

**UNIVERSIDAD DE SEVILLA**

**ESCUELA SUPERIOR DE INGENIEROS  
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y  
AUTOMÁTICA**

**MANUAL DE SIMULINK PARA LA ASIGNATURA DE  
TEORÍA DE SISTEMAS**

**Alicia Arce Rubio, Guilherme Vianna Raffo**

Sevilla, Marzo de 2009.



# Índice general

<b>1. Introducción</b>	<b>1</b>
<b>2. Entorno Gráfico</b>	<b>2</b>
<b>3. Bloques Principales</b>	<b>5</b>
3.1. Introducción . . . . .	5
3.2. Librería de Sistemas Lineales en Tiempo-Continuo ( <i>Continuous</i> ) . . . . .	6
3.2.1. Bloque <i>Derivative</i> . . . . .	6
3.2.2. Bloque <i>Integrator</i> ) . . . . .	6
3.2.3. Bloque <i>State-Space</i> . . . . .	7
3.2.4. Bloque <i>Zero-Pole</i> . . . . .	8
3.2.5. Bloque <i>Transfer Fcn</i> . . . . .	9
3.2.6. Bloque <i>Transport Delay</i> . . . . .	9
3.3. Librería de Puertos y Subsistemas ( <i>Ports &amp; Subsystems</i> ) . . . . .	10
3.4. Librería de Sumidero ( <i>Sink</i> ) . . . . .	11
3.4.1. Bloque <i>Scope</i> . . . . .	11
3.4.2. Bloque <i>Display</i> . . . . .	13
3.4.3. Bloque <i>To Workspace</i> . . . . .	13
3.5. Librería de Funciones Definidas por el Usuario ( <i>User-Defined Functions</i> ) . . . . .	14
3.5.1. Bloque <i>Fcn</i> . . . . .	14
3.5.2. Bloque <i>MATLAB Fcn</i> . . . . .	15
3.6. Librería de Fuentes ( <i>Sources</i> ) . . . . .	16
3.6.1. Bloque <i>From Workspace</i> . . . . .	16
3.6.2. Bloque <i>Constant</i> . . . . .	17
3.6.3. Bloque <i>Signal Generator</i> . . . . .	18
3.6.4. Bloque <i>Ramp</i> . . . . .	18
3.6.5. Bloque <i>Sine Wave</i> . . . . .	19
3.6.6. Bloque <i>Step</i> . . . . .	20
3.7. Librería de Operaciones Matemáticas ( <i>Math Operations</i> ) . . . . .	21
3.7.1. Bloque <i>Sum</i> . . . . .	21
3.7.2. Bloque <i>Product</i> . . . . .	22
3.7.3. Bloque <i>Gain</i> . . . . .	22
3.7.4. Bloque <i>Trigonometric Function</i> . . . . .	23
3.7.5. Bloque <i>Math Function</i> . . . . .	23
3.8. Librería de Ruta de Señales ( <i>Routing Signals</i> ) . . . . .	24
3.8.1. Bloque <i>Manual Switch</i> . . . . .	24
3.8.2. Bloques <i>Mux</i> y <i>Demux</i> . . . . .	24

<b>4. Simulaciones</b>	<b>26</b>
4.1. Obtención de las ecuaciones . . . . .	27
4.2. Modelar las ecuaciones: identificación de los bloques de Simulink® . . . .	27
4.3. Creación del archivo de Simulink® . . . . .	29
4.4. Configuración de los parámetros de la simulación . . . . .	31
4.4.1. Tiempo de simulación . . . . .	32
4.4.2. Método de integración . . . . .	32
4.4.3. Intervalo de muestreo . . . . .	33
4.5. Análisis de los resultados . . . . .	33

# Capítulo 1

## Introducción

La teoría de control pretende resolver la regulación de sistemas reales, para ello es fundamental plantear modelos y diseñar controladores. Dichos modelos y controladores deben ser validados para comprobar su comportamiento antes de plantear una implementación real. Esta validación se realiza mediante simulaciones para diferentes condiciones, es por ello necesario tener claro que significa simular en este campo. Simular sistemas es resolver las ecuaciones que modelan sistemas y controladores durante un periodo de tiempo y para unas condiciones determinadas. Utilizar una buena herramienta de simulación es importante para obtener resultados de forma fácil y rápida. Esta guía presenta el software Simulink<sup>®</sup> que es una herramienta que se utiliza por un gran número de colectivos para realizar tareas de simulación de modelos y controladores avanzados.

El programa Simulink<sup>®</sup> presenta ventajas frente a otros programas matemáticos que podrían ser también utilizados para resolver las ecuaciones de los sistemas, tales como un entorno interactivo y un conjunto de librerías con bloques personalizables que permiten simular, implementar y probar una serie de sistemas variables con el tiempo. Además Simulink<sup>®</sup> está integrado en Matlab<sup>®</sup> y por ello es posible tener acceso a una amplia gama de herramientas que permiten desarrollar algoritmos, analizar y visualizar simulaciones. Esta guía pretende ser de ayuda al alumno para inicializarse en la programación de esta útil herramienta de simulación y está dividida en las siguientes secciones; en el capítulo 2 se presenta las nociones básicas para familiarizarse con el entorno gráfico del programa Simulink<sup>®</sup>, describiendo las opciones más básicas para manejar dicho programa. El capítulo 3, describe en detalle los bloques más usados para crear archivos en Simulink<sup>®</sup>, explicando las funcionalidades y los parámetros configurables de cada uno de ellos. Por último, el capítulo 4 describe los pasos a seguir para simular un sistema, detallando la configuración de los parámetros específicos de simulación.

# Capítulo 2

## Entorno Gráfico

Para poder utilizar el programa Simulink<sup>®</sup> correctamente es importante estar familiarizado con las ventanas y las herramientas. Al programa Simulink<sup>®</sup> se accede a través del programa Matlab<sup>®</sup>. En la figura 2.1 se muestra el entorno gráfico de la ventana principal del programa Matlab<sup>®</sup> desde donde se accede al programa Simulink<sup>®</sup> de varias formas; en el menú 'File' (marcado en la figura como 'Archivo') se desprende un listado de opciones donde en la opción 'New' se debe elegir la sub-opción 'Model', inmediatamente se abrirá otra ventana correspondiente a la figura 2.2, la cual es la ventana principal del programa Simulink<sup>®</sup>; por otro lado también se puede acceder a dicho programa mediante la librería de Simulink<sup>®</sup> seleccionando el icono marcado en la figura 2.1 como 'Librería de simulink<sup>®</sup>', se abre una ventana que corresponde a la figura 2.3, y desde esa ventana se debe seleccionar el menú 'Archivo', la opción 'New' y sub-opción 'Model', o bien pinchar directamente el icono que representa una página en blanco que se marca en la figura 2.3 como 'Nuevo modelo'. Una vez que se llega al programa Simulink<sup>®</sup> es importante tener claro cuales son las opciones más importantes para realizar correctamente las simulaciones.

La figura 2.2 muestra la ventana principal de Simulink<sup>®</sup>. En esta ventana se encuentra el icono marcado como 'Librería de Simulink<sup>®</sup>'. Haciendo clic en esta opción se abre la librería (figura 2.3) donde se encuentran los bloques que permiten crear cualquier tipo de modelo o controlador que se desee simular. También son importantes las opciones marcadas como 'Activar simulación', 'Parar simulación' y 'Tiempo de simulación' que permiten comenzar a simular cualquier archivo Simulink<sup>®</sup> creado en esa misma ventana, parar esa simulación en cualquier momento o bien cambiar el tiempo que se desea simular correspondientemente.

Debe ser aclarado que simular un sistema significa resolver unas ecuaciones que describen un sistema real o un controlador durante un periodo de tiempo, por lo que el tiempo de simulación es el periodo total de tiempo para el cual el programa resuelve las ecuaciones, siendo por lo tanto un parámetro que debe ser elegido adecuadamente para poder analizar y representar correctamente los resultados simulados. Por otro lado la opción marcada como 'Parámetros de simulación' en la figura 2.2 abre una ventana más completa donde es posible configurar parámetros más avanzados y mejorar los resultados de las simulaciones. Los parámetros básicos de la configuración de las simulaciones serán explicados detalladamente en capítulos posteriores. Las opciones marcadas como



se utilizan para crear los archivos de Simulink®. Como se ha comentado anteriormente desde esta librería se pueden abrir archivos Simulink® y también crear nuevos archivos mediante las opciones marcadas como 'Archivo' y 'Nuevo modelo'. Cada uno de los bloques, que serán detallados en el siguiente capítulo, tienen referenciado un nombre para poder ser buscados de forma más rápida a través de la casilla marcada en la figura 2.3 como 'Búsqueda de bloques', donde escribiendo el nombre y haciendo clic en el icono que representa unos catalejos se realiza la búsqueda del bloque que tenga el nombre especificado entre todos los 'toolboxes'. Los resultados de la búsqueda se mostrarán en la sub-ventana marcada en la figura 2.3 como 'Listado bloques'.

En la sub-ventana marcada como 'toolboxes' se encuentran listados todos los 'toolboxes', herramientas, de Simulink® disponibles para la versión instalada. Si se hace clic en cada herramienta o 'toolbox' se desplegará en la sub-ventana 'Listado de bloques' los bloques que están dentro de ese 'toolbox' y que podrán ser utilizados para crear archivos de Simulink®. Por último remarcar que la sub-ventana 'Descripción de bloques' describe los bloques que se marquen en la sub-ventana 'Listado bloques' y será de ayuda para entender el funcionamiento de estos mismos, describiendo la funcionalidad, las entradas y las salidas.

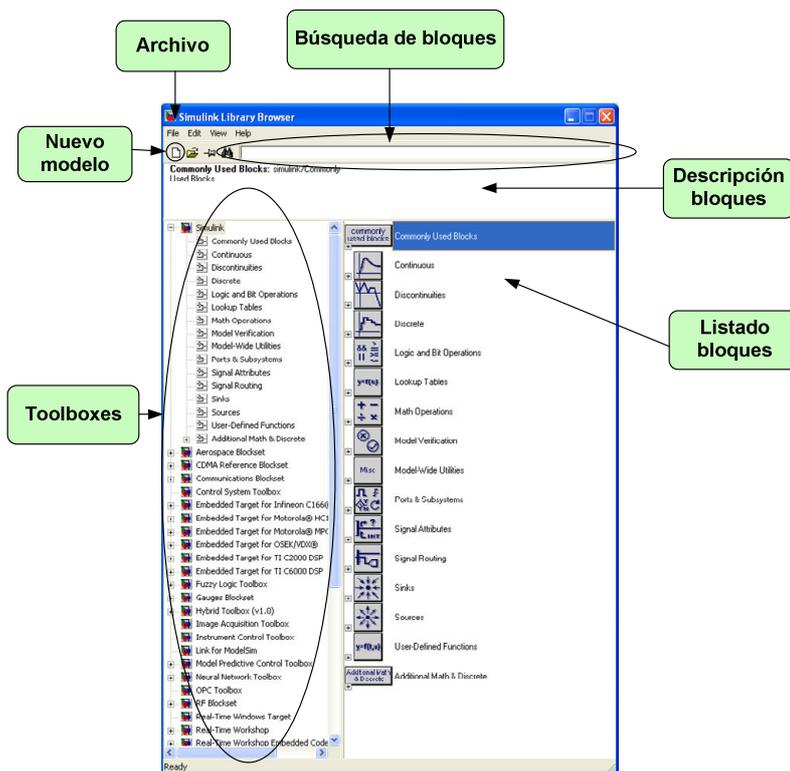


Figura 2.3: Librería Simulink®.

Las opciones 'Help' que aparecen en cada una de las ventanas descritas en este capítulo son de gran ayuda para profundizar en la programación de Simulink®.

Todas la ventanas mostradas en las figuras han sido obtenidas de la versión Matlab/Simulink® 7.1, puede que difieran un poco con otras versiones pero la mayoría de las opciones pueden ser encontradas con la misma apariencia.

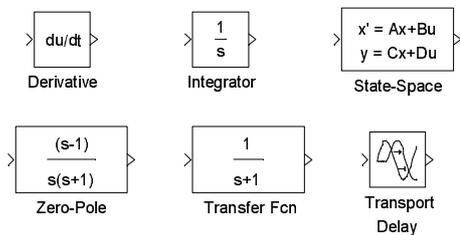
# Capítulo 3

## Bloques Principales

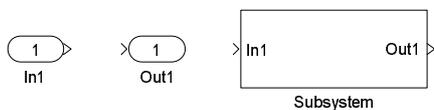
### 3.1. Introducción

En este capítulo serán presentados los bloques más utilizados en Simulink<sup>®</sup> y necesarios para la asignatura de Teoría de Sistemas. En la figura 3.1 se ilustran los bloques que se detallarán en este capítulo. Cada de uno de estos bloques podrán ser configurados por el alumno dependiendo de las exigencias del problema que se pretenda resolver mediante la ventana de diálogo de configuración de cada bloque. A esta ventana se accede haciendo doble clic sobre el bloque seleccionado.

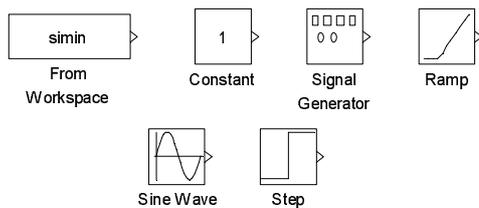
#### Librería de Sistemas Lineales en Tiempo-Continuo (Continuous)



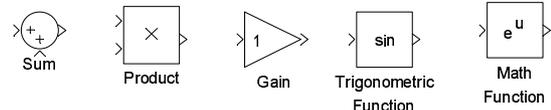
#### Librería de Puertos y Subsistemas (Ports & Subsystems)



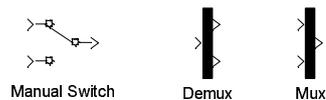
#### Librería de Fuentes (Sources)



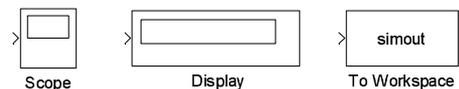
#### Librería de Operaciones Matemáticas (Math Operations)



#### Librería de Ruta de Señales (Signal Routing)



#### Librería Sumidero (Sink)



#### Librería de Funciones Definidas por el Usuario (User-Defined Functions)

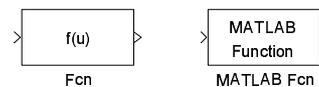


Figura 3.1: Bloques principales.

## 3.2. Librería de Sistemas Lineales en Tiempo-Continuo (Continuous)

En este apartado se describirán los bloques para la implementación de sistemas lineales en tiempo-continuo comúnmente utilizados en la asignatura. Estos bloques son presentados en la figura 3.2.

Librería de Sistemas Lineales en Tiempo-Continuo  
(Continuous)

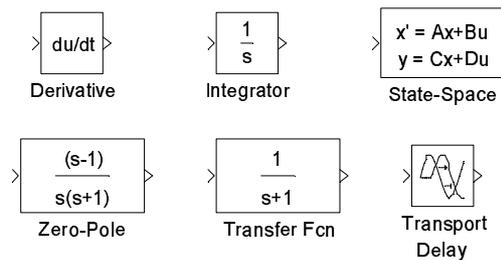
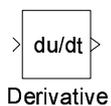


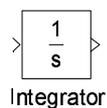
Figura 3.2: Bloques de la librería *Continuous*.

### 3.2.1. Bloque *Derivative*



El bloque '*Derivative*' aproxima la derivada de su entrada considerando los valores iniciales de la salida igual a 0. La exactitud de los resultados dependen del tamaño del periodo de muestreo utilizado en la simulación. Pequeños pasos de muestreo permiten obtener una curva de la salida más suave y exacta, ver section 4.4.3.

### 3.2.2. Bloque *Integrator*



El bloque '*Integrator*' integra su entrada. Los resultados de la integración van a depender del método de integración que se seleccione en el menú 'Configuration Parameters' al que se accede mediante la opción marcada como '*Parámetros de simulación*' de la figura 2.2, ver sección 4.4.2. El programa Simulink<sup>®</sup> trata el bloque integrador como un sistema dinámico con un estado, su salida. La entrada de este bloque es la derivada en el tiempo del estado. El algoritmo de integración numérica seleccionado calcula la salida

del bloque integrador en el periodo de muestreo actual usando el valor de entrada actual y del paso anterior. El bloque también provee una opción de condición inicial que permite configurar el estado inicial.

La ventana de dialogo de parámetros para el bloque '*Integrator*' es presentada en la Figura 3.3.

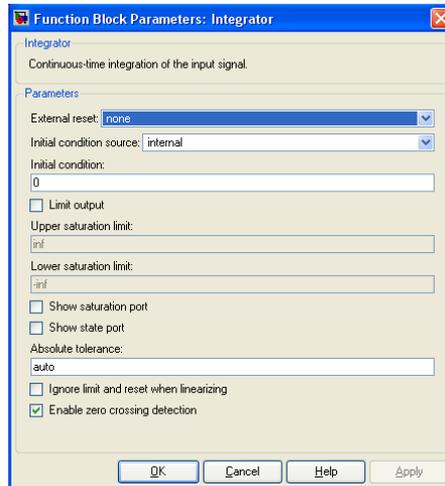


Figura 3.3: Parámetros del bloque *Integrator*.

La ventana de dialogo '*Function Block Parameter*' permite especificar valores para la condición inicial. Esta ventana también permite especificar limites inferiores y superiores para la integración. El resto de parámetros y configuraciones de este bloque no serán necesarios para esta asignatura.

Para determinar los limites inferiores y superiores se debe seleccionar la opción '*Limit output*' y asignar valores en los campos: '*Upper saturation limit*' y '*Lower saturation limit*'.

### 3.2.3. Bloque *State-Space*

$$\left. \begin{array}{l} \dot{x} = Ax + Bu \\ y = Cx + Du \end{array} \right\} \text{State-Space}$$

El bloque '*State-Space*' implementa un sistema definido a través de ecuaciones en el espacio de estados.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (3.1)$$

donde  $x$  y  $u$  son vectores columna, la matriz  $A$  debe ser definida cuadrada  $n \times n$ , con  $n$  siendo el número de estados del sistema, la matriz  $B$  debe de ser definida con dimensión  $n \times m$ , con  $m$  siendo el número de entradas, la matriz  $C$  es definida con dimensión  $r \times n$ , donde  $r$  representa el número de salidas, y la matriz  $D$  posee dimensión  $r \times m$ .

La ventana de dialogo de configuración de parámetros para el bloque '*State-Space*' se presenta en la Figura 3.4

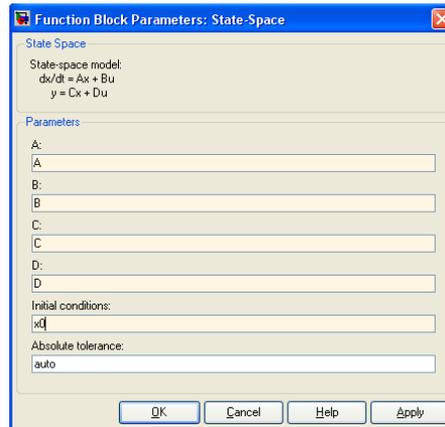
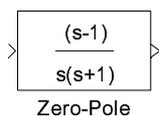


Figura 3.4: Parámetros del bloque *State-Space*.

### 3.2.4. Bloque *Zero-Pole*



El bloque '*Zero-Pole*' implementa un sistema con ceros, polos y ganancia especificada en el dominio- $s$ . Este bloque representa la función de transferencia particularizada con los parámetros especificados para un sistema dado.

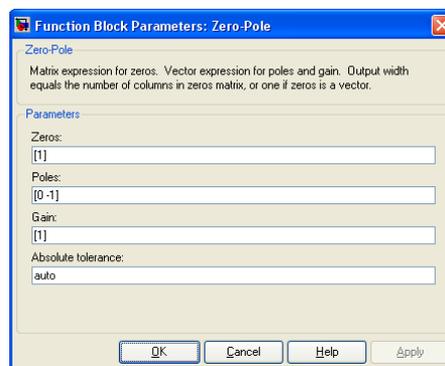


Figura 3.5: Parámetros del bloque *Zero-Pole*.

Los parámetros del sistema pueden ser especificados en la ventana de configuración como una expresión o como un vector. El bloque modificará su apariencia dependiendo de la especificación de los parámetros. Por ejemplo, si en la ventana de dialogo '*Function Block Parameters*' se especifica '*Zeros*' como  $[-2 \quad -4 \quad -6 \quad -8]$ , '*Poles*' como  $[-1 \quad -3 \quad -5 \quad -7 \quad -9]$ , y '*Gain*' como 25, el bloque se presentará como se muestra en la figura 3.6.

Si cada parámetro es especificado como una variable simbólica, por ejemplo, **ceros**, **polos**, **Ganancia**, a la que se le asigna unos valores en la ventana de comando del

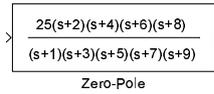


Figura 3.6: Bloque *Zero-Pole* especificado a través de vectores.

programa MatLab<sup>®</sup> (figura 2.1, 'ventana comandos Matlab'): **ceros=[-2 -4 -6 -8]; polos=[-1 -3 -5 -7 -9]; Ganancia=25**, el bloque representará la función de transferencia a través de las variables simbólicas seguidas por (s), como se muestra en la figura 3.7.

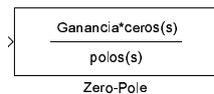
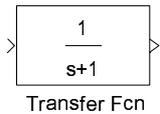


Figura 3.7: Bloque *Zero-Pole* especificado a través de variables.

### 3.2.5. Bloque *Transfer Fcn*



El bloque '*Transfer Fcn*' implementa una función de transferencia con la entrada  $U(s)$  y la salida  $Y(s)$ , como se muestra a continuación:

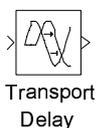
$$G(s) = \frac{U(s)}{Y(s)} = \frac{num(s)}{dens} \quad (3.2)$$

Asumiendo un sistema de primera orden con un polo en  $s = -10$  y un zero en  $s = -2$ , modelado por la función de transferencia:

$$G(s) = \frac{U(s)}{Y(s)} = \frac{s+2}{s+10} \quad (3.3)$$

Este modelo se programa utilizando el bloque '*Transfer Fcn*', a través de la ventana de dialogo presentada en la figura 3.8, donde el numerador es [1 2] y el denominador es [1 10].

### 3.2.6. Bloque *Transport Delay*



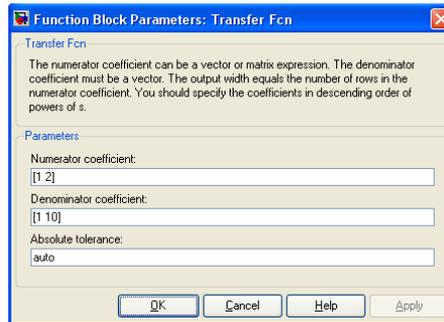


Figura 3.8: Parámetros del bloque *Transfer Fcn*.

El bloque '*Transport Delay*' retrasa la entrada del bloque por un determinado periodo de tiempo. Este bloque puede ser usado para simular retrasos de tiempo. En la ventana de configuración, el bloque permite seleccionar un valor inicial '*Initial output*' que se mantendrá constante hasta que el tiempo de la simulación exceda el valor del retraso '*Time delay*'. El parámetro '*Time delay*' no debe de ser negativo. El tiempo de muestreo debe ser seleccionado adecuadamente de manera que el tiempo de retraso sea mayor que ese valor.

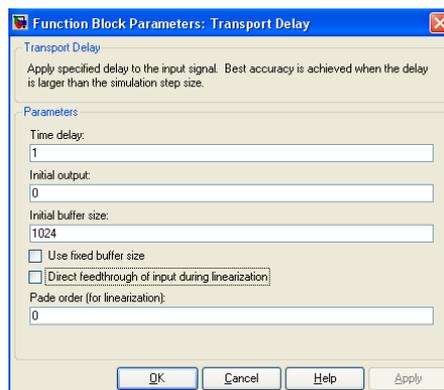


Figura 3.9: Parámetros del bloque *Transport Delay*.

### 3.3. Librería de Puertos y Subsistemas (*Ports & Subsystems*)

En este apartado se detallarán únicamente los bloques de esta librería presentados en la figura 3.10.

Los bloques de puerto de entrada y puerto de salida (del inglés *Inport Block* y *Outport Block*) son puertos que sirven para conectar un sistema externo con un subsistema (interno). El bloque Subsistema (del inglés *Subsystem*) representa un subsistema de un sistema. Cuando el modelo o el sistema de control aumenta en tamaño y complejidad se puede simplificar agrupando bloques en subsistemas.

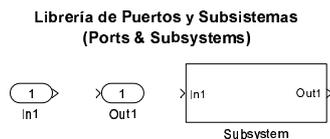


Figura 3.10: Bloques de la librería *Ports & Subsystems*.

Para crear un subsistema, se inserta el bloque ‘*Subsystem*’ en la ventana Simulink (figura 2.2). Este bloque abre una ventana (doble clic sobre el bloque) en la que se debe programar los bloques pertenecientes al subsistema. Si se desea crear un subsistema directamente de un grupo de bloques que se hayan programado en la ventana principal de Simulink<sup>®</sup>, se debe seleccionar los bloques deseados y pulsar el botón derecho del ratón sobre estos eligiendo la opción ‘*Create Subsystem*’ del menú.

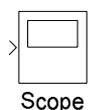
### 3.4. Librería de Sumidero (*Sink*)

Este apartado describirá los bloques más utilizados de la librería ‘*Sink*’. Los bloques detallados son presentados en la figura 3.11.



Figura 3.11: Librería *Sink*.

#### 3.4.1. Bloque *Scope*



El bloque ‘*Scope*’ representa gráficamente la entrada conectada a este bloque con respecto al tiempo de simulación. Este bloque permite representar varias variables a la vez para el mismo periodo de tiempo. El ‘*Scope*’ permite ajustar el tiempo y el rango de los valores de entrada presentados. Se puede mover y redefinir el tamaño de la ventana ‘*Scope*’ y se puede modificar los valores de sus parámetros durante la simulación.

Si la señal de entrada al bloque ‘*Scope*’ está formada por varias variables (en lugar de un vector es una matriz), éste asigna colores a cada elemento de la señal en el siguiente orden: amarillo, magenta, cian, rojo, verde y azul oscuro. Cuando la señal posee más de seis elementos, se repite el orden de los colores. Se pueden ajustar los límites del eje-y pulsando el botón derecho sobre la gráfica y seleccionando la opción ‘*Axis Properties*’.

La ventana del bloque 'Scope' posee varios iconos en la barra de herramienta que permiten realizar 'zoom' en la gráfica, preservar las configuraciones de los ejes para la simulación siguiente, limitar los datos presentados y guardar los datos en el espacio de trabajo.

Entretanto, el icono con mayor utilidad es el denominado 'Parameters'. Si se pulsa este botón, la primera pestaña que aparece es la de los parámetros generales, mostrado en la figura 3.12. En esta pestaña un parámetro importante es el 'Number of axes' que

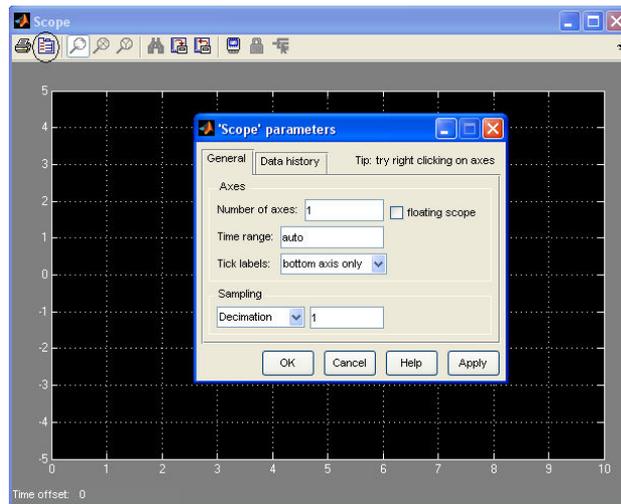


Figura 3.12: Icono *Parameters*.

permite seleccionar el número de ejes que se desean representar en la gráfica del bloque 'Scope'.

En la figura 3.13 se presentan los parámetros de la pestaña 'Data history'. El parámetro 'Limit data points to last' permite especificar cuantos puntos serán representados durante la simulación. Por ejemplo, si la simulación posee un periodo de muestreo muy pequeño, durante la simulación se generará un número muy grande de puntos, lo que implica que esta opción debe de tener un número muy elevado para poder visualizar toda la simulación correctamente. Si esta opción aparece como no seleccionada, el bloque 'Scope' por defecto representará todos los puntos generados.

A través de este cuadro de diálogo es posible también guardar los datos de las variables representadas en el espacio de trabajo del programa Matlab®. En la opción 'Variable name' se define el nombre de la variable y en 'Format' se configura el formato con el se guardarán los datos: 'Structure with time', 'Structure' y 'Array'. Por ejemplo, si se selecciona el formato 'Array', los datos serán guardados en una matriz, donde en la primera columna se almacenan el vector del tiempo de simulación, y de la segunda columna el vector de la señal de entrada del bloque.

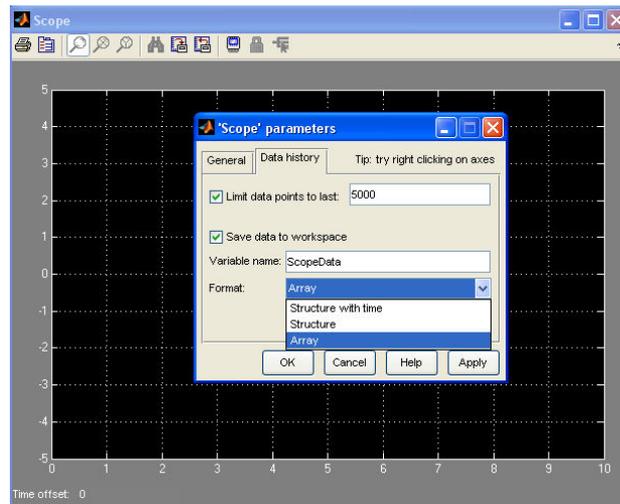
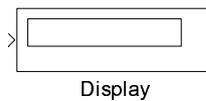


Figura 3.13: Icono *Parameters*.

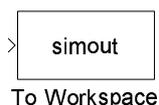
### 3.4.2. Bloque *Display*



El bloque '*Display*' muestra por pantalla el valor de su entrada. Los formatos de los datos visualizados son los mismos utilizados en MatLab<sup>®</sup>. El parámetro '*Decimation*' habilita la visualización de los datos con la  $n$ -ésima muestra, donde  $n$  es el factor de conversión al sistema decimal. El parámetro '*Sample time*' sirve para especificar el intervalo de muestreo para cual cada punto es visualizado.

Se puede modificar el tamaño del bloque para visualizar correctamente los valores mostrados por pantalla. Es posible modificarlo tanto vertical como horizontalmente. La presencia de un pequeño triángulo negro indica que el bloque no está mostrando todos los elementos por pantalla.

### 3.4.3. Bloque *To Workspace*



El bloque '*To Workspace*' envía su entrada al espacio de trabajo '*workspace*' de MatLab<sup>®</sup>. Este bloque envía los valores de la entrada a una variable con el nombre especificado en la opción '*Variable name*'. La opción '*Save format*' determina el formato de la variable de salida.

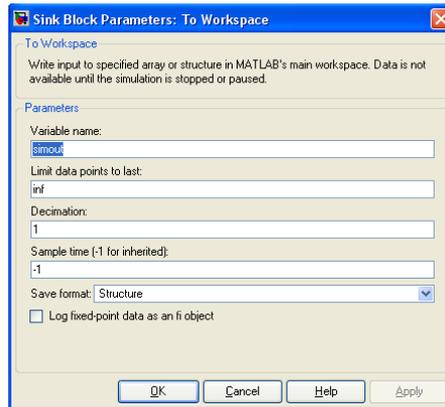


Figura 3.14: Parámetros del bloque *To Workspace*.

### 3.5. Librería de Funciones Definidas por el Usuario (*User-Defined Functions*)

En este apartado se presentarán los bloques '*Fcn*' y '*MATLAB Fcn*' de la librería '*User-Defined Functions*'.

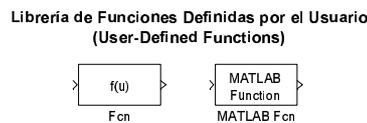
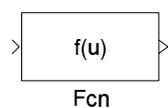


Figura 3.15: Librería *User-Defined Functions*.

#### 3.5.1. Bloque *Fcn*



El bloque '*Fcn*' ejecuta una ecuación especificada que depende de la entrada del bloque, denominada  $u$ . Si  $u$  es un vector,  $u(i)$  representa el  $i$ -ésimo elemento del vector.  $u(1)$  o  $u$  representa el primer elemento. La ecuación puede incluir constantes numéricas, operadores aritméticos, operadores relacionales, operadores lógicos y funciones matemáticas como, `abs`, `acos`, `asin`, `atan`, `atan2`, `ceil`, `cos`, `cosh`, `exp`, `fabs`, `floor`, `hypot`, `ln`, `log`, `log10`, `pow`, `power`, `rem`, `sgn`, `sin`, `sinh`, `sqrt`, `tan`, y `tanh`.

**Ejemplo 1** Dada la siguiente expresión:

$$y = -\frac{1}{4} \cos(2t) - \ln(\sec(2t) + \tan(2t)) + k_1 \cos(2t) + k_2 \sin(2t) \quad (3.4)$$

y representada por el diagrama de bloques de la figura 3.16,

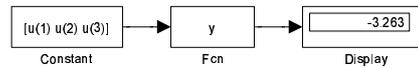


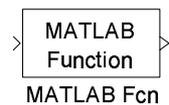
Figura 3.16: Diagrama de bloques del ejemplo 1.

donde  $u(1) = t = \pi/6$ ,  $u(2) = k_1 = -1$  y  $u(4) = k_2 = -3$ , y en la ventana de comando de MatLab son definidos de la siguiente forma:

```
u(1)=pi/6; u(2)=-1; u(3)=-3;
```

```
y=-(1/4)*cos(2*u(1))*log(sec(2*u(1))+tan(2*u(1)))+u(2)*cos(2*u(1))+u(3)*sin(2*u(1));
```

### 3.5.2. Bloque *MATLAB Fcn*



El bloque '*MATLAB Fcn*' ejecuta una determinada función Matlab o expresión de la entrada. Este bloque es más lento que el bloque '*Fcn*' ya que llama a la función MatLab para cada paso de integración.

Si se desea implementar una función Matlab que haya sido previamente creada como se muestra a continuación en el editor de m-function dentro del programa Matlab<sup>®</sup>,

```
function y=ejemplo(u)
```

```
y=-(1/4)*cos(2*u(1))*log(sec(2*u(1))+tan(2*u(1)))...
+u(2)*cos(2*u(1))+u(3)*sin(2*u(1));
```

Al archivo se le debe asignar el mismo nombre con el que se define la función dentro del archivo m-file, en este caso sería `ejemplo.m`. El bloque deberá llamar a esa función por lo que debe ser adecuadamente configurado con el nombre de la función como se ve en la figura 3.17.

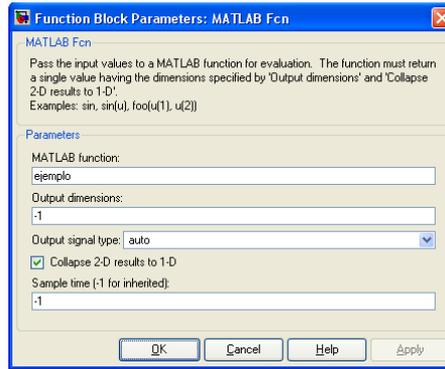


Figura 3.17: Parámetros del bloque *MATLAB Fcn*.

### 3.6. Librería de Fuentes (*Sources*)

En este apartado se presentarán los principales bloques que sirven como fuentes de señales en la simulación. Estos bloques están definidos en la librería '*Source*'. Serán comentados los siguientes bloques: '*From Workspace*', '*Constant*', '*Signal Generator*', '*Ramp*', '*Sine Wave*' y '*Step*'.

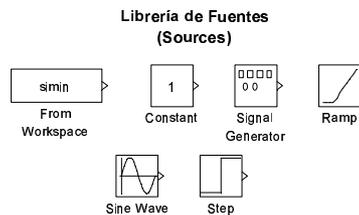


Figura 3.18: Librería *Source*.

#### 3.6.1. Bloque *From Workspace*



El bloque '*From Workspace*' lee datos del espacio de trabajo '*workspace*' de MatLab®. Los datos del espacio de trabajo son especificados con el parámetro '*Data*' a través de una matriz de dos dimensiones (ej. [t,u] son dos variables definidas en el espacio de trabajo).

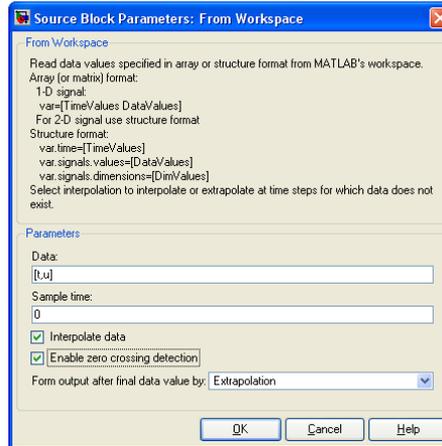
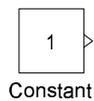


Figura 3.19: Parámetros del bloque *From Workspace*.

### 3.6.2. Bloque *Constant*



El bloque '*Constant*' es usado para definir un valor constante real o complejo. Este bloque acepta salidas escalares, vectores (1-D) o matrices (2-D), dependiendo de la dimensión del parámetro '*Constant value*' que se especifica y si la opción '*Interprete vector parameters as 1-D*' está seleccionada o no. La salida del bloque posee la misma dimensión y los mismos elementos que la opción '*Constant value*'. Si se configura esta opción como un vector (matriz de 1-D), se debe marcar '*Interprete vector parameters*' como 1-D. Si esta opción no es debidamente configurada el bloque considera el parámetro '*Constant value*' como una matriz 2-D.

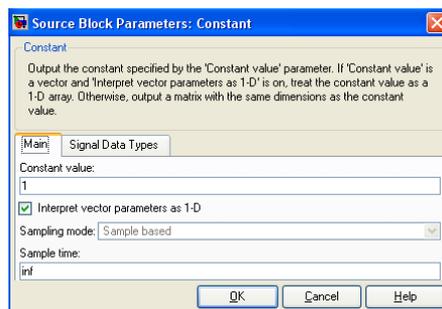
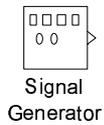


Figura 3.20: Parámetros del bloque *Constant*.

### 3.6.3. Bloque *Signal Generator*



El bloque '*Signal Generator*' puede generar cuatro diferentes tipos de formas de onda: onda seno ('*sine*'), onda cuadrada ('*square*'), onda diente de sierra ('*sawtooth*') y onda aleatoria ('*random*'). Los parámetros de las señales son expresados en hercios o radianes por segundo. Se puede invertir la onda configurando el valor de la amplitud en negativo en la ventana de dialogo de parámetros.

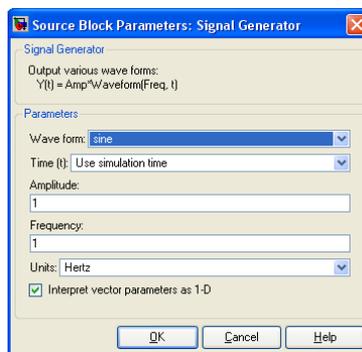
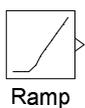


Figura 3.21: Parámetros del bloque *Signal Generator*.

### 3.6.4. Bloque *Ramp*



El bloque '*Ramp*' genera una señal que empieza en un instante de tiempo especificado con un valor también previamente configurado y que evoluciona con una pendiente determinada en el bloque. Las características de la señal generada son configuradas mediante las siguientes opciones: pendiente ('*Slope*'), tiempo de inicio ('*Start Time*') y la condición inicial de la salida ('*Initial Output*'). Ver figura 3.22.

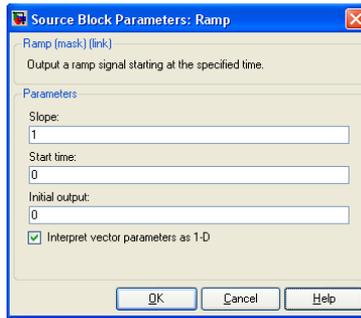


Figura 3.22: Parámetros del bloque *Ramp*.

### 3.6.5. Bloque *Sine Wave*



Sine Wave

El bloque '*Sine Wave*' genera una onda seno. Se puede generar una onda coseno configurando el parámetro de fase ('*Phase*') con el valor  $\pi/2$ . El bloque '*Sine Wave*' puede ser definido de dos modos diferentes a través del parámetro '*Sine type*' como modo basado en tiempo o como modo basado en muestras.

El modo basado en tiempo posee dos sub-modos: sub-modo continuo o sub-modo discreto. Se utiliza el parámetro '*Sample time*' para especificar que el bloque trabaje en sub-modo continuo o discreto. Para el sub-modo continuo se especifica el valor 0, y para el sub-modo discreto se especifica un valor mayor que cero.

El modo basado en muestras requiere un tiempo discreto finito. Un valor del parámetro '*Sample time*' mayor que cero provoca que el bloque se comporte como si estuviera siendo modificado por un mantenedor de orden cero '*Zero Order Holder*'.

La ventana de dialogo de configuración de parámetros de este bloque es presentada en la figura 3.23.

Los parámetros de configuración son descritos a continuación,

- *Amplitud*: la amplitud de la señal;
- *Bias*: valor (DC) constante agregado al seno para producir una salida con '*offset*' en el eje-y;
- *Frequency*: la frecuencia en radianes por segundo. Este parámetro aparece solo para el modo basado en tiempo.
- *Samples per period*: número de muestras por periodo. Este parámetro aparece solo para el modo basado en muestras.
- *Phase*: el desfase en radianes. Este parámetro aparece para el modo basado en tiempo.
- *Number of offset samples*: El desplazamiento en número de muestras de tiempo. Este parámetro aparece solo para el modo basado en muestras.

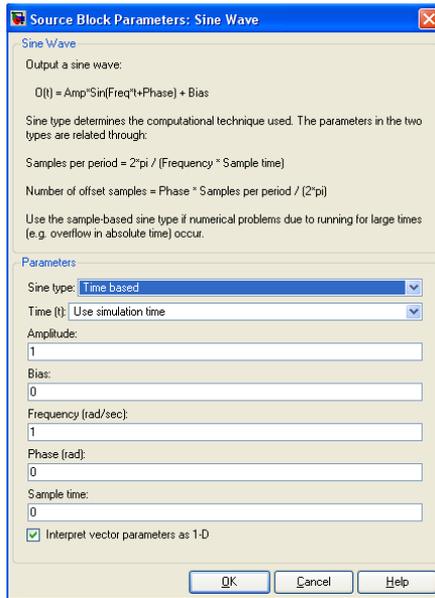
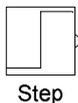


Figura 3.23: Parámetros del bloque *Sine Wave*.

- *Sample time*: Periodo de muestreo. El valor patrón es cero, pero si el tipo de seno es basado en muestras se debe definir el periodo de muestreo mayor que cero.

### 3.6.6. Bloque *Step*



El bloque '*Step*' genera un escalón entre dos niveles definidos en un espacio de tiempo especificado. Si el tiempo de simulación es menor que el valor del parámetro '*Step time*', la salida del bloque será el valor del parámetro '*Initial value*'. Para tiempos de simulación mayores o iguales que el valor de '*Step time*', la salida es el valor del parámetro '*Final value*'.

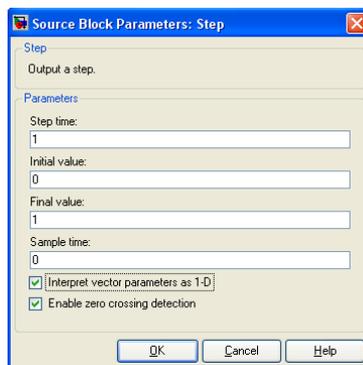


Figura 3.24: Parámetros del bloque *Step*.

## 3.7. Librería de Operaciones Matemáticas (*Math Operations*)

En este apartado se presentarán algunos bloques de operaciones matemáticas comúnmente utilizadas en el entorno Simulink®.

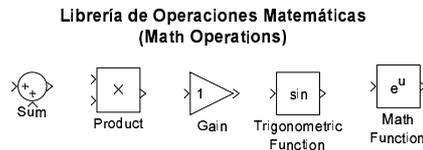


Figura 3.25: Librería *Math Operations*.

### 3.7.1. Bloque *Sum*



El bloque '*Sum*' es la implementación del bloque suma. Este bloque realiza las operaciones de adición o sustracción de sus entradas, pudiendo sumar o sustraer entradas escalares, vectoriales o matriciales. Se puede también sumar los elementos de un único vector entrada. Las operaciones del bloque son definidas en el parámetro '*List of Signs*': más (+), menos (-) y separador (|). El separador crea un espacio extra entre puertos en el icono del bloque. La forma del icono se puede definir como redonda o rectangular (del inglés *round* o *rectangular*) a través del parámetro '*Icon shape*' en la ventana de dialogo de parámetros.

Si hay dos o más entradas, el número de operaciones de suma o resta debe ser igual al número de entradas. Por ejemplo, "+ - +" requiere tres entradas y se configura el bloque para sustraer la segunda entrada a la primera entrada, y luego sumar la tercera.

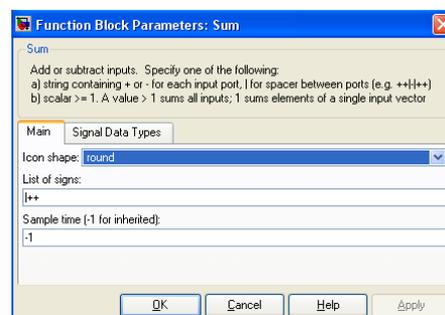
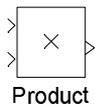


Figura 3.26: Parámetros del bloque *Sum*.

### 3.7.2. Bloque *Product*



El bloque '*Product*' realiza la multiplicación o división de sus entradas. Este bloque calcula la salida multiplicando elemento a elemento o matricialmente, dependiendo del valor del parámetro '*Multiplication*'. El número de operaciones se configura con el parámetro '*Number of inputs*'.

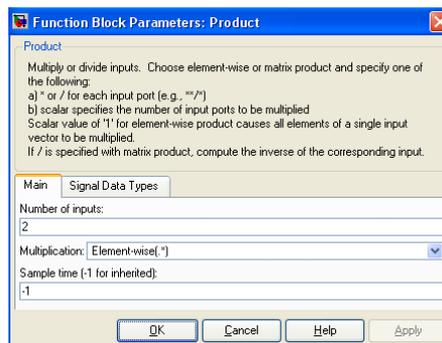
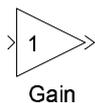


Figura 3.27: Parámetros del bloque *Product*.

### 3.7.3. Bloque *Gain*



El bloque '*Gain*' multiplica la entrada por un valor constante (ganancia). La entrada y la ganancia pueden ser un escalar, un vector o una matriz. El valor de la ganancia se especifica a través del parámetro '*Gain*'. El parámetro '*Multiplication*' determina si la multiplicación es matricial o elemento a elemento. El orden de las multiplicaciones en las operaciones matriciales es configurado a través de este parámetro.

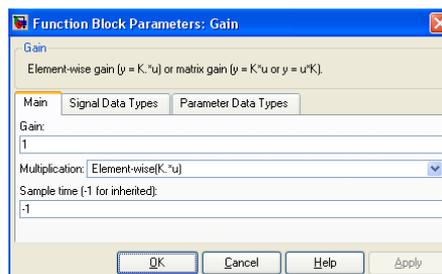
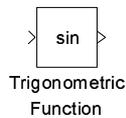


Figura 3.28: Parámetros del bloque *Gain*.

### 3.7.4. Bloque *Trigonometric Function*



El bloque '*Trigonometric Function*' realiza las principales funciones trigonométricas: seno *sine*, coseno *cosine* y tangente *tangent*; y las funciones trigonométricas inversas: arco seno *asin*, arco coseno *acos*, arco tangente *atan* y *atan2*; funciones hiperbólicas: *sinh*, *cosh* y *tanh*, y las funciones hiperbólicas inversas: *asinh*, *acosh*, *atanh*. Si se elige la función *atan2*, el bloque presenta dos entradas, la primera entrada es el eje-*y* o la parte compleja del argumento de la función y la segunda entrada es el eje-*x* o la parte real del argumento de la función.

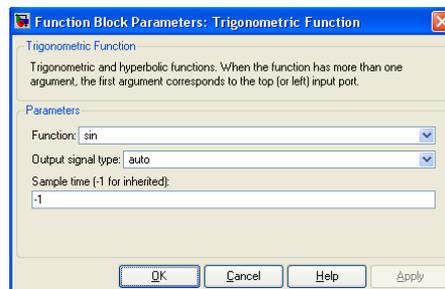
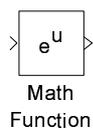


Figura 3.29: Parámetros del bloque *Trigonometric Function*.

### 3.7.5. Bloque *Math Function*



El bloque '*Math Function*' implementa las siguientes funciones matemáticas:  $\exp$ ,  $\log$ ,  $10^u$ ,  $\log_{10}$ , *magnitude*<sup>2</sup>, *square*, *sqrt*, *pow*, *conj* (conjugado complejo), *reciprocal*, *hypot* (cálculo de la raíz cuadrada de la suma de cuadrados), *rem* (resto de la división), *mod* (entero de la división), *transpose* (traspuesta de un vector o matriz) y *hemiltian* (una matriz cuadrada, tal que  $A^T = A$ ).

La salida del bloque es el resultado de la operación de la función sobre la entrada. El nombre de la función aparece sobre el bloque. Se utiliza el bloque '*Math Function*' en el caso que se desee una salida vectorial o matricial ya que el bloque '*Fcn*' tiene características similares pero las salidas son sólo escalares.

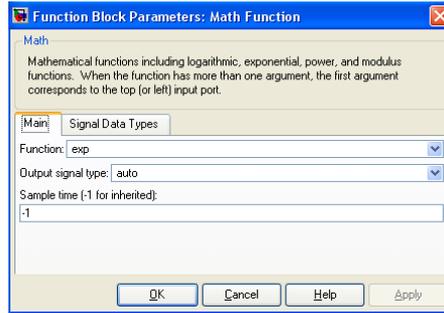


Figura 3.30: Parámetros del bloque *Math Function*.

## 3.8. Librería de Ruta de Señales (*Routing Signals*)

En este apartado se presentarán tres bloques de la librería '*Signal Routing*': '*Manual Switch*', '*Mux*' y '*Demux*'.

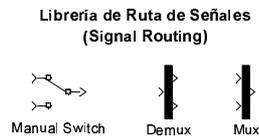
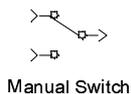


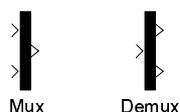
Figura 3.31: Librería *Signal Routing*.

### 3.8.1. Bloque *Manual Switch*



El bloque '*Manual Switch*' conmuta sus entradas pasando sólo una de ellas a través de su salida. No existe ventana de dialogo para este bloque, por lo que para conmutar entre las entradas se debe pulsar dos veces sobre el bloque. Este bloque mantiene el estado determinado cuando el archivo Simulink es guardado.

### 3.8.2. Bloques *Mux* y *Demux*



El bloque '*Mux*' combina sus entradas en una única salida. Las señales de entrada pueden ser escalares, vectores o matrices. El parámetro '*Number of Inputs*' permite especificar el número de señales de entrada y su dimensión. Un valor de  $-1$  significa que el puerto correspondiente puede aceptar señales de cualquiera dimensión.



Figura 3.32: Parámetros del bloque *Mux*.

El bloque '*Demux*' extrae las componentes de una señal de entrada y provee las componentes en separadas señales. El bloque acepta tanto señales vectoriales como buses de señales. El parámetro '*Number of outputs*' permite especificar el número y dimensión de cada puerto de salida. Si no se configura la dimensión de las salidas, el bloque lo determina automáticamente.

# Capítulo 4

## Simulaciones

En este capítulo se describirán los pasos a seguir para crear un archivo Simulink<sup>®</sup> y realizar una simulación. Para ello se va a tomar un ejemplo sencillo de un sistema eléctrico que se desea modelar, el caso en estudio está esquematizado en la figura 4.1.

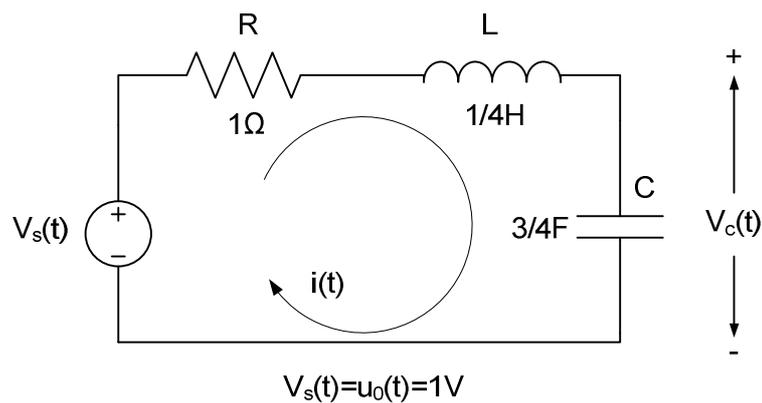


Figura 4.1: Circuito eléctrico.

Los pasos que se deben seguir para simular este ejemplo o cualquier otro, son generalmente los siguientes:

1. Obtención de las ecuaciones
2. Modelar las ecuaciones: identificación de los bloques de Simulink<sup>®</sup>
3. Creación del archivo de Simulink<sup>®</sup>
4. Configuración de los parámetros de la simulación
5. Análisis de los resultados

Para llevar a cabo correctamente cada uno de los pasos, el alumno debe estar familiarizado con las opciones básicas de Simulink<sup>®</sup> y se recomienda que el alumno haya leído detenidamente los capítulos precedentes.

## 4.1. Obtención de las ecuaciones

Esta sección tiene como objetivo plantear las ecuaciones de la respuesta temporal del sistema que se pretende simular. En este caso, el sistema que vamos a estudiar es el presentado en la figura 4.1. Aplicando las leyes de Kirschoff (procedimiento que se detalla en los apuntes de clase, tema 3) las ecuaciones que se obtienen son las siguientes:

$$i(t) = i_L(t) = i_C(t) = C \cdot \frac{dV_C(t)}{dt} \quad (4.1)$$

$$R \cdot i_L(t) + L \cdot \frac{di_L(t)}{dt} + V_C(t) = u_0(t) \quad (4.2)$$

Sustituyendo la ecuación 4.1 en la 4.2, se obtiene:

$$RC \cdot \frac{d^2V_C(t)}{dt^2} + LC \cdot \frac{dV_C(t)}{dt} + V_C(t) = u_0(t) \quad (4.3)$$

Por último, particularizando la ecuación 4.3 con los parámetros mostrados en la figura 4.1, se obtiene,

$$\frac{d^2V_C(t)}{dt^2} + 4 \frac{dV_C(t)}{dt} + 3 = 3u_0(t) \quad (4.4)$$

## 4.2. Modelar las ecuaciones: identificación de los bloques de Simulink®

Las ecuaciones que se han obtenido en la sección anterior pueden ser programadas de diversas maneras en Simulink®. En esta guía presentamos dos formas diferentes, ilustrando la amplia variedad. La primera forma de programar este caso es mediante la obtención de la transformada de Laplace del sistema y con ello la función de transferencia, y la segunda es programando directamente las ecuaciones diferenciales. Es de suma importancia antes de comenzar a programar, tener claro cuales son las variables que quieren ser estudiadas o simuladas, ya que las ecuaciones que se programen dependerán de las variables seleccionadas. Para este ejemplo, pueden ser analizadas la intensidad del circuito o la tensión del condensador conociendo la tensión de la fuente de tensión continua. Por ejemplo, para simular la intensidad del circuito habría que calcular la tensión del condensador mediante la ecuación 4.2 y luego a partir de la ecuación 4.1 obtener los resultados de la intensidad.

El cálculo de la transformada de Laplace puede estudiarse en detalle en el tema 4 de los apuntes de clase, por lo que no se especificarán en la guía los pasos para su obtención. Las transformadas de Laplace de las ecuaciones resultantes de la sección anterior son las siguientes,

$$\mathfrak{I} = \frac{3}{4}s \cdot \mathfrak{V}_c \quad (4.5)$$

$$s^2\mathfrak{V}_c + 4s\mathfrak{V}_c + 3\mathfrak{V}_c = 3\mathfrak{U}_o \quad (4.6)$$

De estas ecuaciones se pueden obtener funciones de transferencias dependiendo de las variables que se quieran analizar. De esta manera para poder analizar la intensidad del circuito y la tensión del condensador tenemos las siguientes ecuaciones,

$$\mathfrak{U}_c(s) = \frac{3}{s^2 + 4s + 3} \mathfrak{U}_0(s) \quad (4.7)$$

$$\mathfrak{I}(s) = \frac{3}{4} \cdot \frac{3s}{s^2 + 4s + 3} \mathfrak{U}_0(s) \quad (4.8)$$

Una vez calculadas las funciones de transferencias, este ejemplo puede ser programado en Simulink<sup>®</sup> mediante el bloque llamado '*Transfer Fcn*' dentro de la herramienta '*Simulink*' y del subgrupo '*Continuous*'. La configuración de este bloque se describe en el capítulo anterior. La figura 4.2 muestra el bloque '*Transfer Fcn*' configurado para la función de transferencia correspondiente a la tensión del condensador.

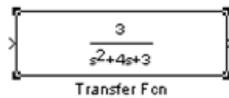


Figura 4.2: Bloque de simulink<sup>®</sup> para modelar funciones de transferencia.

Por otro lado es posible programar las ecuaciones diferenciales de forma más directa sin tener que obtener las transformadas de Laplace. Como se observa en la ecuación 4.4, se deben programar cada uno de los sumandos teniendo en cuenta que la entrada que se conoce en este ejemplo es la tensión de la fuente de continua  $u_0$ . De esta manera necesitaríamos bloques integradores, derivativos, ganancias y sumas. Todos estos bloques se han representado en la figura 4.3.



Figura 4.3: Bloque de simulink<sup>®</sup> para modelar ecuaciones diferenciales.

Los bloques que se muestran pueden encontrarse en la herramienta '*Simulink*', en los subgrupos '*Math operation*' y '*Continuous*' y han sido detallados en el capítulo anterior. Las entradas conocidas del sistema son las mismas para ambas programaciones. Dependiendo de las especificaciones de la simulación podemos programar entradas en forma de escalón, rampa, constante, ... etc. Los bloques correspondientes a la programación de las entradas se muestran en la figura 4.4, los cuales se encuentran en el subgrupo '*Sources*' del toolbox '*Simulink*' de la librería de Simulink<sup>®</sup>. Cada uno de los bloques mostrados se detallan en el capítulo anterior.



Figura 4.4: Bloques de simulink<sup>®</sup> básicos para modelar entradas.

Las variables que se desean visualizar para analizar la simulación deben ser unidas al bloque denominado 'Scope' que permite representar los resultados. Además hay otros bloques que pueden ser de ayuda para obtener información de las variables en estudio, como el bloque 'Display' que muestra los valores que toma una variable determinada durante la simulación y de forma permanente el valor final tomado por la variable una vez terminada la simulación . Estos bloques se resumen en la siguiente figura.



Figura 4.5: Bloques de simulink® básicos para visualizar variables.

Por último recordar que los bloques que se presentan en esta guía son los más usados. Para realizar simulaciones más complejas deberán utilizarse herramientas más avanzadas explotando el gran potencial que tiene el programa Simulink® para realizar simulaciones.

### 4.3. Creación del archivo de Simulink®

Identificadas las ecuaciones y los bloques que son necesarios utilizar para programar, se debe crear el archivo Simulink® para simular el sistema. Para empezar, una vez abierta la ventana que se muestra en la figura 2.2, se debe guardar el archivo con un nombre y en una carpeta que el alumno haya creado previamente. Las opciones de guardar archivo se encuentran en el menú 'File' marcado en la figura 2.2 como 'Archivo' o bien con el icono que representa un disquete. Los bloques que han sido identificados en la sección anterior deben arrastrarse con el ratón desde la ventana marcada como 'Listado bloques' de la librería de Simulink® (figura 2.3) a la ventana del archivo de Simulink® (figura 2.2). Para ello, pinchar encima del bloque de Simulink® que se desee incluir con el botón izquierdo del ratón, dejándolo pulsado mientras se arrastra el bloque a la posición deseada en el archivo Simulink® de la figura 2.2. Ver figura 4.6 donde se esquematiza el procedimiento.

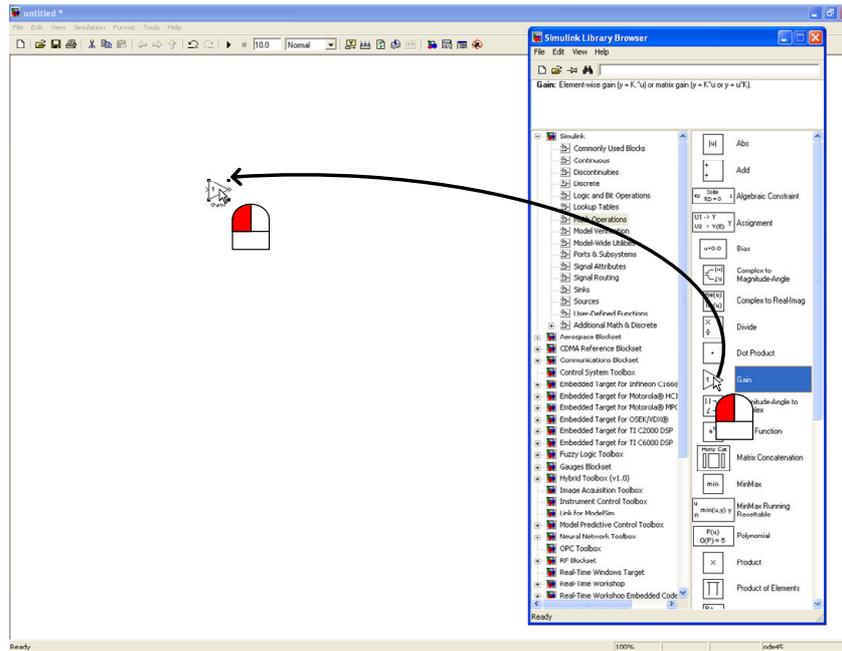


Figura 4.6: Manejo del ratón para incluir bloques al archivo de Simulink®.

Posicionados los bloques en el archivo de Simulink®, deben ser conectados. De esta forma, pinchando sobre la salida del bloque que se desea unir con el botón izquierdo del ratón y manteniéndolo pulsado, unir la salida del bloque con la entrada del siguiente bloque. Si se desean conectar líneas con bloques, se debe pinchar encima de la línea con el botón derecho del ratón, manteniéndolo presionado hasta unir con la línea con la entrada del bloque. Ver figura 4.7 para más aclaración.

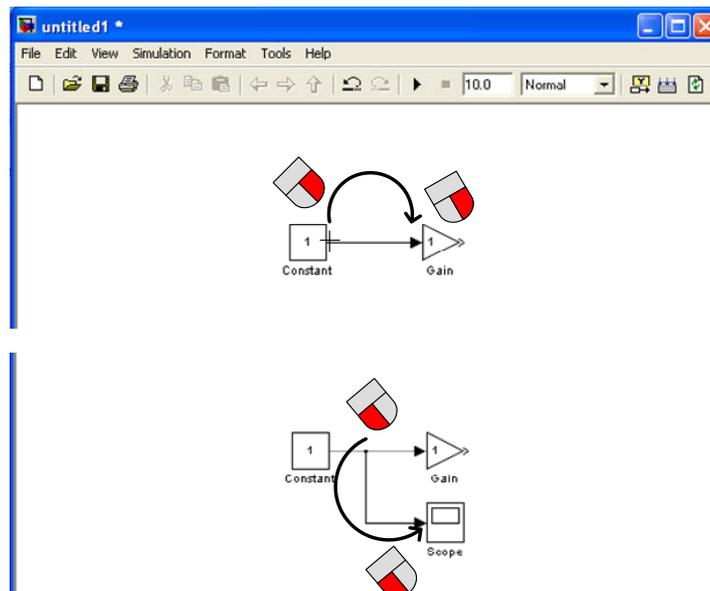


Figura 4.7: Manejo del ratón para unir bloques de Simulink®.

Siguiendo todos los pasos anteriores el sistema quedaría programado para cada uno

de los métodos presentados, como se ve en la figura 4.8. La tensión de la fuente de continua se considera constante e igual a la unidad, además se han incluido bloques 'Scope' para visualizar los resultados de la intensidad del circuito y la tensión del condensador.

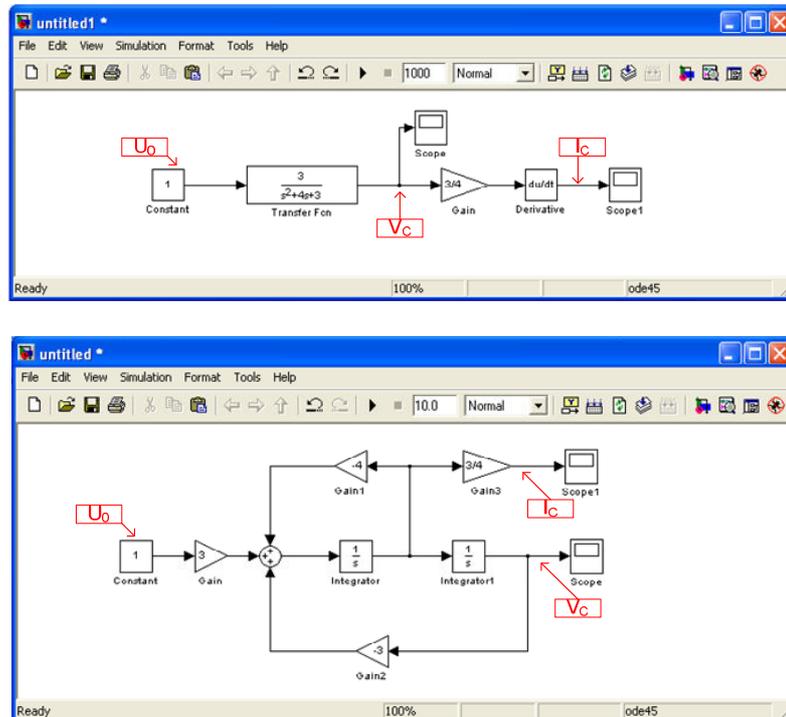


Figura 4.8: Archivos de Simulink<sup>®</sup> para el ejemplo presentado.

Los bloques se pueden invertir como se ve en la figura 4.8, haciendo clic con el botón de la derecha del ratón sobre el bloque que se desea invertir se desplegará un menú donde se debe elegir la opción 'Format' y la sub-opción 'Flip block'. Hay otra sub-opción denominada 'Rotate block' en 'Format' que permite rotar los bloques. La rotación también puede realizarse directamente seleccionando el bloque y pulsando las teclas 'control' y 'r' ('ctrl.+r').

## 4.4. Configuración de los parámetros de la simulación

En esta sección el sistema está preparado para ser simulado como se muestra en la figura 4.8 y es necesario configurar de forma adecuada las siguientes opciones básicas que se enumeran a continuación,

1. Tiempo de simulación
2. Método de integración
3. Intervalo de muestreo

### 4.4.1. Tiempo de simulación

El tiempo de simulación es un parámetro que hay que definir antes de comenzar las simulaciones. El tiempo de simulación puede ser modificado en la casilla marcada en la figura 2.2 como '*Tiempo de simulación*' o bien en la ventana '*Configuration Parameters*' que se muestra en la figura 4.9 y a la que se accede a través de la opción marcada como '*Parámetros de simulación*' en la figura 2.2 y en la sub-opción '*Configuration Parameters*'. El tiempo de simulación puede configurarse en esta ventana modificando el tiempo de comienzo de la simulación y el de parada, casillas denominadas '*Start time*' y '*Stop time*' de la figura 4.9. Por defecto siempre el tiempo de simulación será 10 segundos.

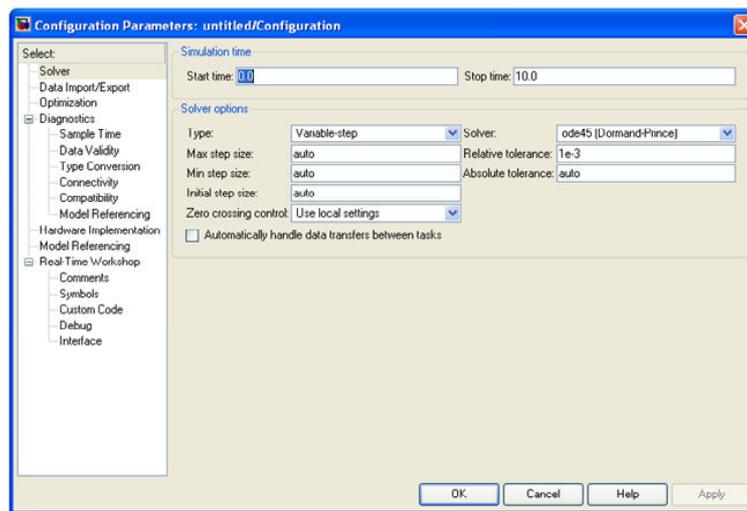


Figura 4.9: Parámetros de configuración del programa Simulink®.

### 4.4.2. Método de integración

Hay dos grupos de métodos de integración, los de paso fijo y los de paso variable. El grupo de los métodos de integración de paso fijo mantienen un tamaño de muestreo fijo para la resolución de las ecuaciones durante todas las simulaciones frente al grupo de los métodos de integración de paso variable que varían el muestreo según detecte que los resultados varíen, por ello si detecta que los resultados se mantienen constantes aumenta el muestreo y si por el contrario hay muchas variaciones reduce el muestreo. De este modo, dependiendo de la dinámica del problema que se pretenda simular, los resultados que se obtienen de las simulaciones pueden variar con el método de integración seleccionado.

El grupo de métodos de integración de paso fijo permite tener un control del muestreo por lo que son adecuados para programar en tiempo real, como desventaja frente al otro grupo, el tiempo que se tarda en obtener resultados de la simulación puede llegar a ser más lento que en el caso que se utilicen métodos de paso variable, y además si el tiempo de muestreo no ha sido correctamente seleccionado, las simulaciones pueden perder dinámica. Por otro lado el grupo de métodos de integración de paso variable simula los archivos de forma más rápida que el anterior pero puede producir errores en los resultados de la simulación para algunos tipos de problemas que mezclen dinámicas rápidas y

lentas. Como se ve en la figura 4.9, la casilla '*Type*' permite cambiar el tipo de método de integración y la casilla '*Solver*' permite elegir el tipo de método de integración para cada grupo que haya sido elegido previamente en la casilla '*Type*'. Estas casillas están marcadas por defecto con las opciones de paso variable y de ode45 (Domand-Prince). El alumno no tendrá que modificar estas opciones para el desarrollo de las prácticas.

### 4.4.3. Intervalo de muestreo

Como se ha comentado en la sección anterior el tiempo de muestreo es muy importante para los métodos de integración y una mala elección puede provocar que no se simule adecuadamente la dinámica de los sistemas en estudio.

Las opciones de configuración del tiempo de muestreo son, para el caso de los métodos de integración de paso variable, las opciones denominadas '*Max step size*' y '*Min step size*' (figura 4.9). Estas opciones están marcadas por defecto como 'auto'. La casilla '*Max step size*' indica el periodo de muestreo máximo que puede tomar el método de integración en el caso que la dinámica no varíe, la casilla '*Min step size*' indica, por el contrario, el periodo de muestreo mínimo que debe tomar el método de integración en el caso que la dinámica del sistema varíe. Si los resultados de la simulación no son los esperados y la dinámica del sistema no es la adecuada, puede ocurrir que el método de integración no esté detectando los cambios en la dinámica y tome un periodo de integración muy grande, provocando errores en la simulación. Para solucionar este problema en el caso de los métodos de integración de paso variable, se debe reducir el tiempo máximo de muestreo en la casilla '*Max step size*'.

En el caso en el que se haya elegido un método de integración de paso fijo, la opción que debe ser configurada es '*Fixed-step size*' que corresponde con el periodo de muestreo fijo que el método de integración toma para resolver las ecuaciones. Por defecto está marcada como 'auto'.

Una vez que se hayan configurado los parámetros de simulación se debe proceder a ejecutar la simulación haciendo clic en el icono marcado en la figura 2.2 como '*Activar simulación*'. Para pausar la simulación se debe hacer clic el mismo icono de ejecución y para parar la simulación pinchar el icono marcado en la figura 2.2 como '*Parar la simulación*'.

## 4.5. Análisis de los resultados

Una vez que se hayan realizado las simulaciones, es importante analizar y visualizar la evolución de las variables del sistema en estudio, además de poder acceder a los valores de las variables en forma vectorial a través del espacio de trabajo '*workspace*' del programa Matlab® y de este modo poder utilizar todos los medios que este programa presenta para poder representar y trabajar con esas variables.

Como hemos comentado anteriormente los bloques utilizados para visualizar las variables deseadas se muestran en la figura 4.5 y son explicados en el capítulo anterior detalladamente. El bloque más utilizado es '*Scope*' como se ve en la figura 4.10. Haciendo doble clic en el bloque '*Scope*' se abre una ventana (figura 4.12 que muestra la evolución de la variable o de las variables que se hayan conectado al bloque. En la figura

4.10 se ha simulado el circuito eléctrico presentado en la figura 4.1 para un voltaje de entrada de la fuente de tensión continua que evoluciona en forma de escalón.

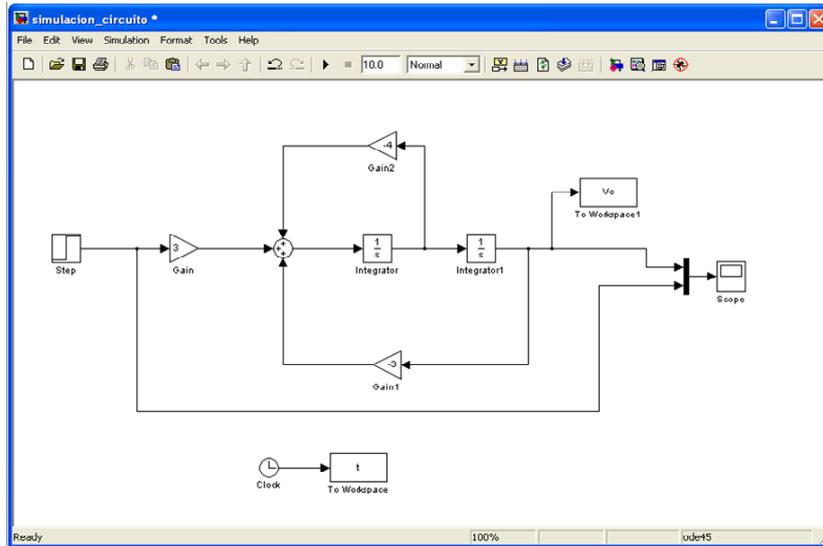


Figura 4.10: Archivo de Simulación del ejemplo.

Antes de simular es importante verificar que los parámetros del bloque 'Scope' están correctamente configurados porque si no es así, las variables no se representarán de forma adecuada. Para configurar los parámetros, hacer doble clic en el bloque de 'Scope' antes de la simulación y se abrirá la ventana que se muestra en la figura 4.11.

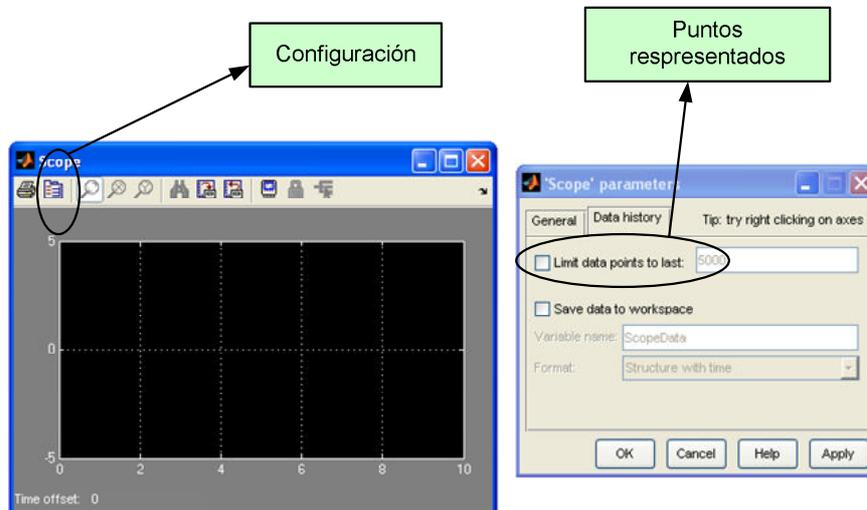


Figura 4.11: Configuración del bloque Scope.

Simplemente pinchando sobre el icono marcado como 'Configuración' en la figura 4.11, hay que verificar que la opción marcada en la misma figura como 'Puntos representados' no está seleccionada, ya que de esta manera el bloque representará todos los puntos que se calculen y no se limitará a 5000. Tener en cuenta que por defecto siempre esta opción está seleccionada.

La figura 4.12 muestra los resultados de la simulación de la figura 4.10. Como se puede ver en la figura 4.10 se han conectado dos variables al bloque 'Scope' mediante el bloque 'Mux', que son la tensión de la fuente de alimentación y la tensión del condensador. En la figura 4.12 se han destacado las opciones de aumento de la gráfica que son de gran utilidad para poder analizar y visualizar correctamente los resultados. La opción marcada como 'Herramienta zoom' permite aumentar regiones de la gráfica obtenida, para ello una vez seleccionada esta opción, hacer clic con el botón derecho del ratón en la región de la gráfica que se desee aumentar y dejar presionado este mismo botón a la vez que se arrastra el marco que aparece abarcando toda la región deseada. Otras herramientas muy útiles son los zoom parciales de ejes, por ejemplo la opción marcada como 'Zoom eje X' permite aumentar regiones del eje X que funciona de forma semejante a la opción anterior. Se pincha con el botón derecho del ratón en un punto de la región que se quiera aumentar y dejando presionado el botón se arrastra la barra que aparece abarcando la región deseada. Por último la opción marcada como 'Zoom eje Y' funciona de la misma manera que las anteriores pero en el eje Y. Por otro lado para ajustar la gráfica de manera que se visualice la simulación completa seleccionar la opción marcada en la figura como 'Ajuste pantalla'.

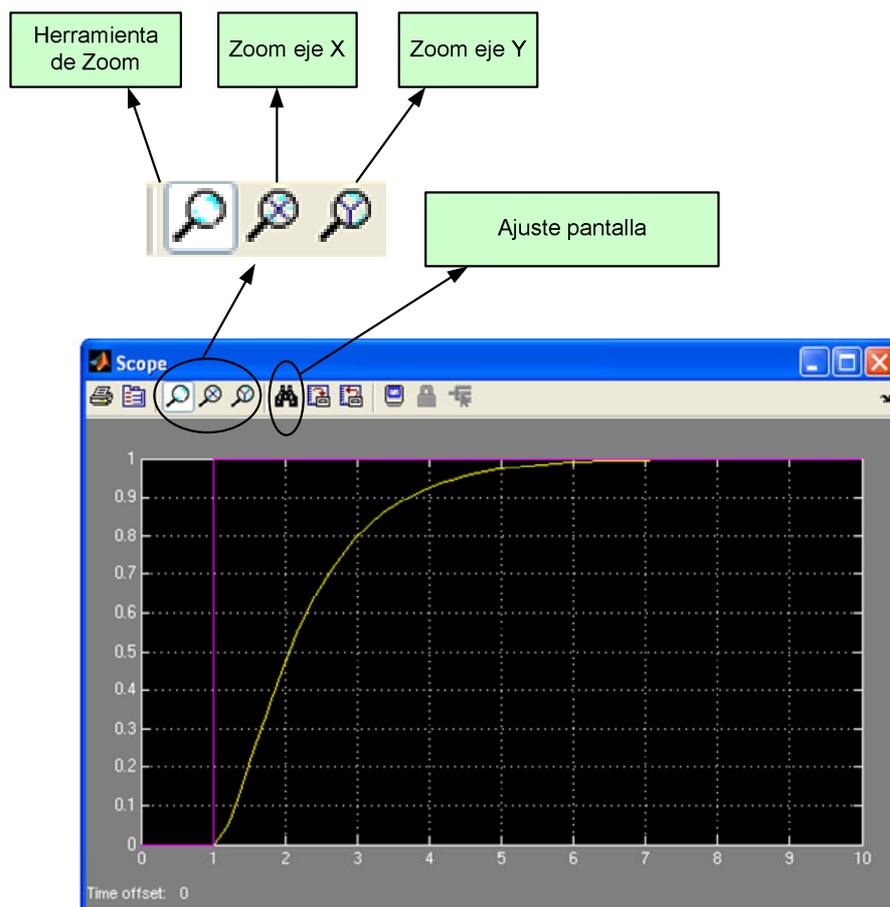


Figura 4.12: Bloque Scope.

Las variables que se han obtenido de la simulación pueden enviarse al espacio de trabajo del programa Matlab<sup>®</sup> denominado 'workspace' mediante el bloque llamado 'To

*Workspace* que está dentro de la herramienta *Simulink* y en el subgrupo *Sinks* de la librería de *Simulink*<sup>®</sup>. Este bloque debe también ser configurado como se muestra en la figura 4.13. Haciendo doble clic en el bloque seleccionado se abrirá la ventana de la figura 4.13. Primero hay que elegir un nombre para la variable que corresponderá con el nombre de la variable en el *workspace*. Esta opción se cambiará en la casilla denominada *Variable Name*. Además deberá seleccionarse la opción *Array* en la casilla *Save format*. A este bloque deben conectarse variables individualmente y estas mismas variables, una vez que se ejecute la simulación, aparecerán en la ventana del *workspace* de la figura 2.1 marcada como *Variables workspace*.

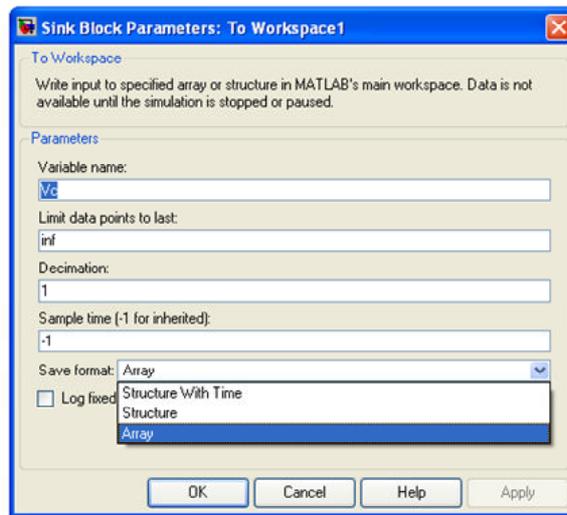


Figura 4.13: Configuración bloque *To Workspace*.

Para poder trabajar en Matlab<sup>®</sup> con los valores de las variables que se envían mediante el bloque *To Workspace*, es necesario tener la información del vector tiempo. Es decir si enviamos las variables al *workspace* sin el vector tiempo no podemos asociar a esas variables el tiempo para cada uno de los puntos obtenidos. De esa manera se incluye, como se puede observar en la figura 4.10, el bloque *Clock* que se encuentra en el grupo *Simulink*, subgrupo *Sources*. Este bloque genera automáticamente el vector de tiempo para las simulaciones y debe ser conectado a un bloque *To Workspace* para poder enviar ese vector al espacio de trabajo de Matlab<sup>®</sup> y trabajar adecuadamente con las variables.

Por ejemplo teniendo los valores de las variables y del tiempo se puede representar gráficamente en el Matlab<sup>®</sup> los resultados de la simulación. Las gráficas de Matlab<sup>®</sup> pueden ser incorporadas a documentos de Word fácilmente con la opción *Copy figure* dentro del menú *Edit*. El programa representado en La figura 4.10 envía al *workspace* la variable tiempo que se denomina en la figura 4.10 *t* y la tensión del condensador que se denomina *Vc*. Ejecutando en la ventana de comando de Matlab<sup>®</sup> marcada en la figura 2.1 como *Ventana comandos Matlab* la función *plot(t,Vc)* obtenemos la figura 4.14.

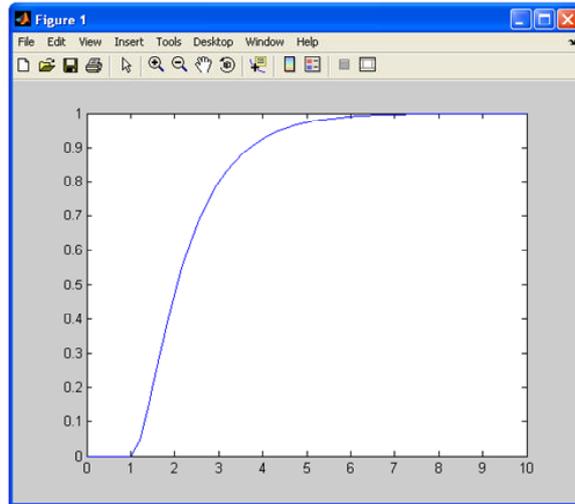


Figura 4.14: Gráfica obtenida mediante Matlab<sup>®</sup>.