

## Práctica 7. Introducción a Simulink

**Objetivo.** Realizar operaciones y simulaciones sencillas mediante la herramienta gráfica de simulación de Matlab llamada Simulink.

### Introducción.

MATLAB dispone de un entorno gráfico para el desarrollo que puede realizar las mismas actividades que se realizan en el área de trabajo basada en comandos que se describió en las prácticas anteriores. Este entorno gráfico llamado Simulink está basado en diagramas de bloques y está orientado a realizar simulación de sistemas dinámicos en el dominio del tiempo.

El uso de Simulink está orientado a que el usuario defina sistemas a simular mediante diagramas de bloques de manera que el usuario pueda interactuar con los bloques, los cuales representan subsistemas. Los bloques se pueden situar donde se requiera, modificar su tamaño, etiquetarlos, especificar sus parámetros, e interconectarlos para completar sistemas a partir de los cuales se pueden ejecutar simulaciones.

Las características básicas de Simulink son: Simulación de sistemas dinámicos, entorno gráfico, el sistema se representa como una interconexión de bloques elementales, cada bloque lleva asociado un modelo matemático que representa su relación entrada/salida.

### Iniciando Simulink

Para iniciar **Simulink**, se puede proceder de diferentes formas, la más sencilla es teclear desde la ventana de comandos de Matlab:

```
>> simulink
```

Otra manera de posible acceder al programa es mediante el botón de **Simulink** de la barra de botones de Matlab:



Figura 7.1.- Botón de Simulink de la barra de herramientas de Matlab

Una vez que inicia Simulink, aparece una ventana como la mostrada en la figura 7.2 , que corresponde al Navegador de librerías o **Library browser** de Simulink. En esta ventana se muestran todas las librerías de bloques disponibles; cada una de ellas contiene bloques elementales con los que se pueden construir sistemas. Dependiendo de la versión y la instalación concreta de Simulink, el número de librerías puede ser menor o puede ser mayor.

Desde el navegador de librerías se pueden abrir archivos de **Simulink** creados anteriormente, o bien, crear archivos nuevos mediante las opciones marcadas en la figura 7.2 como 'Archivo' y 'Nuevo modelo'. Cada uno de los bloques disponibles para construir modelos, poseen un nombre para poder ser buscados de forma más rápida a través de la casilla marcada en la figura 7.2 como 'Búsqueda de bloques', donde escribiendo el nombre y haciendo clic en el ícono que representa unos binoculares se realiza la búsqueda del bloque con el nombre especificado, la búsqueda se realiza dentro de todos los 'toolboxes'. Los resultados de la búsqueda se mostrarán en la sub-ventana marcada en la figura 7.2 como 'Listado bloques'. En la sub-ventana marcada como 'Toolboxes' se encuentran listados todos los toolboxes disponibles para la versión instalada.

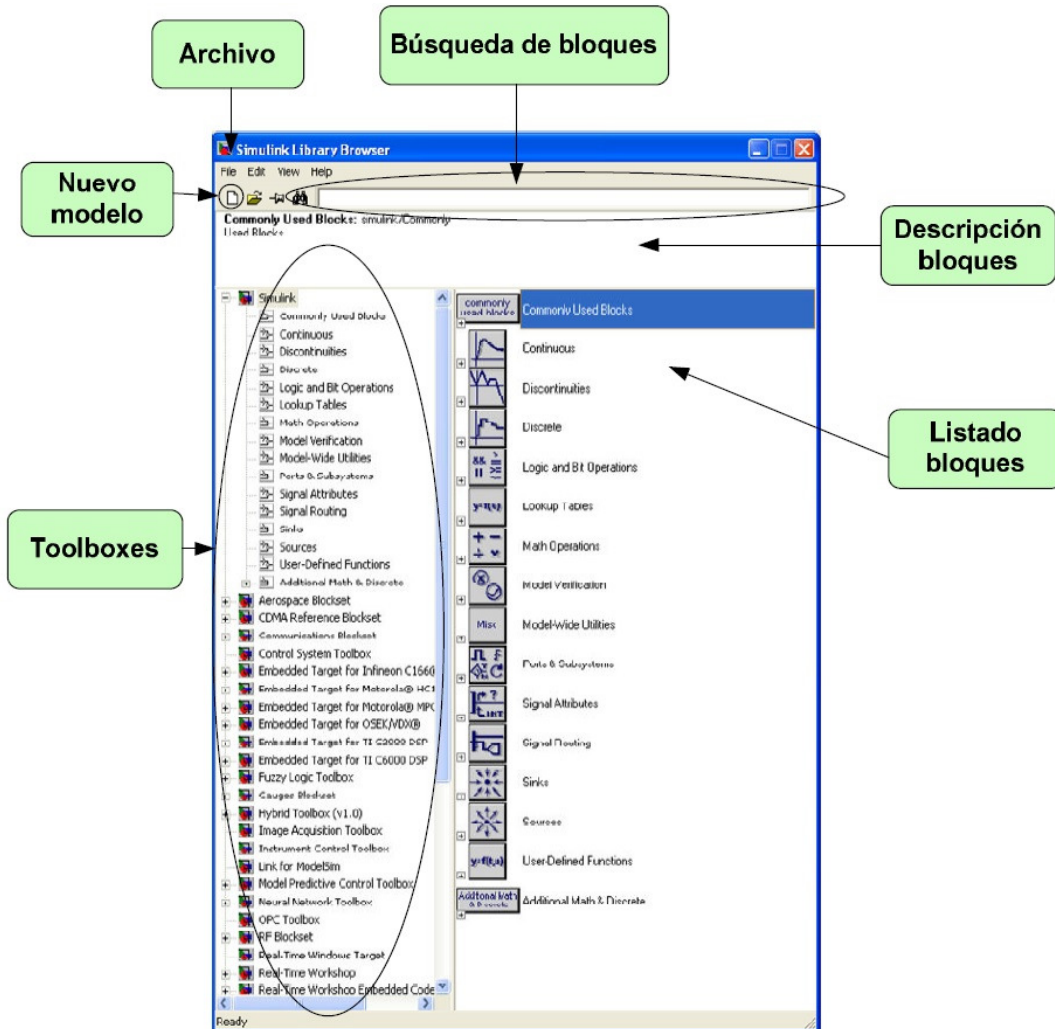


Figura 7.2 Navegador de librerías de simulink.

### Tipos de bloques en Simulink

Como puede observarse, existe multitud de bloques distintos; todos los bloques de Simulink se estructuran en diferentes categorías. La ventana que aparece al iniciar Simulink permite comprobar cuales son las categorías de los bloques existentes. Basta con hacer doble clic sobre el bloque de la categoría deseada, o clic sobre el signo + a la izquierda de dicho bloque de categoría

de la lista, para visualizar los bloques disponibles.

En la figura 7.3 se muestra, en la ventana de la izquierda, el aspecto inicial del navegador y en la ventana de la derecha el resultado de pulsar sobre el signo + de la categoría '**Continuous**'. Los elementos que aparecen a la derecha son los bloques elementales con los que se puede construir un modelo.

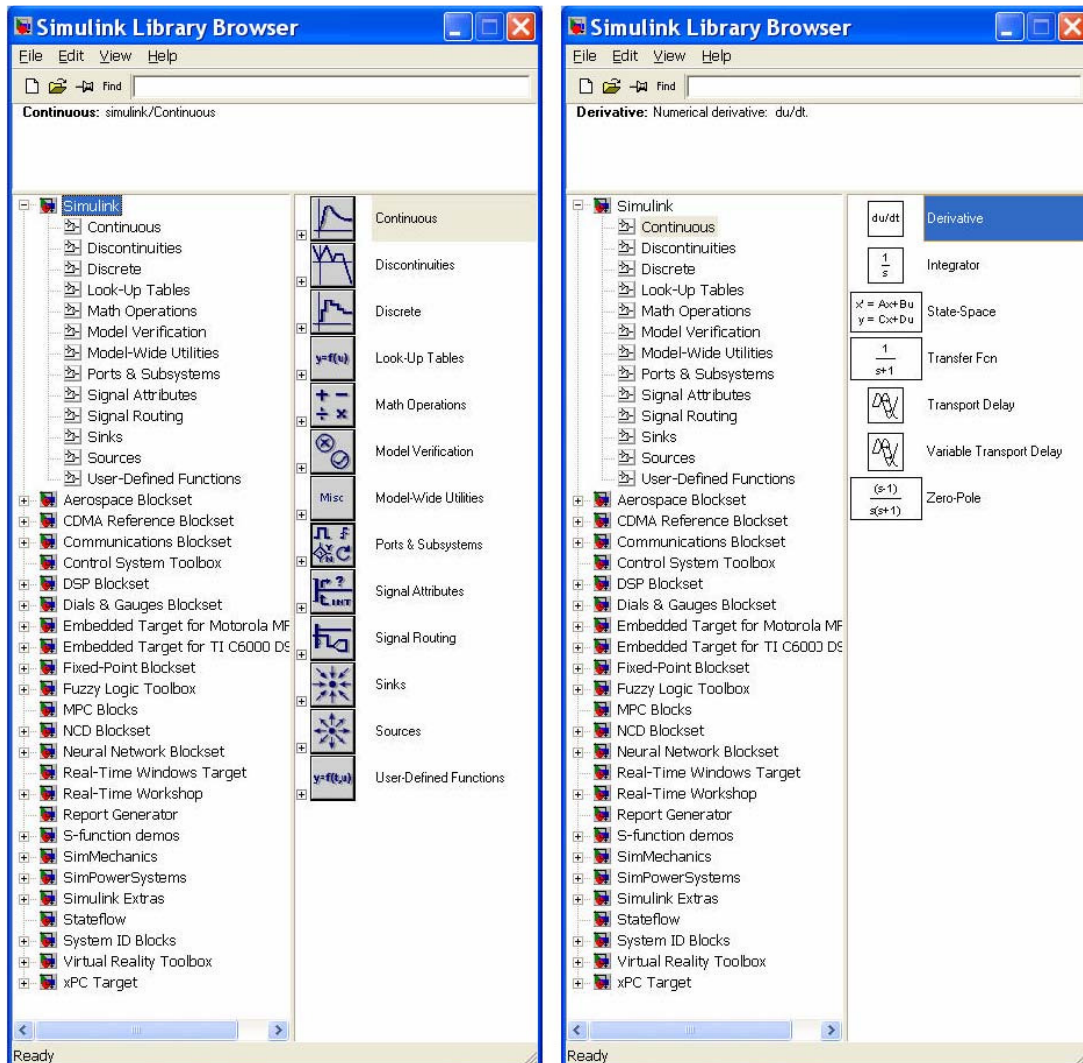


Figura 7.3.- Doble clic sobre la categoría de bloques "**Continuous**".

A continuación se indican algunos de los bloques básicos que se usarán en esta práctica y las siguientes y la categoría a que pertenecen.

**Continuous:** Bloques para introducir sistemas continuos lineales y retardos de transporte

- **Derivative** (bloque derivador: la salida es la derivada de la entrada)
- **Integrator** (bloque integrador: la salida es la integral de la entrada)
- **Transfer Fcn** (Función de transferencia expresada como cociente de polinomios en la variable de Laplace  $s$ )
- **Zero Pole** (Función de transferencia expresada por sus polos, ceros y ganancia)

**Sources:** Entradas o fuentes de señales

- **Step** (escalón)
- **Ramp** (rampa)
- **Sine wave** (sinusoidal)
- **Pulse generator** (tren de pulsos)
- **Random Number** (Señal aleatoria con distribución Gaussiana)
- **Constant** (Valor constante)
- **From workspace** (lectura de datos desde Matlab)

**Sinks:** Salidas o dispositivos de visualización/almacenamiento de variables del sistema

- **Scope** (osciloscopio)
- **Display** (indicador numérico)
- **To workspace** (envío de datos a Matlab)

**Math operators:** Operaciones matemáticas sobre señales

- **Sum** (sumador/restador de señales)
- **Gain** (ganancia o multiplicación de una señal por una constante)

**Signal Routing:** Permite realizar conexiones especiales entre señales

- **Mux** (multiplexor: Agrupa distintas señales en un vector o bus)
- **Demux** (demultiplexor: Desagrupa señales provenientes de un bus)

**User-Defined Functions:** Permiten introducir expresiones algebraicas que procesan la entrada del bloque para calcular la salida.

- **Fcn** (Expresión algebraica genérica: Calcula la salida mediante la expresión definida por el usuario en función de la entrada o entradas al bloque)
- **Matlab Fcn** (Función de Matlab: Calcula la salida mediante una expresión algebraica que contiene funciones estándar de Matlab definida por el usuario y aplicada a la señal de entrada.)

### Creación de un modelo sencillo

El procedimiento para la creación de un modelo consiste en seleccionar los bloques adecuados de entre los presentes en cada categoría, colocarlos sobre la ventana de diseño y establecer las conexiones correspondientes entre ellos. A continuación se ilustra el procedimiento mediante un sistema sencillo.

**Ejemplo.** Crearemos un modelo que nos permita generar y visualizar una señal sinusoidal con frecuencia angular  $\omega=3$  rad/seg, y amplitud 4, es decir, la señal a generar será  $f(t) = 4\sin(3t)$ .

#### Primer paso: creación del modelo.

Desde la ventana '**Simulink Library Browser**', haremos clic sobre el botón marcado como '**Nuevo modelo**' en la figura 7.2. Automáticamente se abrirá una ventana en blanco como la mostrada en la figura 7.4, la cual es la ventana de diseño en donde podemos crear nuestro modelo.

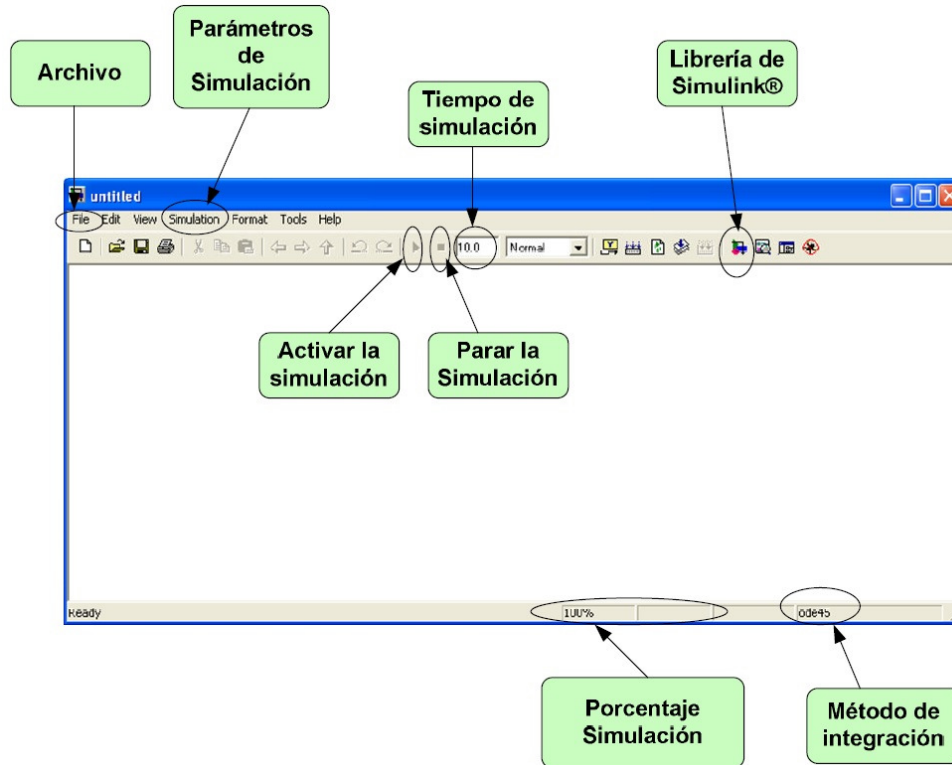


Figura 7.4.- Ventana de diseño para la creación, edición y ejecución de modelos en Simulink.

### **Segundo paso: introducción de bloques en el modelo.**

Los elementos se introducen arrastrándolos con el ratón desde la ventana '*Simulink Library Browser*' que contiene el listado de los bloques hasta la ventana de diseño. En nuestro caso requeriremos los siguientes bloques:

- Dentro de la categoría '**Sources**', el bloque '**Sine Wave**'. Este elemento corresponde a una señal sinusoidal, que es la que deseamos generar.
- Dentro de la categoría '**Sinks**', el bloque '**Scope**'. Será el que utilizemos para visualizar la señal.

Una vez introducidos estos bloques, la ventana de diseño presenta un aspecto como el que se muestra en la Figura 7.5.

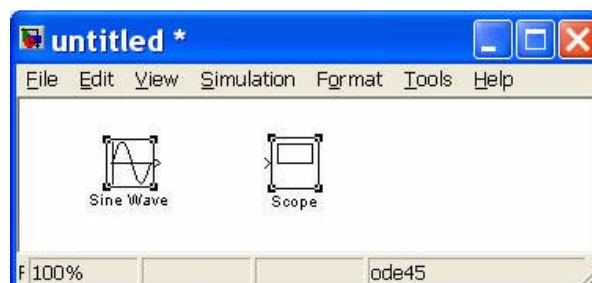


Figura 7.5.- Primeros dos bloques introducidos para el ejemplo.

**Tercer paso: modificación de parámetros de los bloques introducidos.**

Todos los bloques de Simulink permiten una cierta configuración. En particular, del bloque correspondiente a la función sinusoidal se puede configurar su amplitud, frecuencia, fase, etc. Para ello se debe hacer doble clic sobre el bloque, con lo que aparecerá una ventana de introducción de parámetros.

Para nuestra primera simulación, introduciremos los siguientes valores:

- Sine type: time based (buscamos generar una señal continua y no discreta)
- Amplitude: 4 (la señal tomará valores de -4 a +4)
- Bias: 0 (componente de CD = 0)
- Frequency: 3 rad/s (frecuencia angular de la señal sinusoidal)
- Phase: 0 (desfase respecto del instante inicial  $t = 0$ )
- Sample time: 0 (tiempo de muestreo, será cero para señales continuas)

**Cuarto paso: interconexión de elementos.**

Una vez situados los elementos en la ventana de diseño, es necesario establecer conexiones entre ellos. Para conectar 2 elementos debe hacerse un arrastre con el ratón desde la salida de uno de ellos hasta la entrada del elemento al que se quiere conectar. Una vez establecidas las dos conexiones necesarias para nuestro sistema, debemos obtener un resultado como el que se muestra en la Figura 7.6.

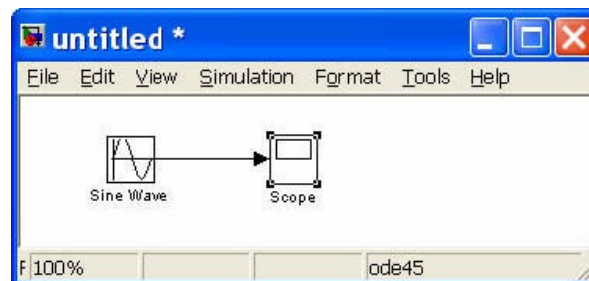


Figura 7.6.- Bloques interconectados del ejemplo.

**Quinto paso: Inicio de la simulación y comprobación de resultados.**

Lanzaremos la simulación con la opción 'Start' del menú 'Simulation' o bien, haciendo clic en el botón triangular marcado como 'Activar simulación' en la figura 7.4 y visualizaremos los resultados obtenidos haciendo doble clic sobre el bloque 'Scope'. Aparecerá una ventana gráfica como la que se muestra en la figura 7.7 sobre la que podremos hacer zoom para ajustar la gráfica.

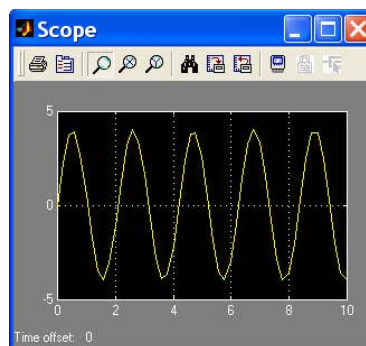


Figura 7.7.- Gráfica visualizada al dar doble clic sobre el bloque 'Scope'.

✎ **Ejercicio.** Una vez creado el esquema, se probará la generación de diferentes señales que nos ofrece Simulink dentro de la categoría '**Sources**'. Para ello borraremos del esquema el bloque '**Sine Wave**' y en su lugar insertaremos otro bloque de esa categoría. probar al menos con los bloques siguientes para probar el efecto de modificar sus parámetros:

- **Step** (escalón)
- **Ramp** (rampa)
- **Pulse generator** (tren de pulsos)
- **Random Number** (Señal aleatoria con distribución Gaussiana)

### Creación de un segundo modelo.

A continuación crearemos un modelo algo más complicado, en el que se realice alguna operación con las señales. Se va a crear un sistema para calcular la integral de una señal. El bloque que realiza la integral de una señal se encuentra dentro de la categoría '**Continuous**'. Es el bloque denominado '**Integrator**'. Con el bloque integrador y con el generador de pulsos en lugar de la señal sinusoidal, se creará un diagrama como el que se muestra en la Figura 7.8.

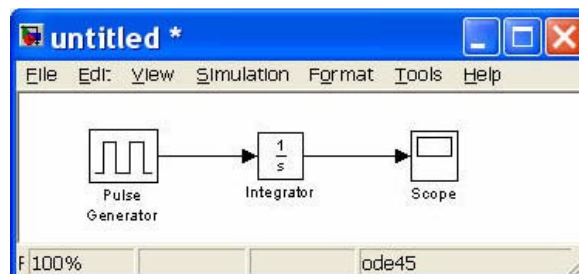


Figura 7.8.- Diagrama de simulación para calcular la integral de una señal pulsante.

A continuación se añade un cambio en el esquema que nos permita ver en un mismo gráfico las dos señales: la original y su integral. Para visualizar dos señales en el elemento '**Scope**' es necesario recurrir al bloque multiplexor '**Mux**' de la categoría '**Signals Routing**'. Este bloque tiene por finalidad agrupar dos o más señales para formar un vector. De este modo, se agruparán la señal de entrada y la señal de salida y el conjunto será lo que se envíe al bloque '**Scope**'. Las conexiones se deben realizar como se muestra en la figura 7.9, en la cual hay algo nuevo que aún no sabemos hacer: cómo crear una rama nueva partiendo de un punto intermedio de una conexión ya existente. La forma de crear este tipo de conexiones es pulsando el botón '**Ctrl**' al mismo tiempo que se empieza a arrastrar con el ratón.

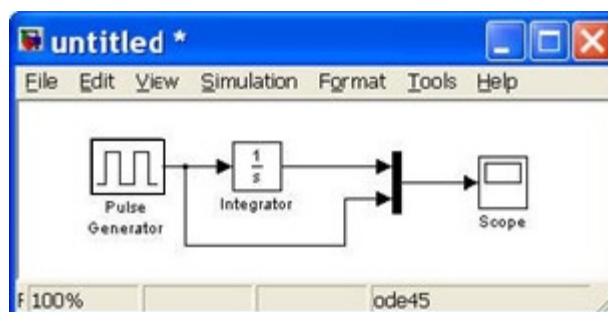


Figura 7.9.- Diagrama de simulación para visualizar dos señales en un solo bloque '**Scope**'

Arrancando la simulación, debemos comprobar que en la ventana del osciloscopio se muestran las dos señales: la original y su integral, tal y como muestra la figura 7.10

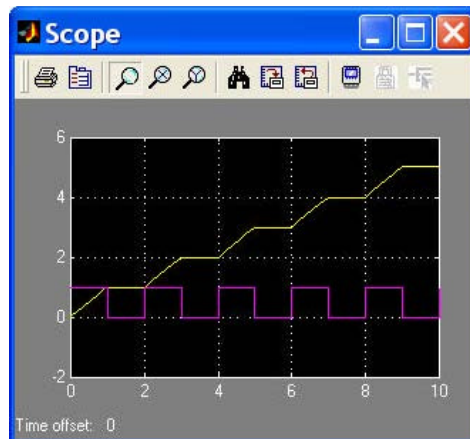


Figura 7.10.- Tren de pulsos y su integral.

### Simulación de un sistema descrito por una ecuación diferencial

Se desea simular un sistema mecánico consistente en una masa **M** sometida a una fuerza externa **f(t)** y unida a un resorte de constante elástica **K**, y con un rozamiento viscoso **B**, como se describe en la figura 7.11

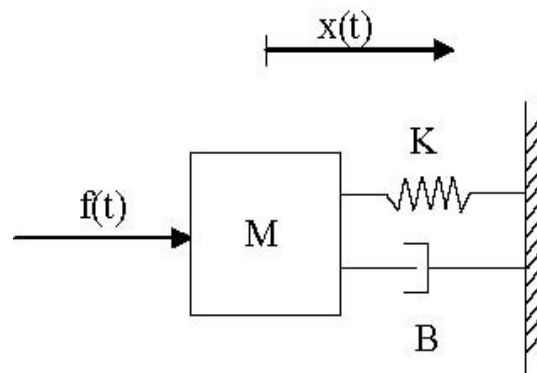


Figura 7.11.- Sistema mecánico masa-resorte-amortiguador.

El objetivo será ver cómo afecta la fuerza aplicada **f(t)** al desplazamiento de la masa, descrito por **x(t)**.

La ecuación diferencial que rige el comportamiento de este sistema se puede obtener mediante la segunda ley de Newton y es el siguiente:

$$M \frac{d^2 x(t)}{dt^2} + B \frac{dx(t)}{dt} + Kx(t) = f(t) \quad (7.1)$$

Para representar este sistema en Simulink, seguiremos el siguiente procedimiento:



1) **Definir la salida del sistema y despejar su máxima derivada.**- En este caso la salida es  $x(t)$  y su máxima derivada es

$$\frac{d^2x(t)}{dt^2} = -\frac{B}{M} \frac{dx(t)}{dt} - \frac{K}{M} x(t) + \frac{1}{M} f(t) \quad (7.2)$$

2) **Crear un diagrama de simulación con tantos integradores en cascada como el orden del sistema** en donde la salida del sistema es la salida del último integrador.- En este caso el orden del sistema es 2 y la salida del sistema  $x(t)$  se toma del segundo integrador. Hasta este punto el diagrama de simulación lucirá como en la figura 7.12. Obsérvese que si la salida del segundo integrador es  $x(t)$ , forzosamente su entrada deberá ser  $\frac{dx(t)}{dt}$ , la cual es la salida del primer integrador y por lo tanto su entrada deberá ser  $\frac{d^2x(t)}{dt^2}$

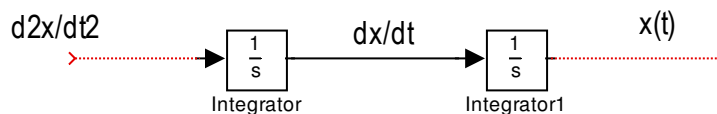


Figura 7.12.- Obtención de  $x(t)$  mediante la doble integración de  $\frac{d^2x(t)}{dt^2}$

3) **Obtener la máxima derivada requerida en el diagrama anterior mediante bloques de simulink** a partir de la ecuación despejada (7.2).- En este caso solo se requieren sumas, restas y multiplicaciones por constante (ganancias constantes). Esto se debe a que el sistema es un SLIT. Con esto completamos el diagrama de simulación para obtener el diagrama de la figura 7.13.

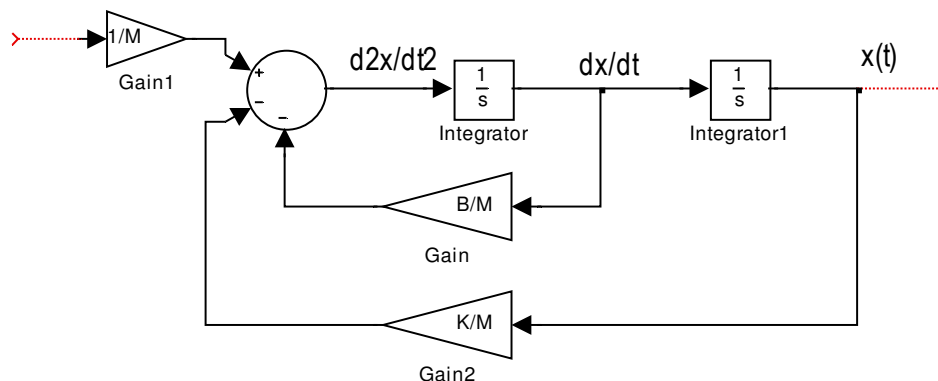


Figura 7.13.- Diagrama de simulación de la ecuación diferencial (7.1)

4) **Conexión de una señal de entrada y un bloque de visualización para la salida.** Para generar una señal de entrada  $f(t)$  en este ejemplo se usará una señal escalón unitario y se visualizará la salida mediante un bloque 'Scope'. El diagrama completo queda finalmente como se muestra en la figura 7.14.

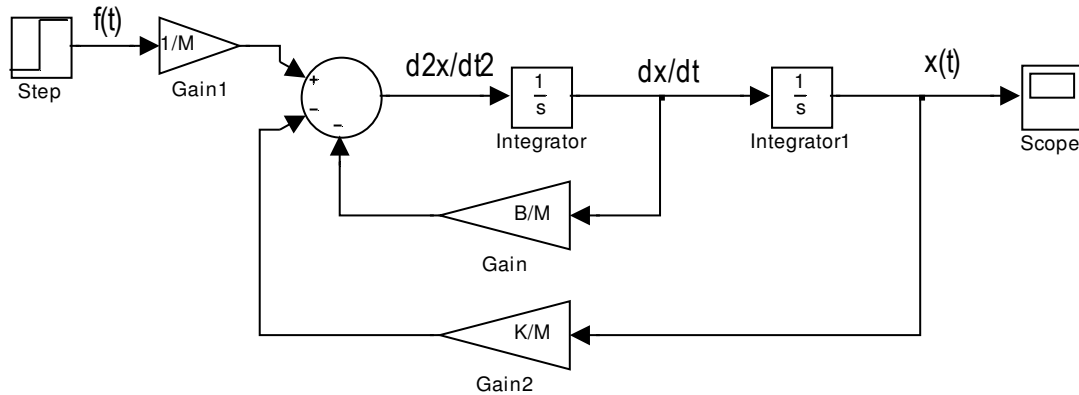


Figura 7.14.- Diagrama de simulación completo para el sistema masa-resorte-amortiguador.

5) **Asignación de valores a los bloques de ganancias constantes.**- En este caso consideraremos que  $M=1\text{Kg}$ .  $K=10\text{ Nt seg/m}$ ,  $B=1\text{ Nt}$ . Como en este caso los bloques de ganancia constante se dejaron indicados en términos de las ganancias  $K$ ,  $B$  y  $M$ , se tienen dos opciones para arrancar la simulación:

- Calcular a mano los valores numéricos de cada ganancia y modificar los bloques de ganancia con los valores numéricos calculados.
- Guardar el diagrama de simulación bajo el nombre '*sistema\_mec.mdl*' por ejemplo y Lanzar la simulación desde la línea de comandos de Matlab como sigue:

```
>> M=1;
>> K=10;
>> B=1;
>> sim('sistema_mec.mdl');
```

Una vez arrancada la simulación se abrirá el bloque 'Scope' para mostrar el comportamiento de la variable de salida  $x(t)$ , la cual deberá ser similar a la mostrada en la figura 7.15.

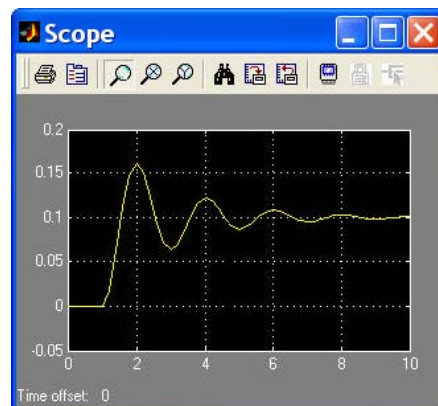


Figura 7.15.- Salida  $x(t)$  después de la simulación.

### Simulación de un Sistema No Lineal.

El procedimiento explicado en este último ejemplo es el más adecuado para simular **Sistemas No Lineales** a partir de su ecuación diferencial no lineal, por ejemplo, para la varilla actuada habíamos obtenido la siguiente ecuación diferencial en la práctica No. 5.

$$\ddot{\theta} = -a_2\dot{\theta} - a_1 \sin \theta + b_0\tau \quad (7.3)$$

donde:  $a_2 = \frac{3\mu}{ml^2}$ ,  $a_1 = \frac{3g}{2l}$  y  $b_0 = \frac{3}{ml^2}$

Usando el procedimiento descrito en el ejemplo anterior obtenemos el diagrama de simulación de la figura 7.16

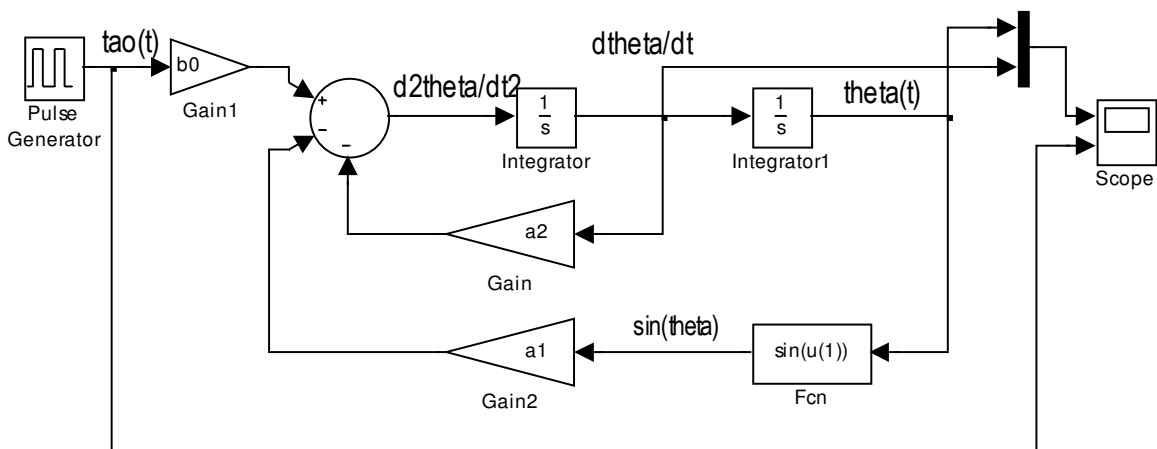


Figura 7.16.- Diagrama de simulación de la varilla actuada.

En la figura 7.16 se observa que el bloque 'Scope' utilizado tiene dos entradas en lugar de una, El número de entradas de un bloque 'Scope' se puede configurar dando clic sobre el icono de configuración de parámetros del bloque, el cual tiene la forma mostrada en la figura 7.17:



Figura 7.17.- Icono de configuración de parámetros del bloque 'Scope'

Al dar clic sobre este ícono aparece la ventana de configuración mostrada en la figura 7.18, en la cual se deberá introducir el número de subgráficas a mostrar en el cuadro de texto 'Number of axes', en este caso se seleccionan 2 subgráficas.

Además, en el diagrama de simulación de la figura 7.16 se observa que se está utilizando como bloque de entrada un generador de pulsos. Este bloque se configura para generar un pulso de amplitud 1 y de ancho 0.5 seg que inicia en el instante 0.5 y luego permanece en cero durante el resto del tiempo de simulación (10 seg). Para lograr esto el bloque 'Pulse Generator' se configura al darle doble clic con los valores de los parámetros mostrados en la figura 7.19.

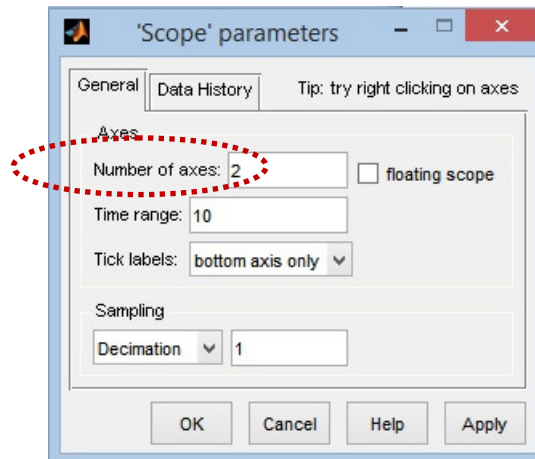


Figura 7.18.- Selección de dos subgráficas para el bloque 'Scope'

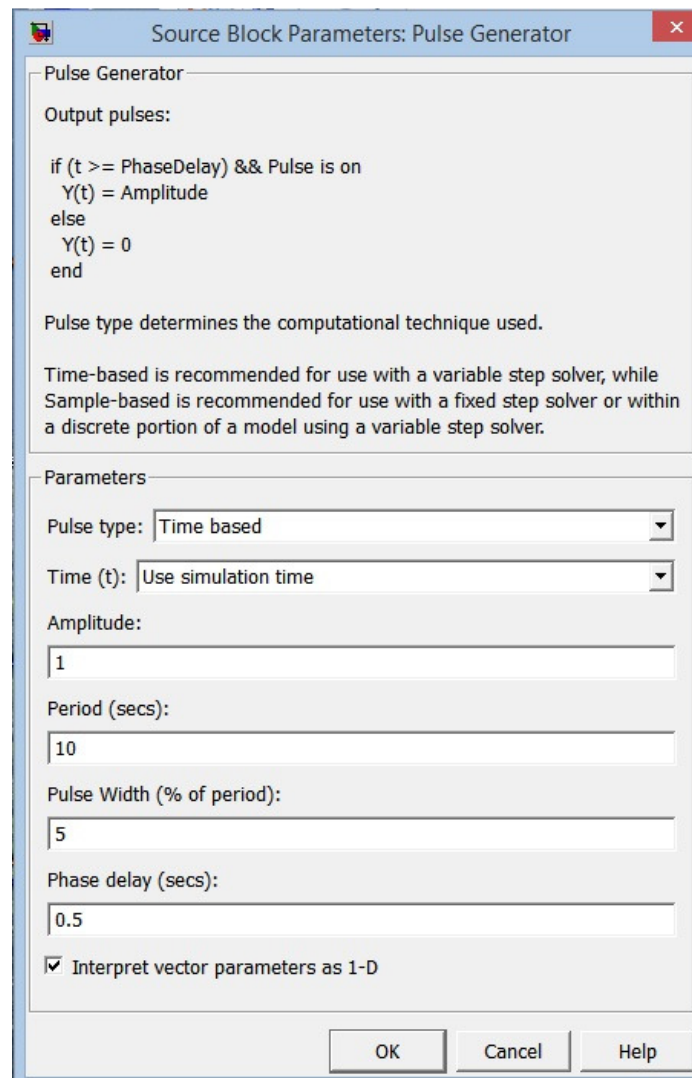


Figura 7.19.- Configuración del bloque 'Pulse Generator' para generar un pulso de amplitud 1, ancho 0.5 seg que inicia en el instante 0.5 seg.

Para ejecutar la simulación desde la línea de comandos de Matlab con los valores utilizados en la práctica No. 5, se guarda el diagrama de simulación con el nombre '*varilla\_actuada.mdl*' y se ejecuta el siguiente código:

```
>> m=0.1;
>> mu=0.01;
>> L=0.5;
>> g=9.81;
>> a2=3*mu/m/L^2;
>> a1=3*g/2/L;
>> b0=3/m/L^2;
>> sim('varilla_actuada.mdl');
```

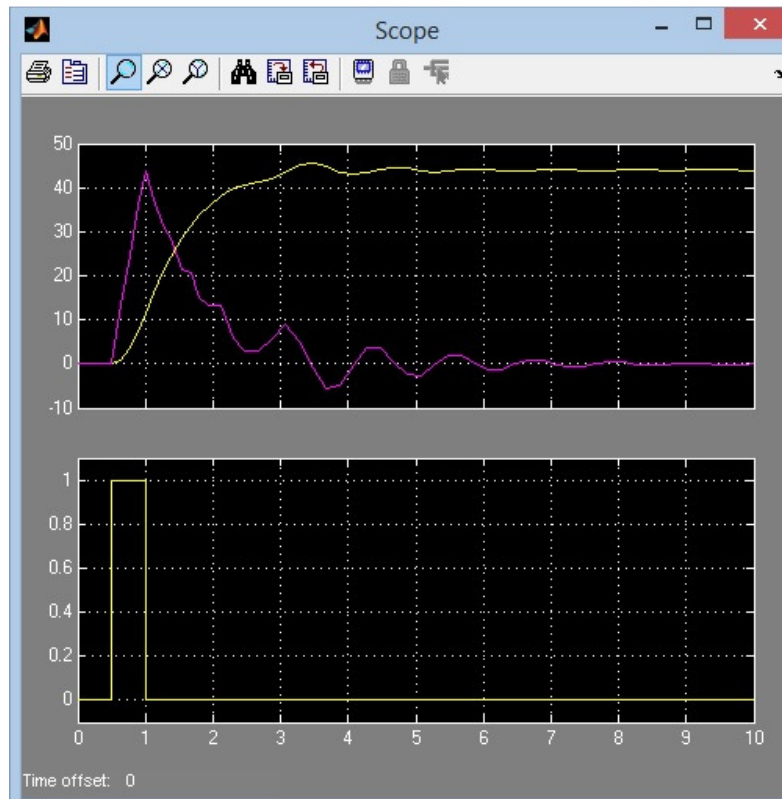


Figura 7.20.- Respuesta de la varilla actuada mostrada por el bloque 'Scope'

**Ejercicio.** Simular el comportamiento del circuito eléctrico de la figura 7.21.

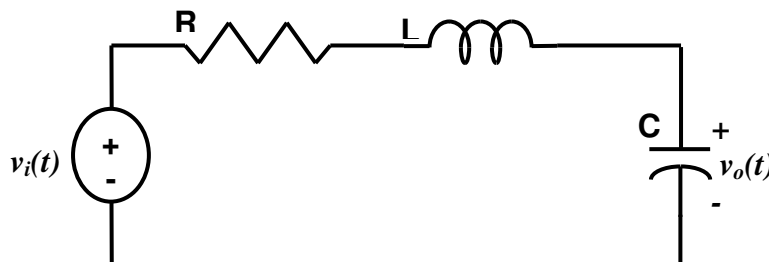


Figura 7.21.- Circuito RLC a simular

Cuya ecuación diferencial es

$$LC \frac{d^2 v_o(t)}{dt^2} + RC \frac{dv_o(t)}{dt} + v_o(t) = v_i(t) \quad (7.4)$$

Se tomará como entrada la señal de voltaje  $v_i(t)$  y como salida la señal de voltaje  $v_o(t)$  y se probarán los siguientes dos conjuntos de valores para los parámetros del circuito:

Conjunto No. 1	L=500 mH, R=20 ohms, C=500 $\mu$ F
Conjunto No. 2	L=500 mH, R=100 ohms, C=500 $\mu$ F

Se comprobarán los resultados obtenidos para dos tipos de entradas simulando de cero a 0.5 seg:

- $v_i(t) = \text{escalón unitario}$ . Este caso corresponde a la respuesta transitoria del circuito a un voltaje de corriente directa.
- $v_i(t) = \text{senoidal}$ . (amplitud 1V y frecuencia  $20\pi$  rad/seg). Este caso corresponde a la respuesta transitoria del circuito a un voltaje de corriente alterna.

### Desarrollo de la Práctica.

1. Probar todos los ejemplos propuestos por el profesor conforme los va explicando.
2. Realizar todos los ejercicios propuestos.
3. Contestar el cuestionario de evaluación de la práctica.

### Reportar:

1) El diagrama de simulación elaborado para el ejercicio del circuito RLC y las gráficas obtenidas en las cuatro simulaciones.

2) Obtener la función de transferencia del circuito del ejercicio  $G(s) = \frac{V_o(s)}{V_i(s)}$  y comparar los resultados obtenidos en Simulink ante una entrada escalón unitario utilizando el comando **step** de Matlab.

3) En la respuesta del circuito del ejercicio ante una entrada de corriente alterna se pueden apreciar dos partes de la respuesta La primera parte es la respuesta transitoria, la cual dura un tiempo  $t_s$  y luego queda una respuesta puramente sinusoidal que dura hasta el infinito, llamada respuesta en estado estacionario o estable.

3a) ¿cuánto vale aproximadamente  $t_s$ ?

3b) Extiende la simulación hasta un tiempo grande, por ejemplo 5 seg y observa ¿cual es la amplitud de la respuesta en estado estacionario?