LABORATORIO DE COMUNICACIÓN DIGITAL

PRÁCTICA N°1

1. TEMA

INTRODUCCIÓN A MATLAB Y SIMULINK

2. OBJETIVOS

2.1. Familiarizar al estudiante con el uso de MATLAB.
2.2. Familiarizar al estudiante con el paquete de SIMULINK.
2.3. Aplicar MATLAB y SIMULINK para analizar las señales en el dominio del tiempo.

3. MARCO TEÓRICO

Matlab es un software interactivo para computación numérica y visualización de datos, basado en el uso de matrices (de ahí su nombre que es una abreviación de "Matrix Laboratory"). Éste integra cálculos, programación y visualización en una herramienta flexible, amigable y 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 la Figura 1, donde se representa de forma gráfica una función de dos variables en tres dimensiones, de la misma manera que se pueden representar funciones, datos y realizar mediciones de magnitudes físicas.

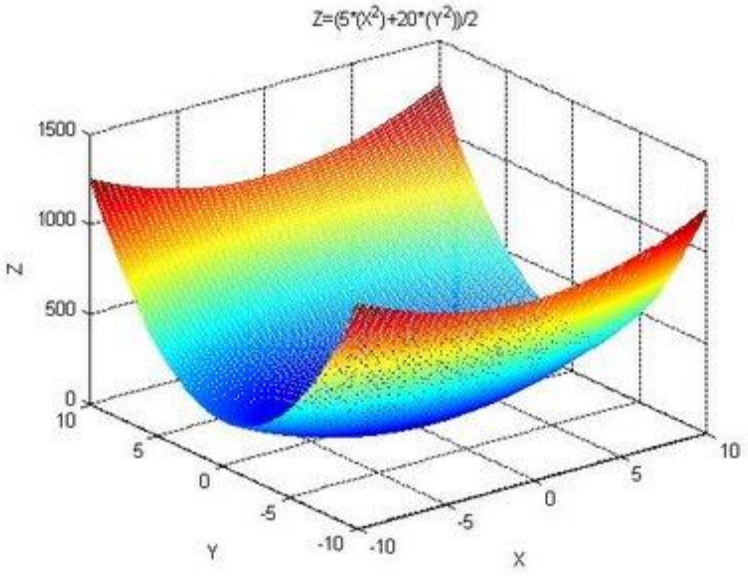

Figura 1: Representación en tres dimensiones de la función $Z= (5x^2+ 20y^2)/2$

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

3.1.1 INTRODUCCIÓN AL USO DE MATLAB

El entorno de trabajo de MATLAB se puede personalizar, por lo que las secciones disponibles pueden variar de usuario a usuario o de versión a versión, pero podemos notar las siguientes secciones particularmente útiles ejemplificadas en la Figura 2:

- **Workspace (1):** contiene todas las variables, vectores, matrices, estructuras u otros objetos existentes en la sesión. El *workspace* es volátil, por lo que al reiniciar Matlab, se reseteará. Para evitar pérdida de datos se pueden guardar de forma manual en archivos .mat.
- **Línea de comandos o *prompt* (2):** aquí se pueden correr comandos individuales para creación, manipulación o eliminación de datos, llamar a funciones o acceder a otras funcionalidades de MATLAB.
- **Editor (3):** aquí se puede visualizar y trabajar con archivos .m, los cuales contienen scripts que MATLAB puede interpretar y correr. Se pueden tener varias pestañas para trabajar con varios archivos .m de manera simultánea.
- **Directorio de trabajo (4):** nos indica la carpeta en la cual estamos ubicados actualmente y sus contenidos. Se pueden abrir archivos directamente desde esta sección.
- **Menús (5):** desde aquí podemos ubicar de manera gráfica e intuitiva las varias funcionalidades de MATLAB como opciones de edición, aplicaciones, *toolboxes*, opciones de publicación o vistas.

Para acceder a las funcionalidades extra de MATLAB se lo puede realizar mediante la línea de comando si se conocen los comandos apropiados, o a través de los menús de una manera gráfica e intuitiva.

Como un pequeño ejemplo ilustrativo se genera una señal sinusoidal de una frecuencia de 20 [Hz] durante dos periodos. Para esto es necesario definir una frecuencia de muestreo, debido a que MATLAB trabaja de manera digital.

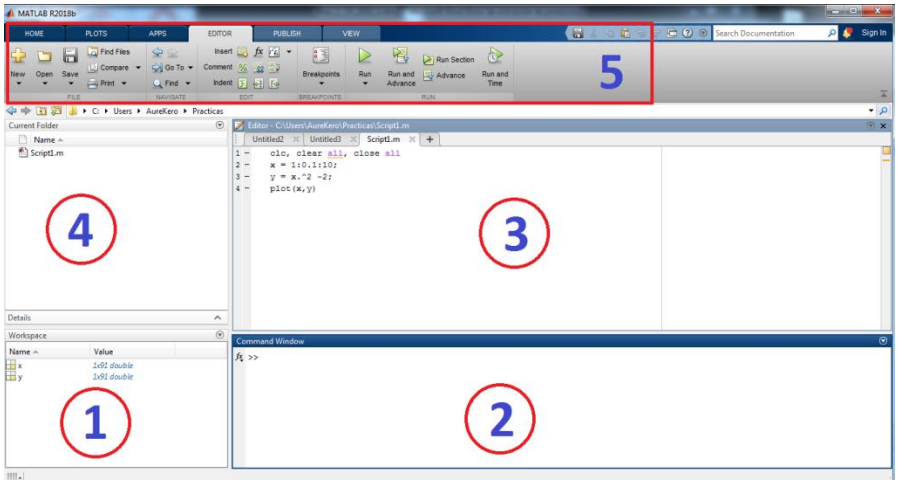

Figura 2: Entorno de trabajo de MATLAB.

Con el código de la Figura 3 se puede generar el ejemplo descrito, una señal de 20 [Hz] con una duración de dos periodos y una amplitud definida de 10 (por ejemplo: voltios, amperios, etc.).

```
clc, clear all, close all        % Se limpia la línea de comandos, el workspace y se cierran todas las ventanas

A = 10;              % Definimos la amplitud de la señal
f = 20;              % Definimos la frecuencia de la señal
t = 0:0.01/f:2/f;    % Definimos el dominio de la señal en tiempo
                     % Los pasos dependen de la frecuencia para que se tenga una curva de apariencia suave
                     % El rango se define en función de la frecuencia para obtener 2 periodos
x = A*sin(2*pi*f*t);     % Obtenemos la señal deseada
plot(t,x)                % Graficamos la señal deseada en función del tiempo
```

Figura 3: Código de una función sinusoidal

La gráfica obtenida con este script se presenta en la Figura 4.

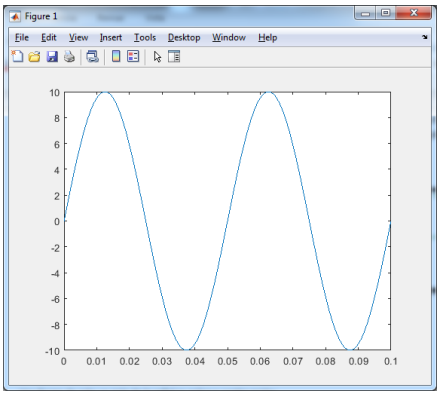

Figura 4: Función senoidal de 20Hz

Como se puede ver en la Figura 3, se define el rango de tiempo como un vector con un punto de inicio, un punto de fin y un ancho de intervalo separados por el carácter de dos puntos (:).

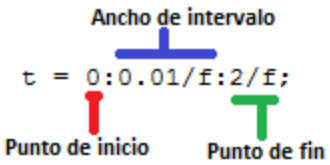

Figura 5: Definición de un vector por intervalos

Utilizar la frecuencia como referencia para el ancho de intervalo nos permite hacer que los intervalos dependan del periodo, y para el punto de fin nos permite definirlo como un múltiplo del periodo.

El uso de scripts en archivos .m nos permite conservar nuestro código, reutilizarlo, modificarlo y tener más flexibilidad sobre nuestras instrucciones.

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.

Se generará un ejemplo de archivo .m para la generación de una señal sinusoidal de 50 Hz por un periodo con una frecuencia de muestreo de 1kHz (Figura 7) y se lo guardará con el nombre “ejemplo.m” (Figura 6).

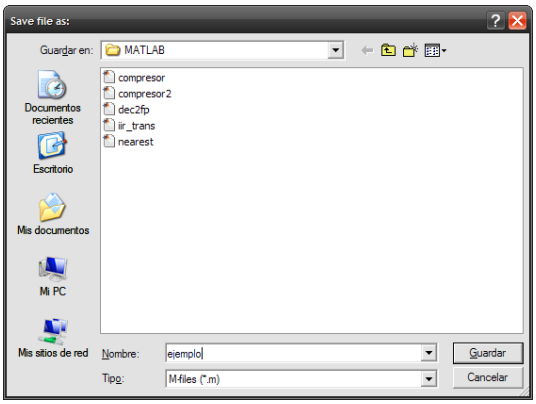

Figura 6: Guardando el archivo .m

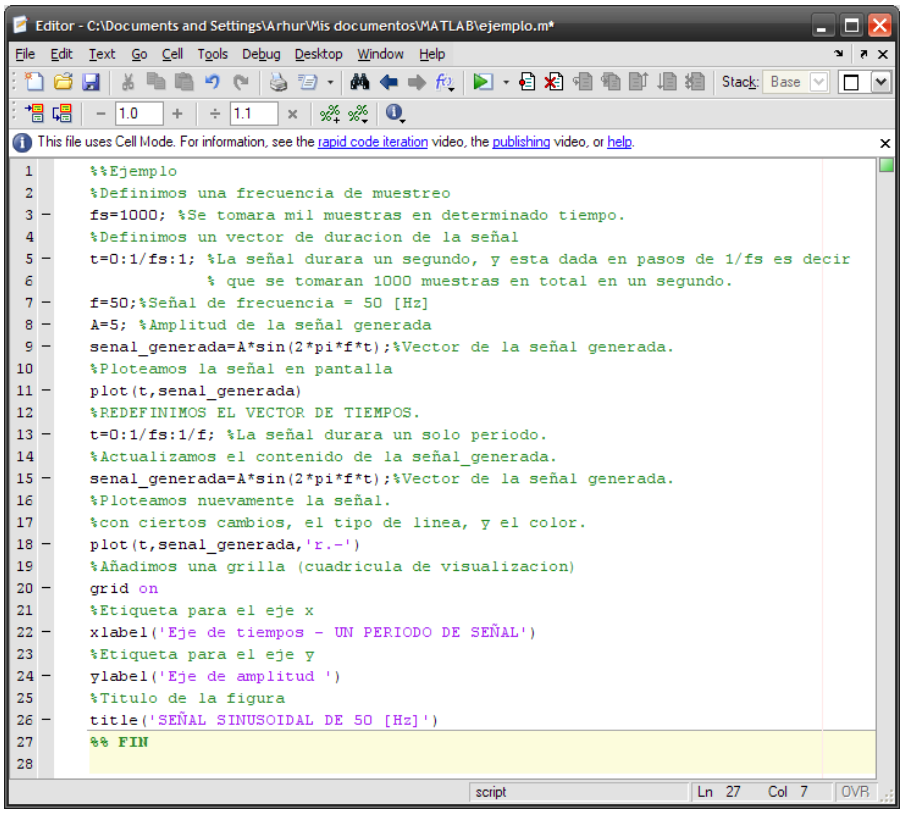

Figura 7: Entorno de trabajo del Editor de archivos de MATLAB con el código de ejemplo.m

Hay tres formas de ejecutar el código generado: (1) con la tecla F5, run [archivo.m], (2) dando *click* en el siguiente botón de la barra de herramientas. y (3) escribiendo el nombre del archivo en la línea de comandos.

3.2 SIMULINK

3.2.1 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.

<u>Sistemas de Modelación a través de diagramas de bloques</u>

Para representar sistemas que se desean analizar se ha desarrollado una técnica ilustrativa en los textos técnicos denominada 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 de *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 diferenciales.

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 una interfaz denominada *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:

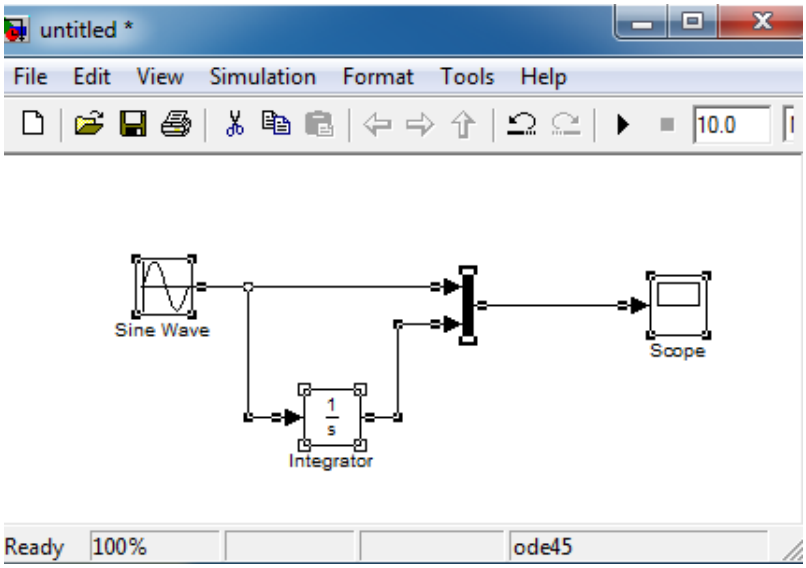

Figura 8: Diagrama de bloques para simular una señal senoidal y su integral.

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*.

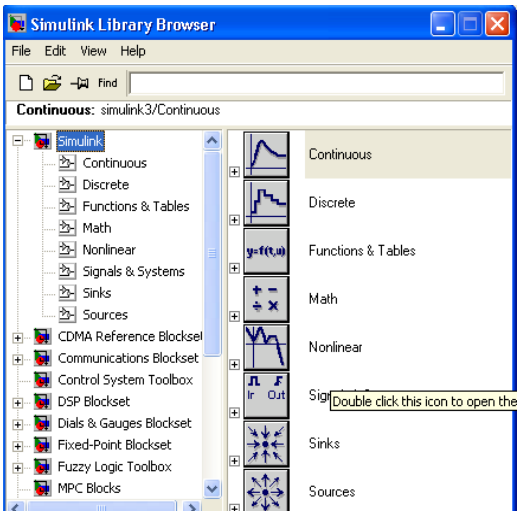

Figura 9: Librerías del SIMULINK

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 *Sine Wave*, 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 10.

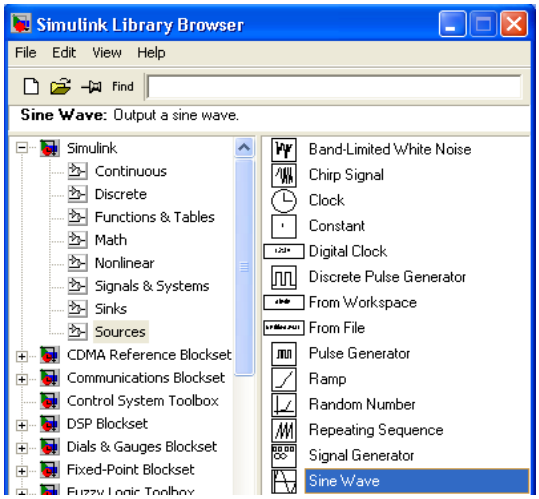

Figura 10: Librería Sources o fuentes

También se pueden ubicar los bloques utilizando la barra de búsqueda.

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

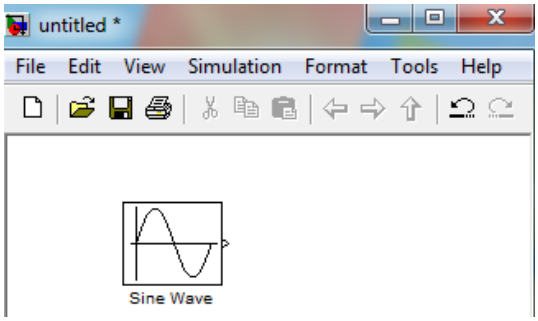

Figura 11: Modelo a simular denominado por defecto en un inicio untitled (sin título).

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 12.

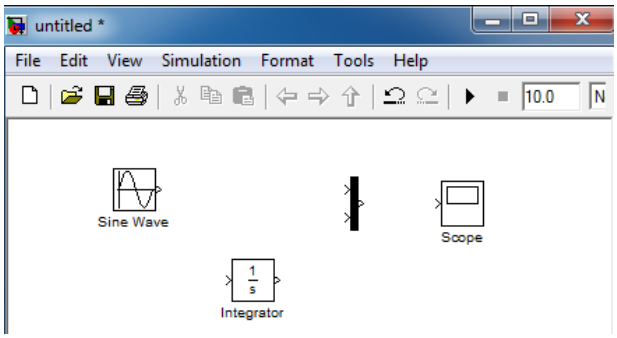

Figura 12: Modelo a simular con los bloques copiados y sin conexiones.

En cada bloque se tiene un tipo de señalización, cuando se tiene el símbolo " > " a la izquierda del bloque, entonces simboliza 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 13.

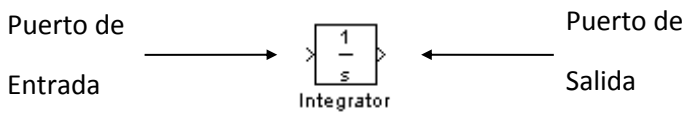

Figura 2: Puertos de entrada y salida de un bloque

Como se puede observar en la Figura 12 el bloque *mux* debe tener 2 entradas, con dos *clicks* de ratón en el bloque se muestra un recuadro de dialogo (Figura 14), entonces se coloca el valor de 2 en el número de puertos de entrada.

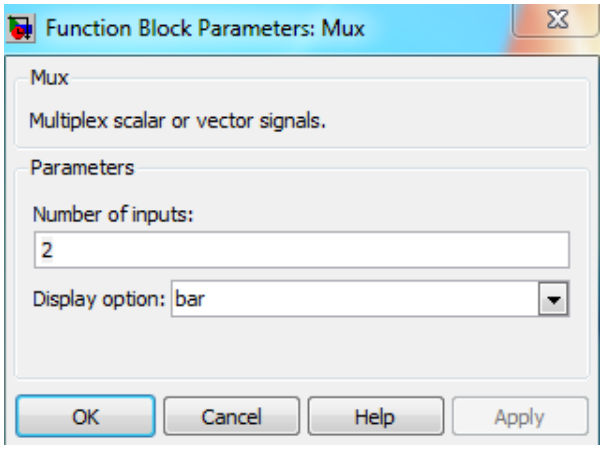

Figura 14: Recuadro de dialogo del bloque Mux con sus propiedades

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 15), 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 de

MATLAB y otras opciones más. Posteriormente a esto se inicia la simulación seleccionando del menú *Simulation* la palabra *Start (CTRL+T)* (Figura 16) o dando *click* en el ícono ▶.

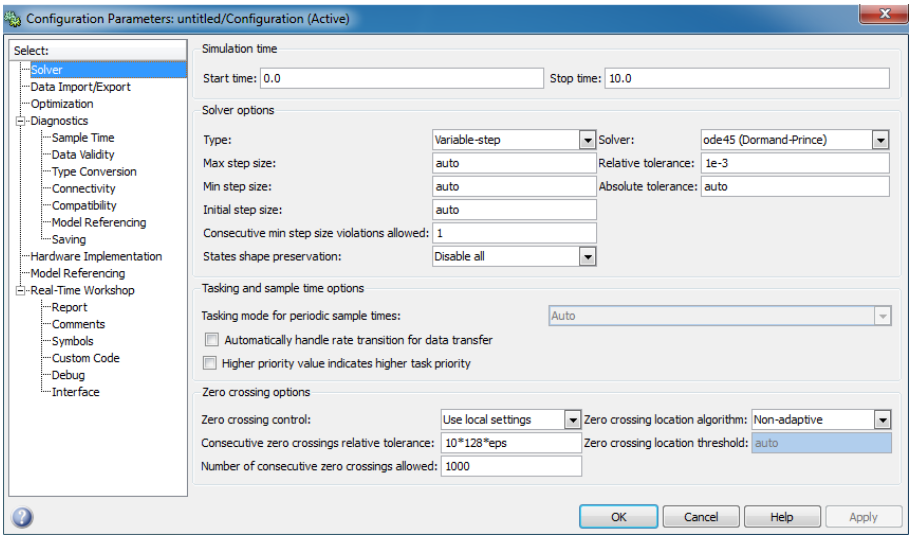

Figura 15: Parámetros de la Simulación.

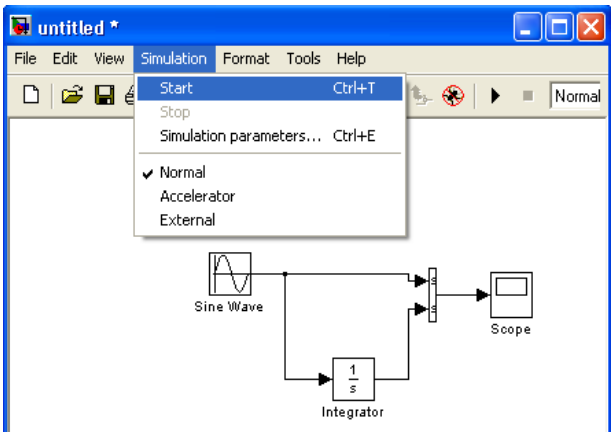

Figura 16: Menú de selección para iniciar la simulación, detener temporalmente (pause) o definir parámetros.

Los resultados se presentan en la Figura 17 después de hacer dos *clicks* en el bloque *Scope*:

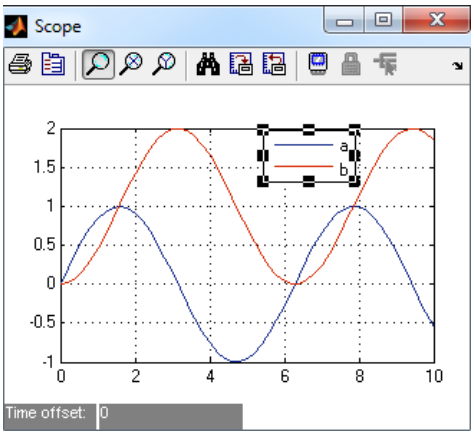

Figura 17: Resultado gráfico de la simulación del ejemplo, 'a' es la integral de la señal senoidal y 'b' es la señal senoidal

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 y hacer mediciones.

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 18.

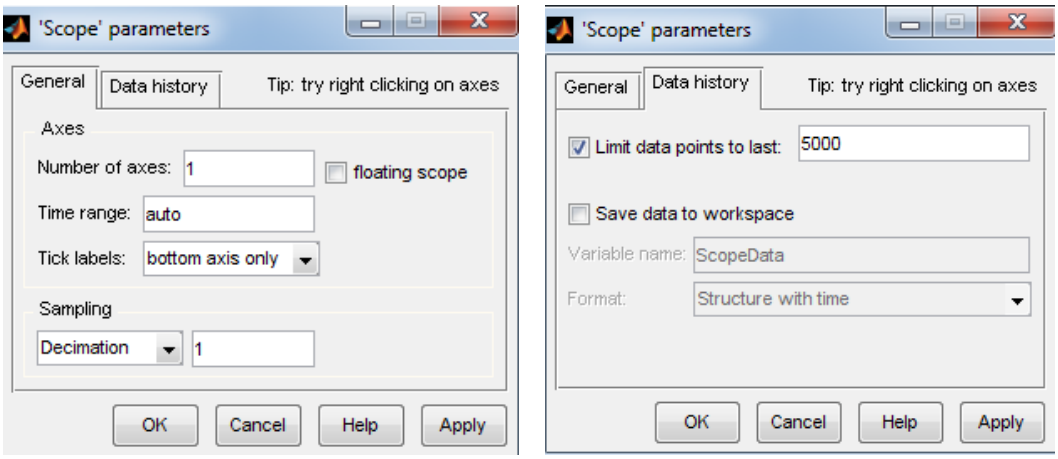

Figura 18: Plantilla de datos del bloque Scope: opciones General y Data history

En la Figura 18 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 especificar 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 18) 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 19 se indica cómo conformar un grupo de bloques que sirvan para evaluar la ecuación y = mx+b.

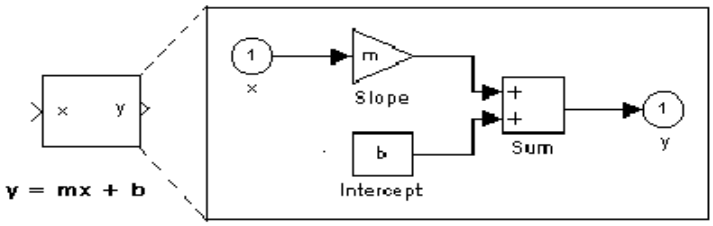

Figura 19: Ejemplo de Subsistema que calcula los resultados de la ecuación y = mx+b

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 19. El subsistema tiene como entrada la señal *x* y como salida la señal *y* de la ecuación *y = mx+b*.

4. TRABAJO PREPARATORIO

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

5. EQUIPO Y MATERIALES

- Computadora
  - Software MATLAB

6. PROCEDIMIENTO

6.1. Con ayuda del instructor generar archivos .m que permitan obtener y graficar diferentes funciones en el dominio del tiempo, en un intervalo de tiempo razonable.

- Generar una onda senoidal de amplitud A, periodo T y offset A/2.
- Generar una onda cuadrada simétrica de amplitud máximo +A y mínimo –A, con periodo T.
- Generar una onda cuadrada con un *duty* del 30% de amplitud máximo +A y mínimo –A/2, con periodo T.
- Generar una onda diente de sierra de amplitud máximo +A/4 y mínimo –A/4, con periodo T.
- Genere la siguiente señal: $10 - \sin^2(2*pi*f1*t) + \cos(6*pi*f2*t) + 3t/(1+t^2)$, donde f1 = 5KHz y f2=10KHz.

6.2 Crear los archivos .m que permitan obtener y graficar las funciones en el dominio del tiempo que indique el instructor.

6.3 Utilizando SIMULINK generar los modelos que permitan representar en tiempo las funciones de los literales 6.1 y 6.2.

6.4 Tomar nota de los resultados obtenidos en el numeral anterior para su respectivo análisis que se deberá incluir en el informe.

7. INFORME

7.1 Consultar sobre las instrucciones que se utilizan para las gráficas 2D en MATLAB. Obtener las gráficas de las señales de la práctica utilizando dichos comandos y la correcta sintaxis en dos escalas distintas, con todos los elementos de la gráfica (etiquetas, nombres de ejes, título).

7.2 Graficar la función $x^2/\cos^2$ (2*pi*x) en el intervalo [-pi,pi] en líneas entrecortadas de color rojo. Adjuntar código y gráfica.

7.3 Realizar un archivo .m que permita obtener la gráfica en tiempo de la función *tangente hiperbólica (utilizando exponenciales)*. Adjuntar código y gráficas.

7.4 Realizar un archivo .m que permita graficar la siguiente función en tiempo:

- GR 1: $3f^4(t) - g^2(t)$
- GR 2: $-3f^2(t) + f(t) - 1/g(t)$

Donde f(t) es la función de una onda senoidal de amplitud A/2 y periodo T/3 y g(t) es la función diente de sierra con periodo T/2. El archivo debe permitir ingresar los valores de amplitud y periodo.

El grafico resultante debe presentarse con nombre de gráfico y nombre de los ejes.

7.5 Presentar los resultados obtenidos en la práctica, los códigos de los programas debidamente comentados.

7.6 Conclusiones y Recomendaciones.

7.7 Bibliografía.

8. REFERENCIAS

[1] MathWorks, «MathWorks,» 2017. [En línea]. Disponible: https://www.mathworks.com/help/comm/blocklist.html. [Último acceso: 10 enero 2018].

[2] «MatLab Tutorial» 2017. [En línea]. Disponible: https://ocw.mit.edu/resources/res-18-002-introduction-to-matlab-spring-2008/other-matlab-resources-at-mit/matlab_t.pdf. [Último acceso: 06 abril 2018].