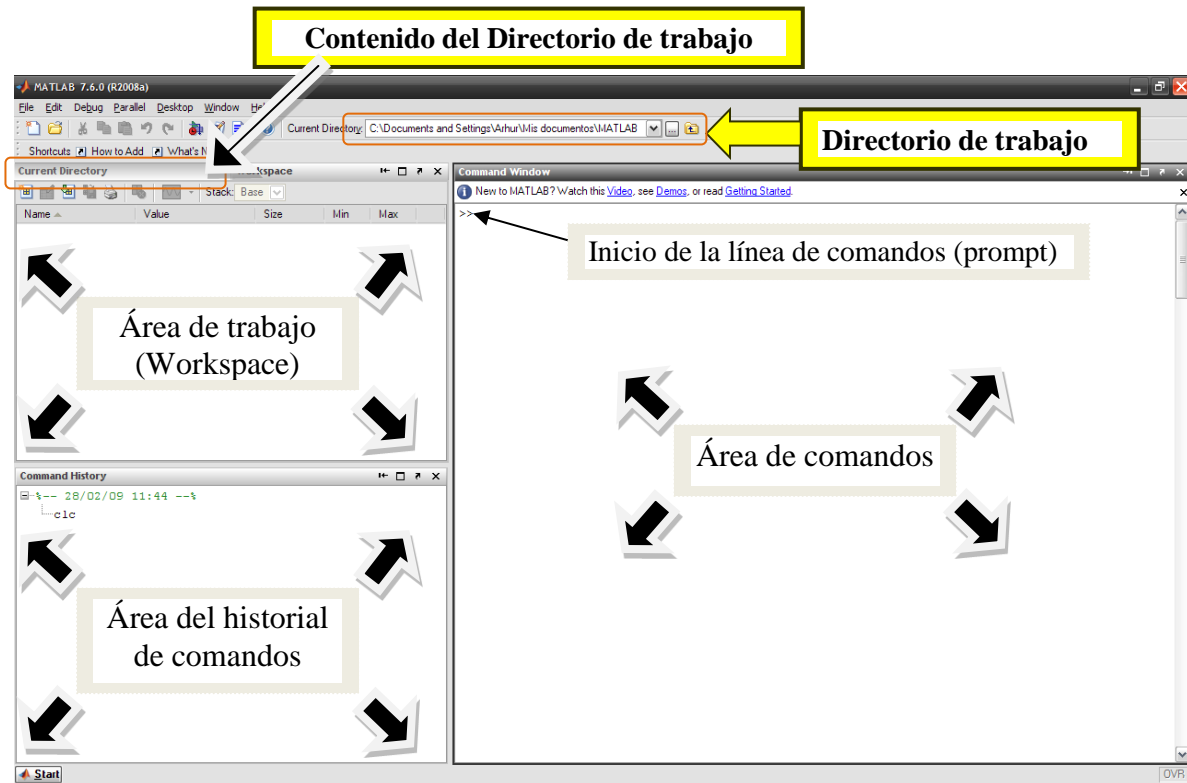


Figura 2: Entorno de trabajo de Matlab.

- **DIRECTORIO DE TRABAJO**, guarda todos los archivos implicados del trabajo como scripts, .m , .gui, etc. Para evitar errores todos los archivos a utilizar deben ser ubicados en el directorio activo definido por el usuario.
- **AREA DE TRABAJO**, en donde residen todas las variables creadas en cada sesión de trabajo, lo especial de esta área es que las variables a pesar de ser de tipo volátil residirán ahí de modo que al cerrar el programa y abrirlo nuevamente se mantienen, hasta ejecutar el comando `clear [variable]`, para eliminar la [variable] específica o `clear all`, para eliminar todas las variables.
- **AREA DEL HISTORIAL DE COMANDOS**, se detalla cada uno de los comandos invocados en el prompt del Matlab, cada vez que se abre el programa se agrupan todos los comandos de esa sesión con fecha y hora, esta funcionalidad hace que los comandos sean reciclables dando doble click sobre ellos, también se puede volver a invocar comandos anteriores ubicándonos en el prompt y presionado la tecla de la flecha de arriba.

En la línea de inicio del prompt del Matlab se encuentra el símbolo:

>>

Desde aquí se puede invocar el uso de cualquier comando una función del Matlab.

Como un pequeño ejemplo ilustrativo se genera una señal sinusoidal de una frecuencia de 50 [Hz] durante un segundo, se reduce su duración a un ciclo. Para esto es necesario definir una frecuencia de muestreo, debido a que Matlab trabaja de manera digital.

Con el código que se presenta a continuación se puede generar el ejemplo descrito, una señal de 50 [Hz] con una duración de un segundo y una amplitud definida de 5 (por ejemplo voltios, amperios, etc.).

```

>> %Definimos una frecuencia de muestreo
>> fs=1000; %Se tomara mil muestras en determinado tiempo.
>> %Definimos un vector de duracion de la señal
>> t=0:1/fs:1; %La señal durara un segundo, y esta dada en pasos de 1/fs es decir
>> % que se tomaran 1000 muestras en total en un segundo.
>> f=50;%Señal de frecuencia = 50 [Hz]
>> A=5; %Amplitud de la señal generada
>> senal_generada=A*sin(2*pi*f*t);%Vector de la señal generada.
>> %Ploteamos la señal en pantalla
>> plot(t,senal_generada)
>>
    
```

Name	Value	Size	Min	Max
A	5	1x1	5	5
f	50	1x1	50	50
fs	1000	1x1	1000	1000
senal_generada	<1x1001 double>	1x1001	-5	5
t	<1x1001 double>	1x1001	0	1

Figura 3: Código de una Función Sinusoidal

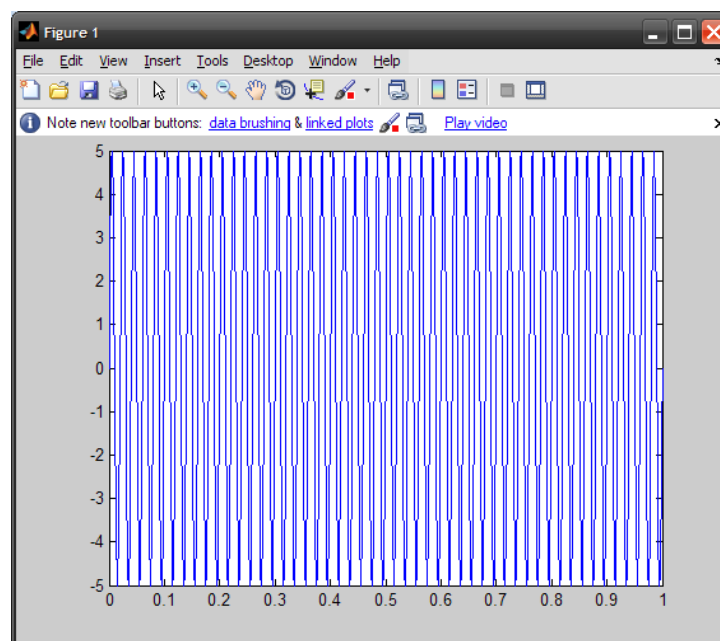


Figura 4: Función Sen(x)

Una forma de solo un ciclo de la señal se aplica a continuación:

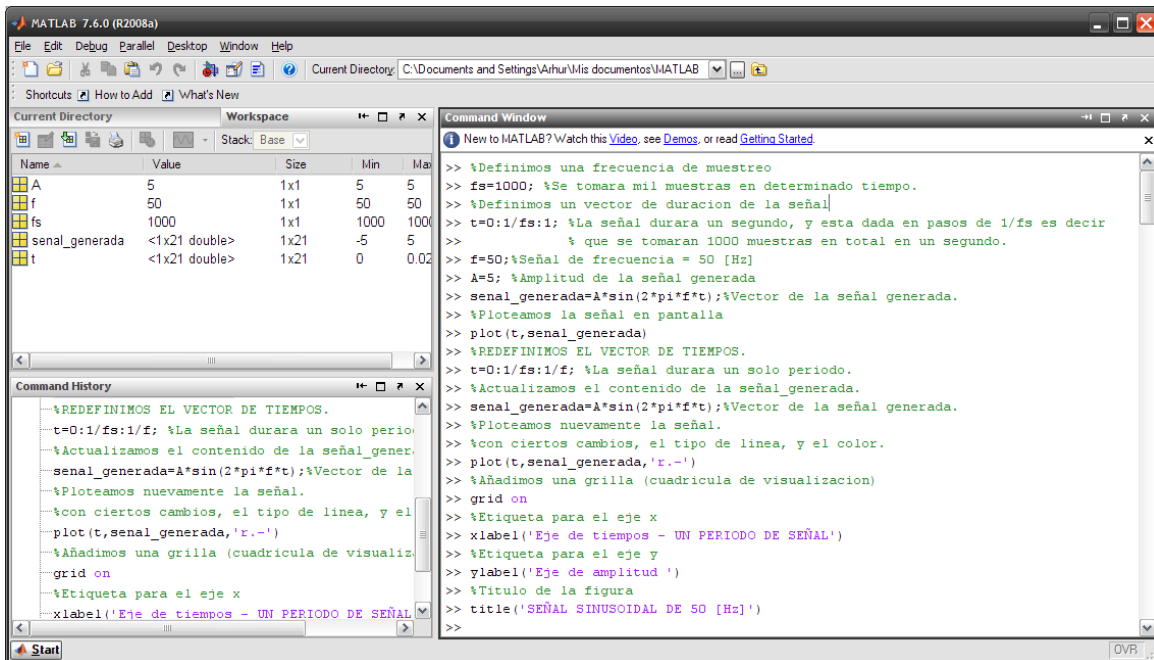


Figura 5: Área de trabajo del Matlab con las nuevas líneas de comando.

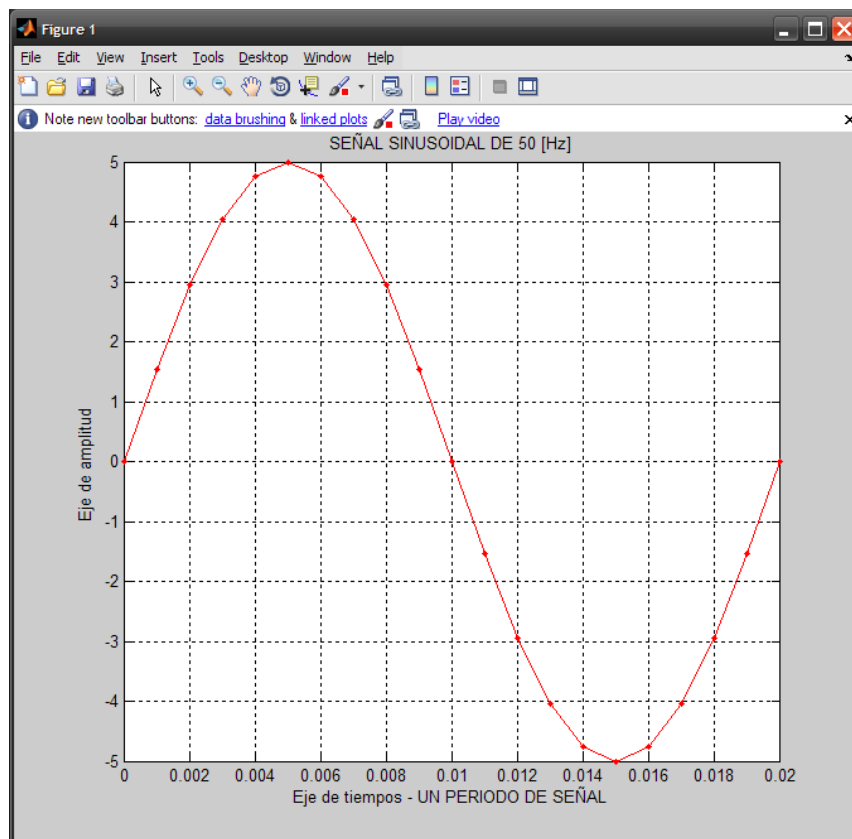


Figura 6: Señal sinusoidal un ciclo.

Para no repetir un grupo de instrucciones y optimizar el uso del programa, o con el fin de hacer funciones de usuario, se agrupan una serie de comandos en un archivo que se guarda con la extensión “*.m”, donde * será el nombre del archivo que se genera.

Los archivos .m son aquellos que representan un listado de comandos o funciones, son hechos en lenguaje de texto común, escritos en cualquier editor de texto (por ejemplo *notepad.exe*) propio de *Microsoft Windows* o cualquier otro. En MATLAB se tiene un editor incorporado, el cual tiene una visualización de sintaxis con colores y además proporciona la facilidad directa de realizar depuraciones controladas del flujo de programa que representa el archivo.

A continuación se crea un archivo .m como lo indica la Figura 7.

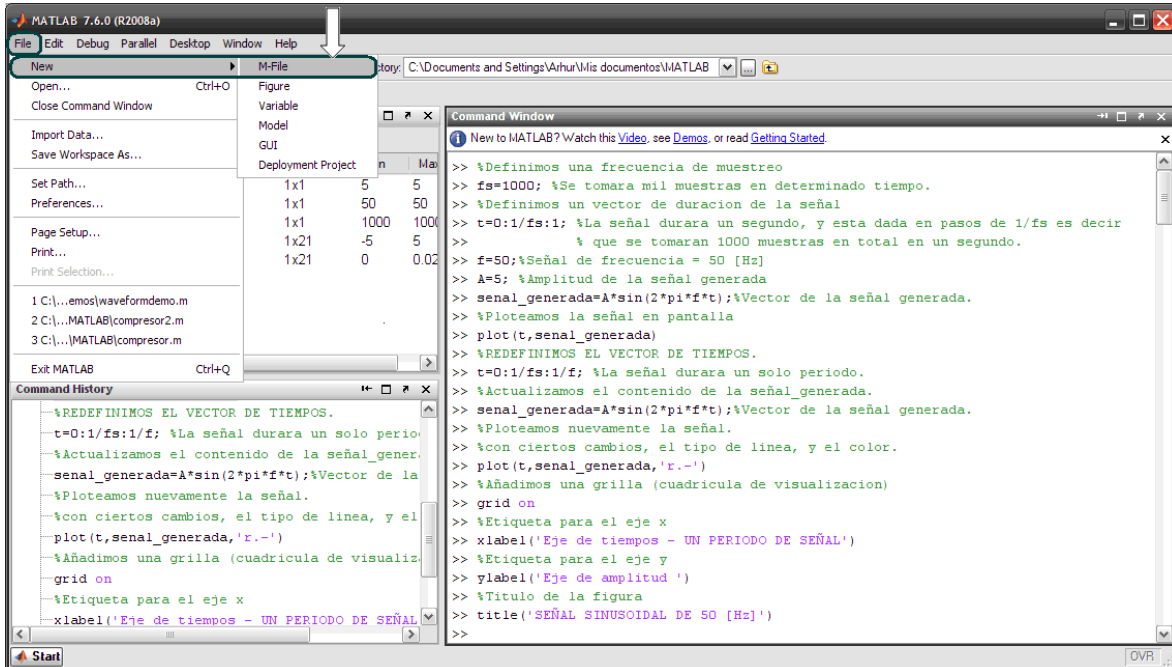


Figura 7: Creación de un archivo .m

El archivo, se guardará en el directorio de trabajo para evitar errores en la ejecución del archivo.m .

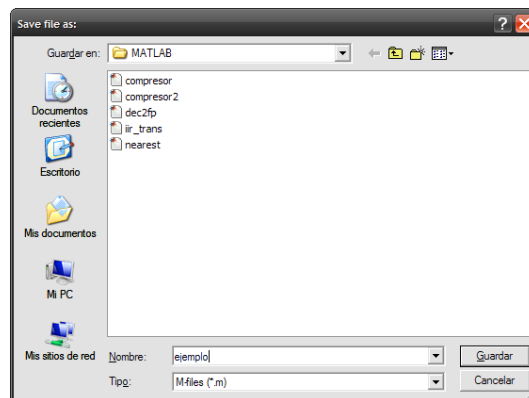



Figura 8: Guardando el archivo .m

```

1 %%Ejemplo
2 %Definimos una frecuencia de muestreo
3 fs=1000; %Se tomara mil muestras en determinado tiempo.
4 %Definimos un vector de duracion de la señal
5 t=0:1/fs:1; %La señal durara un segundo, y esta dada en pasos de 1/fs es decir
6 % que se tomaran 1000 muestras en total en un segundo.
7 f=50;%Señal de frecuencia = 50 [Hz]
8 A=5; %Amplitud de la señal generada
9 senal_generada=A*sin(2*pi*f*t);%Vector de la señal generada.
10 %Ploteamos la señal en pantalla
11 plot(t,senal_generada)
12 %REDEFINIMOS EL VECTOR DE TIEMPOS.
13 t=0:1/fs:1/f; %La señal durara un solo periodo.
14 %Actualizamos el contenido de la señal generada.
15 senal_generada=A*sin(2*pi*f*t);%Vector de la señal generada.
16 %Ploteamos nuevamente la señal.
17 %con ciertos cambios, el tipo de linea, y el color.
18 plot(t,senal_generada,'r.-')
19 %Añadimos una grilla (cuadrícula de visualizacion)
20 grid on
21 %Etiqueta para el eje x
22 xlabel('Eje de tiempos - UN PERIODO DE SEÑAL')
23 %Etiqueta para el eje y
24 ylabel('Eje de amplitud ')
25 %Titulo de la figura
26 title('SEÑAL SINUSOIDAL DE 50 [Hz]')
27 %% FIN
28

```

Figura 9: Entorno de trabajo del Editor de archivos de Matlab con el código del ejemplo cargado, y guardado como ejemplo.m

Hay dos formas de ejecutar el código generado en el archivo, con la tecla F5, run [archivo.m], o dando click en el siguiente botón de la barra de herramientas. 

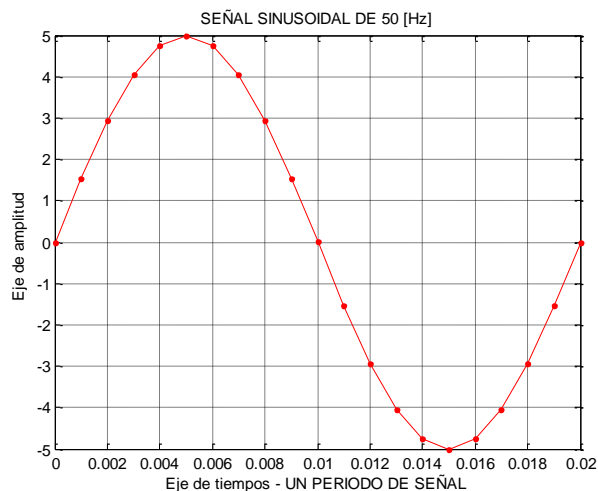


Figura 10: Resultado final de ejecutar el archivo ejemplo.m

2.2 SIMULINK

2.1.2 Características de SIMULINK

SIMULINK es una herramienta para modelar y simular una gran variedad de sistemas dinámicos, los cuales pueden ser lineales, no lineales, en tiempo discreto, en tiempo continuo y sistemas que son una combinación de éstos. El programa provee la facilidad de crear modelos a simular, como si se tratara de animar un diagrama de bloques de un texto, los elementos de estos diagramas de bloques se obtienen de librerías copiándolos gráficamente y modificando sus variables con ayuda del ratón, una vez generado el modelo se puede visualizar la simulación. A continuación se indican las características algunos modelos a simularse.

Sistemas de Modelación a través de diagramas de bloques

Para representar sistemas que se desean analizar se ha desarrollado una técnica ilustrativa en los textos técnicos, los denominados diagramas de bloques, en *SIMULINK* para la simulación de sistemas se parte de esta idea intuitiva de los diagramas de bloques para generar el modelo a simular.

Las características de la generación de Modelos para la simulación son las siguientes:

- **Modelos concatenados**, es decir que se crean sistemas complejos a partir de sistemas simples, sin límites de bloques o conexiones, esto se realiza agrupando bloques para que trabajen como subsistemas (un sistema dentro otro) de esta manera se puede generar un proyecto sin perder la visión de su aplicación.
- **Librería de bloques**, la cual contiene los elementos más usados para la simulación como *sources* (generadores de funciones), *sinks* (equipos de visualización de datos), *Discrete* (para sistemas discretos), *linear* (sistemas lineales), *no linear* (sistemas no lineales) y *connections* (conexiones, como entradas en un subsistema o interconexiones con subsistemas, etc.).
- **Conexiones escalares y vectoriales**, los modelos pueden tener bloques de entradas y salidas de una sola señal o múltiples señales dependiendo de la definición de cada bloque.
- **Asignación de nombres a señales y puertos**, es muy útil rotular las líneas de conexión para constancia y estudio de la misma, así como también poner nombre a los puertos de entrada y salida de un bloque o subsistema creado.

Resumen de las librerías del *SIMULINK*:

Source (fuentes)

- ♦ Generación de señales senoidales, rampas y ondas cuadradas.
- ♦ Generador de ruido.

Sinks (sumideros)

- ♦ Presentación de datos (Scope, campo), visión numérica y bloques de gráficos.
- ♦ Transferencia de datos a archivos.
- ♦ Salidas de bloques al campo de trabajo MATLAB.

Discrete (Sistemas discretos)

- ♦ Funciones de Transferencia (Transformada Z), filtros, retardos y bloques descritos por ecuaciones en diferencias.

Linear (Sistemas lineales)

- ♦ Funciones de transferencia (Transformada de Laplace), bloques de ganancia y sistemas descritos por ecuaciones diferenciales.
- ♦ Integradores.
- ♦ Diferenciadores.

Nolinear (Sistemas no lineales)

- ♦ Limitadores, bloques con histéresis y muestreadores de señal.
- ♦ Operadores lógicos y relacionadores de señal.

Connections (Conexiones)

- ♦ Bloques multiplexores y demultiplexores analógicos de señal.
- ♦ Puertos de entrada y salida de subsistemas.
- ♦ Habilitación de un subsistema por estado (*Enable*) o por cambio de estado (*trigger*).

Simulación y análisis

SIMULINK y *MATLAB* permiten al usuario pasar sin problemas de un estado de análisis a diseño y simulación; en la simulación pueden variarse los parámetros de los bloques al mismo tiempo que se está efectuando la simulación y observar los resultados casi instantáneamente.

Con los resultados de la simulación se pueden extraer modelos lineales, optimización de parámetros, análisis paramétrico (variación de varios parámetros al mismo tiempo) e incluso animación, ya que mediante un bloque programable denominado *S-function* permite asociar los datos de entrada con comandos del *MATLAB* y conseguir la animación.

Arquitectura abierta

SIMULINK facilita ampliar los modelos de simulación, personalizar los bloques ya existentes y crear librerías.

Existen cuatro maneras de añadir bloques especializados a la librería de bloques:

- ♦ Agrupando bloques en un subsistema.
- ♦ Con la ayuda de la interfaz de personalizar un bloque.
- ♦ Mediante algoritmos en *C* y *Fortran*.
- ♦ Empleando los archivos.m de *MATLAB*.

SIMULINK permite personalizar un bloque mediante un interfaz denominado *MASK*, el cual crea una ventana de dialogo con la facilidad de poner nombres a las variables, iconos y comentarios de ayudas, además de realizar ciertos comandos previos en lenguaje *MATLAB*, necesarios para dicho bloque.

Construcción de un modelo simple

Por razones didácticas se presenta un ejemplo de la creación de un modelo básico para simulación, más adelante se tendrán aclaraciones específicas de las herramientas empleadas según su necesidad.

El modelo a construir consta de un generador de señal senoidal y del cálculo de su integral, el diagrama a crearse se verá así:

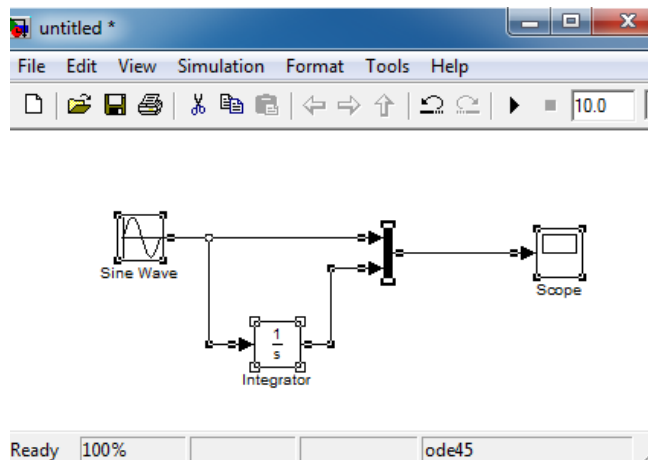



Figura 11: Diagrama de bloques para simular una señal senoidal y su integral.

Para acceder a SIMULINK y sus librerías de bloques se escribe *simulink* en la ventana de comandos de MATLAB, con lo cual se obtiene el siguiente resultado. Otra forma es dar *click* en el ícono , o dar click en Start (botón de la parte inferior izquierda de MATLAB) > Simulink > Library Browser.

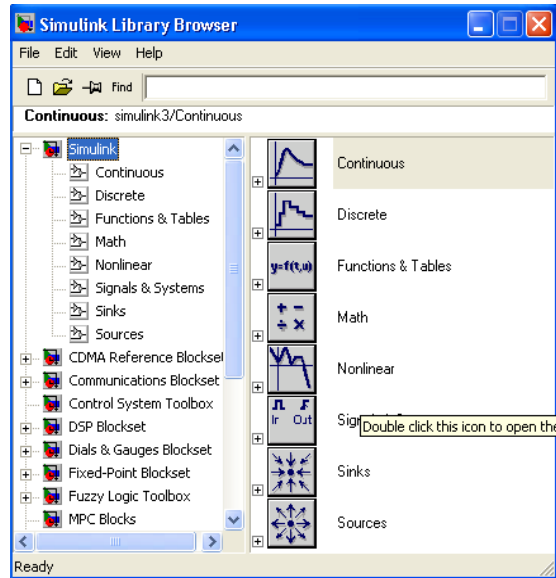


Figura 12: Librerías del SIMULINK

Del menú *File* de esta ventana se selecciona *new*, para crear un nuevo modelo. Los bloques a utilizarse se obtienen de esta ventana de la siguiente forma:

- El bloque Sine Wave block de la librería *Sources*.
- El bloque Scope de la librería *Sinks*.
- El bloque Integrator de la librería *Continuous*.
- El bloque Mux de la librería *Signals and Systems*.

Al abrir la librería *Sources* se tiene acceso al bloque seno, para esto se debe ejecutar un doble *click* (el presionar el botón activo izquierdo del ratón, se suele denominar *click*) en el ícono (o gráfico del bloque) respectivo. Todos los bloques en esta librería son generadores de señal, es así como se ve en la Figura 13:

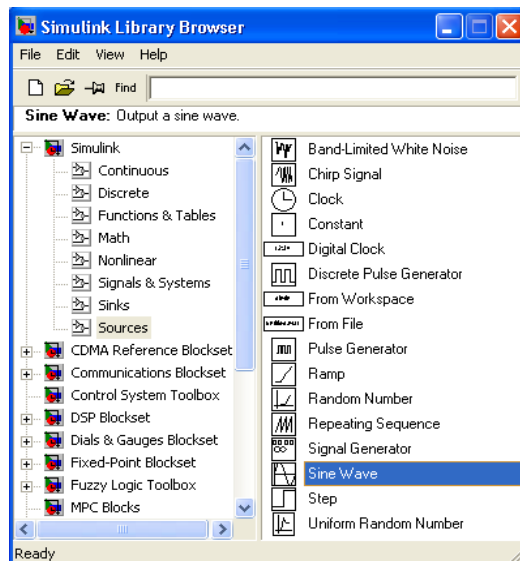


Figura 13: Librería *Sources* o fuentes

También se pueden ubicar los bloques utilizando la barra de búsqueda

Se añaden bloques al modelo nuevo dando *click* derecho y seleccionando “Add to *” donde * es el nombre del modelo, otra forma es arrastrar el bloque con el raton hasta el espacio de trabajo del modelo, manteniendo presionado el botón click izquierdo. Para este ejemplo se copia el bloque del generador de onda senoidal:

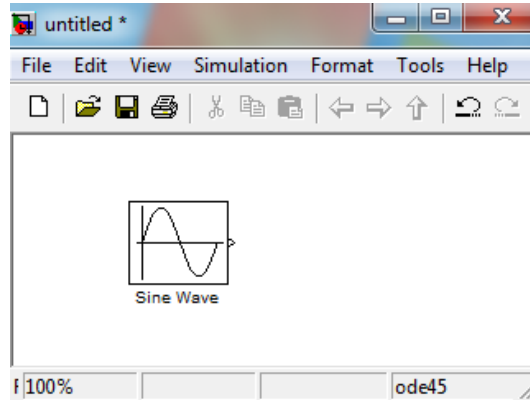


Figura 14: Modelo a simular denominado por defecto en un inicio *untitled* (sin título).

Una vez copiado el bloque, se puede volverlo a copiar presionando CTRL+click en el bloque y arrastrándolo en la misma ventana, además es posible definir el tamaño del bloque y cambiar su nombre, de esta manera se copian todos los bloques requeridos para el ejemplo, como se puede apreciar en la Figura 15.

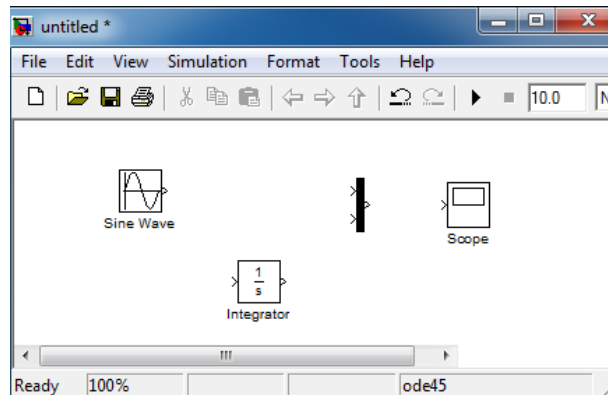


Figura 15: Modelo a simular con los bloques copiados y sin conexiones.

En cada bloque se tiene un tipo de señalización, cuando se tiene el símbolo “>” y a la izquierda, el bloque entonces es un puerto de entrada de una señal; de lo contrario es una salida del bloque. Entonces se tienen puertos de entrada y salida como se observan en la Figura 16.

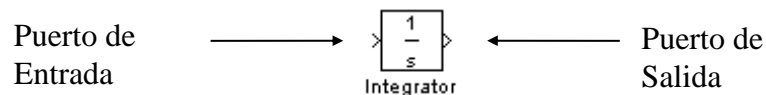


Figura 16: Puertos de entrada y salida de un bloque

Como se puede observar en la Figura 15 el bloque *mux* debe tener 2 entradas, con dos "clicks" de ratón en el bloque se muestra un recuadro de dialogo (Figura 17), entonces se coloca el valor de 2 en el número de puertos de entrada.

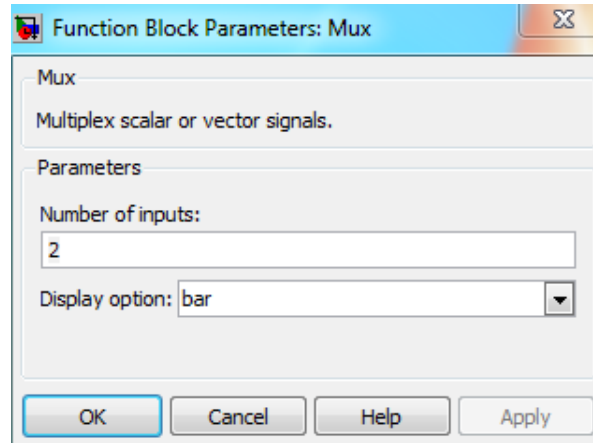


Figura 17: Recuadro de dialogo del bloque *Mux* con sus propiedades

Manteniendo presionado el botón izquierdo del ratón se puede conectar puntos desde un puerto de salida a un puerto de entrada; para realizar conexiones desde una línea, se presiona adicionalmente la tecla *CTRL*.

Antes de la simulación se debe inicializar los parámetros seleccionando del menú *Simulation* la opción *Configuration Parameters* (Figura 18), aquí se indican los valores de tiempo de simulación, tipo de simulación y características, si se desea que los datos se envíen al campo de datos del *MATLAB* y otras opciones más. Posteriormente a esto se inicia la simulación seleccionando del menú *Simulation* la palabra *Start* (*CTRL+T*) (Figura 19) o dando click en el ícono ▶.

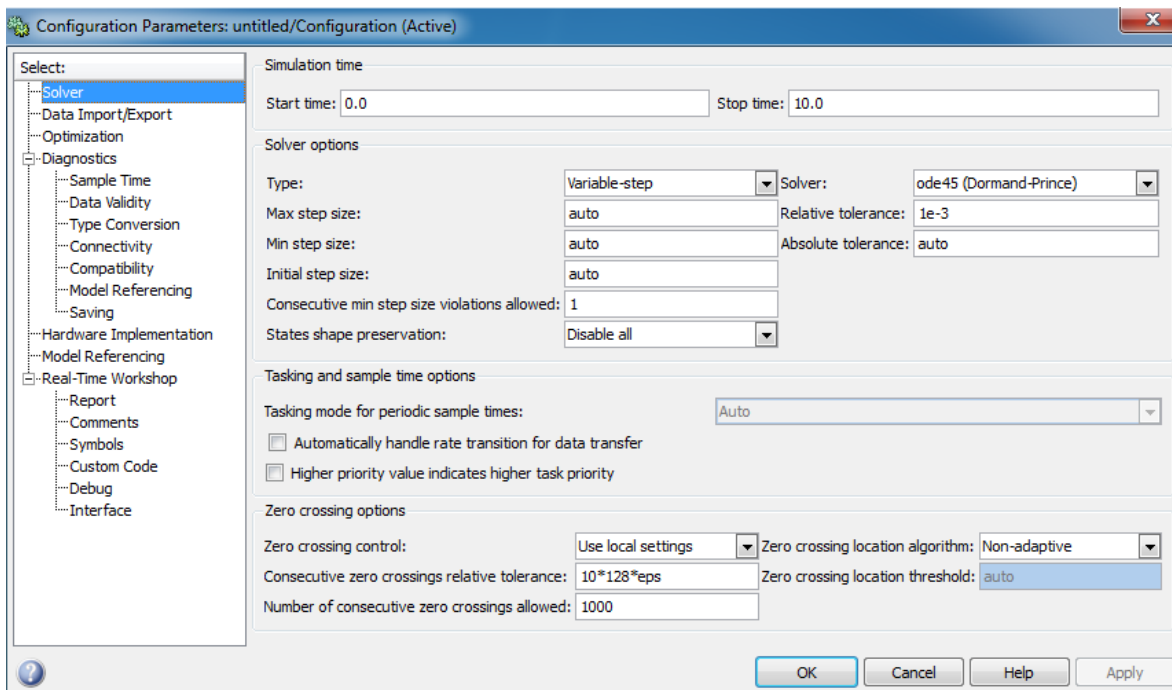


Figura 18: Parámetros de la Simulación.

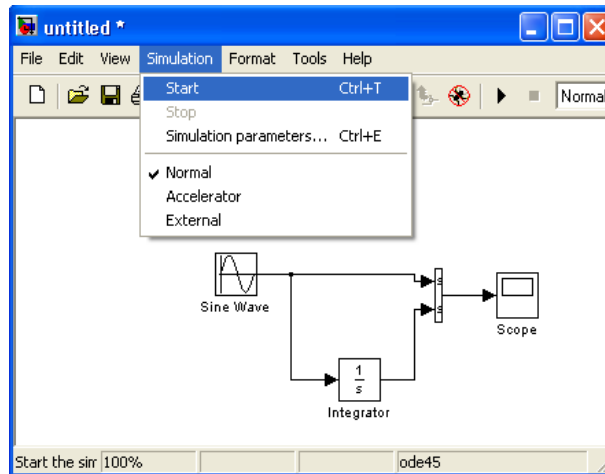


Figura 19: Menú de selección para iniciar la simulación, detener temporalmente (*pause*) o definir parámetros.

Los resultados se presentan en la Figura 20 después de hacer dos "*clicks*" en el bloque *Scope*:

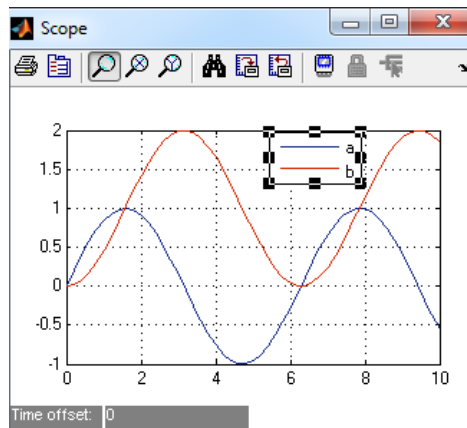








Figura 20: Resultado gráfico de la simulación del ejemplo, a es la integral de la señal senoidal y b es la señal senoidal

De esta manera se ha indicado brevemente como generar un modelo sencillo. Es necesario explicar las funciones de esta pantalla o bloque *Scope*, ya que sirve para analizar los resultados, lo cual es muy importante para la medición de resultados.

-  Esta opción selecciona una región del gráfico, la cual será ampliada hasta el tamaño de la ventana *Scope*.
-  Esta ayuda realiza una ampliación del gráfico, pero solo en el eje horizontal.
-  Este botón realiza una ampliación del gráfico en el eje vertical.
-  Define los límites de los ejes con el valor máximo y mínimo de las señales a representarse gráficamente.
-  Una vez visualizada la imagen requerida se define los valores de escalas actuales por defecto para las siguientes simulaciones.
-  Permite ajustar los máximos y mínimos de las escalas de acuerdo a valores escogidos por el usuario de manera manual. Permite al usuario definir la cantidad de puntos a presentarse en el bloque de simulación y si el bloque *scope* puede funcionar como bloque flotante. Cuando se toma esta opción se tiene la plantilla de datos representada en la Figura 21.

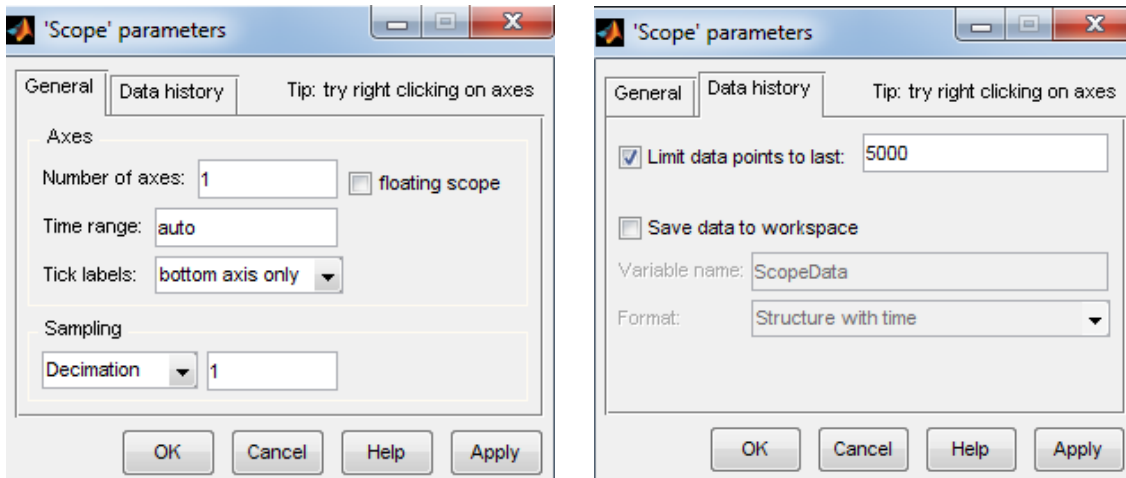


Figura 21: Plantilla de datos del bloque *Scope*: opciones *General* y *Data history*

En la Figura 21 se presenta la página *General*. En la opción *Axes* se tiene campos para ingresar el valor máximo (*Y max.:*) y el mínimo (*Y min.:*) para la variable dependiente. El valor ingresado en *Time range* puede ser un número o dejar la opción *auto*; cuando se tiene esta opción el tiempo es aquel ingresado en la parte *Simulation > Configuration Parameters*. También se presentan dos campos. El uno es un menú desplegable con dos opciones y el otro es una ventana que permite recibir datos numéricos. Las dos opciones del menú son: *Decimation* y *Sample time*. Si se elige la opción *Decimation* y se ingresa en la ventana de datos el valor de 1 (valor por defecto), cada punto que ingresa al bloque *Scope* es dibujado; si se ingresa el valor de 2, cada dos puntos son dibujados y así sucesivamente. Si se elige la opción *Sample time*, se debe ingresar en la ventana de datos el espacio absoluto entre puntos a dibujarse.

Finalmente se tiene la opción de que el bloque *scope* funcione como un osciloscopio flotante o fijo en la elección *Floating scope*. Cuando se elige esta opción se debe indicar, la línea de interconexión entre los bloques del modelo que se desea ver la señal; y, en caso de no tomar esta opción se necesita conectar una línea desde la parte de interés a la entrada del bloque *scope*.

La página *Data history* (Figura 21) permite controlar el número de puntos mostrados y guardar los datos (no el gráfico) a la memoria general del *MATLAB* o denominada también *workspace*.

El bloque *Scope* almacena los datos que representan gráficamente un vector cuyo límite está dado por el valor ingresado en la opción *Limit data points to last*. Estos datos almacenados son los que más tarde ayudan a la manipulación del gráfico, como son ampliaciones o búsquedas de coordenadas de un punto dibujado. Luego en esta misma sección se tiene la posibilidad de grabar en el *workspace* los datos de este bloque, para utilizarse dentro del *MATLAB*.

Subsistemas

Los subsistemas son modelos sencillos agrupados con el propósito de simplificar la visión conceptual de un modelo a simularse, ésta es una manera de generar nuevos bloques de simulación a partir de otros ya existentes. La forma de agrupación de los bloques para formar un subsistema debe tener en cuenta una relación funcional entre los bloques seleccionados, es decir que en conjunto cumplen con una determinada tarea. Por ejemplo en la Figura 22 se indica cómo conformar un grupo de bloques que sirvan para evaluar la ecuación $y = mx+b$.

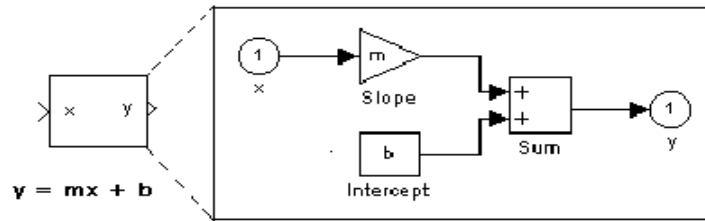


Figura 22: Ejemplo de Subsistema que calcula los resultados de la ecuación $y = mx+b$

Después de desarrollar un modelo de simulación como se vio anteriormente (es decir buscando en las librerías respectivas los bloques y realizando las conexiones necesarias) en este modelo se selecciona con ayuda del ratón los bloques del recuadro, seguido de seleccionar la opción *Create Subsystem* del menú *Edit*, creándose el bloque que se observa a la izquierda en la Figura 22. El subsistema tiene como entrada la señal x y como salida la señal y de la ecuación $y = mx+b$.

3. TRABAJO PREPARATORIO

3.1. Leer y **entender** el marco teórico expuesto en las hojas guías.

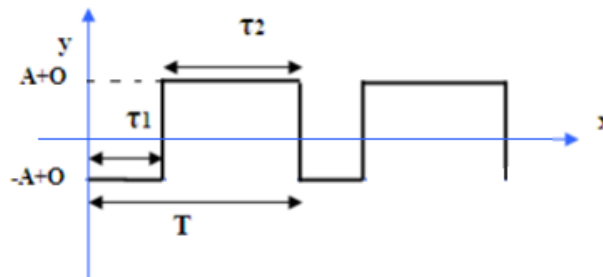
4. PARTE PRÁCTICA

4.1. Generar archivos .m que permitan obtener y graficar diferentes funciones en el dominio del tiempo.

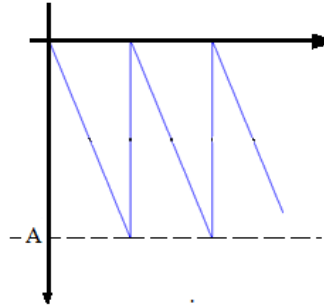
- Generar una onda senoidal de amplitud A y periodo T , con offset $A/2$.
- Generar una onda cuadrada simétrica de amplitud máximo $+A$ y mínimo $-A$, con periodo T .
- Generar una onda diente de sierra de amplitud máximo $+A/2$ y mínimo $-A/2$, con periodo T .
- Genere la siguiente señal: $-3+2*\cos(2*\pi*t)+\sin^2(5*\pi*t)$

• Grupo 1 (GR2): Lunes 16-18

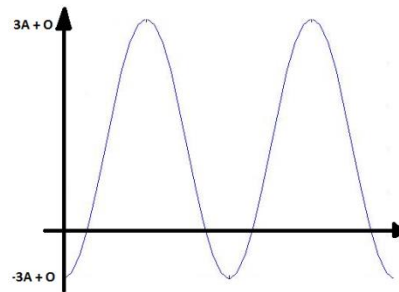
- Generar una onda cuadrada con tiempo de duración en alto de τ_2 , período T y valor de offset O .



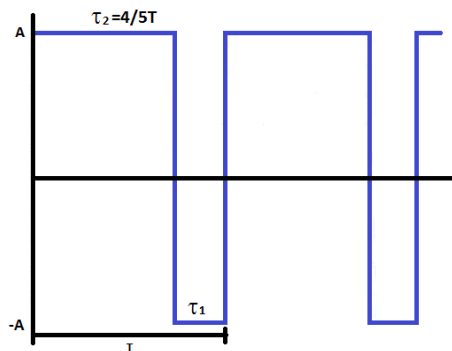
- Generar una onda diente de sierra de amplitud $-A$ y periodo T .



- Grafique las 2 funciones en un solo gráfico y colores distintos
- Grupo 2 (GR3): Lunes 16-18
 - Generar una onda cosenoidal de amplitud $-3A$, periodo T y offset en amplitud de $2A$.



- Generar una onda cuadrada de amplitud A , periodo T y un $\tau_2=4/5T$.



Nota: Los valores de amplitud “A” y periodo “T” deben ser ingresados por el usuario mediante el teclado.

- 4.2. Utilizando SIMULINK generar los modelos que permitan representar en tiempo las funciones del literal 4.1.
- 4.3. Tomar nota de los resultados obtenidos en el numeral anterior para su respectivo análisis que se deberá incluir en el informe.

5. **INFORME**

- Enviar los .m que se indican en la practica y enviar adjunto un archivo realizando el correspondiente analisis de cada señal obtenida.
- Adicionalmente realizar 3 conclusiones por cada persona del grupo.

6. **BIBLIOGRAFIA**

- http://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf
- http://www.sisoft.ucm.es/Manuales/MATLAB_r2006b.pdf