

5.- MATLAB Y SIMULINK

5.1.- Matlab

5.1.1.- Introducción

Matlab es un programa de gran aceptación en la Ingeniería destinado a realizar cálculos técnico-científicos y de propósito general. En él se integran operaciones de cálculo, visualización y programación, donde la interacción con el usuario emplea una notación matemática clásica.

Los usos y aplicaciones típicas de Matlab son las siguientes:

- Matemáticas y cálculo.
- Desarrollo de algoritmos.
- Adquisición de datos.
- Modelación, simulación y prototipado.
- Análisis y procesado de datos.
- Gráficos científicos y de ingeniería.
- Desarrollo de aplicaciones.

El tipo de variable con el que trabaja Matlab es una matriz que no requiere ser dimensionada previamente en la declaración. Una de las características más interesantes consiste en que el álgebra lineal y matricial se expresa con la misma sintaxis que las operaciones aritméticas escalares. Por ejemplo, en lenguaje C, para realizar la suma de dos variables enteras o reales **b** y **c**, escribiremos:

$$a = b + c;$$

mientras que en Matlab emplearemos la misma sentencia tanto si **b** y **c** son enteros, reales, vectores o matrices.

5.1.2.- Componentes de Matlab

Matlab consta de cuatro partes fundamentales:

- **Entorno de desarrollo.** Se trata de un conjunto de utilidades que permiten el uso de funciones de Matlab y ficheros en general. Muchas de estas utilidades son interfaces gráficas de usuario. Incluye el espacio de trabajo Matlab y la ventana de comandos.
- **La librería de funciones matemáticas Matlab.** Se trata de un amplio conjunto de algoritmos, comprendiendo las funciones más elementales, como la suma, senos y cosenos o la aritmética compleja, hasta funciones más sofisticadas como la inversión de matrices, el cálculo de autovalores y transformadas rápidas de Fourier.
- **Gráficos.** Matlab dispone de un conjunto de utilidades destinadas a visualizar vectores y gráficas en forma de gráficos. Existe una gran cantidad de posibilidades para ajustar el aspecto de los gráficos, destacando la visualización tridimensional con opciones de iluminación y sombreado, y la posibilidad de crear animaciones.
- **El interfaz de aplicación de Matlab (API).** Consiste en una librería que permite escribir programas ejecutables independientes en C y otros lenguajes, accediendo, mediante DLLs, a las utilidades de cálculo matricial en Matlab.

La gestión de complementos en Matlab se realiza mediante lo que se denomina *toolboxes*, (paquetes de herramientas). Un **toolbox** de Matlab es un conjunto de funciones y algoritmos de cálculo especializados en un área de conocimiento: finanzas, tratamiento de señales, teoría de sistemas, etc.

5.1.3.- El entorno de trabajo de Matlab

5.1.3.1.- Ayuda en línea

Si se ha instalado la opción correspondiente durante la instalación de Matlab, podemos acceder a la ayuda en línea en todo momento pulsando la tecla F1. Dicha

documentación está organizada con un índice en forma de árbol y mediante herramientas de navegación como hipervínculos. Es sumamente recomendable su uso, tanto a modo de introducción como de referencia para temas específicos. Si se desea conocer documentación específica asociada a un comando de Matlab, entonces se tecleará:

```
>> doc nombre_del_comando
```

en la línea de comandos de Matlab.

5.1.3.2.- Organización de ventanas

La Figura 5.1 muestra la organización por defecto de ventanas que nos encontramos cuando arrancamos Matlab por primera vez. Las ventanas que en ella aparecen son, de arriba hacia abajo: en la parte izquierda, la estructura del directorio donde nos encontramos, y debajo de ella la historia de comandos que se han tecleado en el pasado; en la mitad derecha nos encontramos, arriba, la ventana de edición de programas Matlab (que se escriben en un lenguaje propio de Matlab y que se almacenan en ficheros .m), y debajo la línea de comandos, donde se sitúa el cursor para teclear comandos de Matlab.

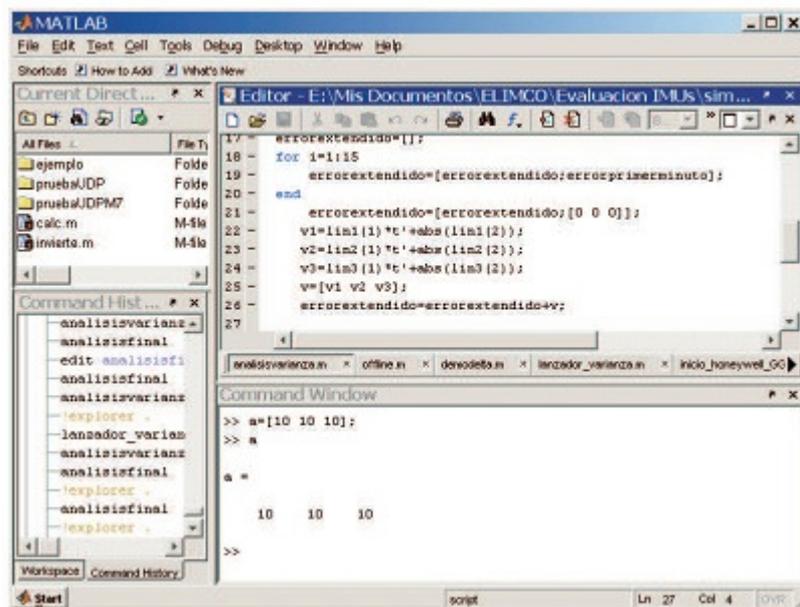


Fig. 5.1: Entorno de trabajo Matlab

5.1.3.3.- Operaciones básicas en la línea de comandos

Como se ha dicho previamente, en Matlab todos los objetos son matrices. Un escalar no es más que una matriz 1x1. En la línea de comandos podemos asignar un nombre simbólico para identificar una matriz:

```
>> a = [15; 36; -12];      %Asignación
```

Esto es una asignación de un vector columna que llevará el nombre a. A su derecha aparece un comentario, que tiene su utilidad cuando redactemos programas en Matlab.

Los objetos pueden crearse en cualquier momento. Para ello, basta con asignarles un valor mediante una asignación, como en el ejemplo previo. Los identificadores asignados para designar matrices son de libre elección, con la salvedad de que no pueden empezar por un número ni contener espacios en blanco.

Una vez creado un objeto de Matlab, este pasa a formar parte del espacio de trabajo ocupando una porción de memoria. Por tanto, a veces, tras horas de trabajo con Matlab, necesitamos eliminar los objetos que ya no se utilicen. Para ello se emplea el comando **clear**.

```
>> clear a;              % Borra a de la memoria  
>> clear;                % Borra todos los objetos del espacio de trabajo
```

En las sentencias previas, aparece el signo “;” al final de cada comando. Este comando sirve para separar unos comandos de otros. Por ejemplo, cuando escribimos varios comandos en una misma línea, estos deben estar separados por punto y coma. Además, si escribimos un comando aislado (sin “;”) y pulsamos ENTER, Matlab proporcionará siempre una salida en respuesta del comando.

Sin embargo, si escribimos el comando seguido de “;”, no se mostrará en pantalla la respuesta. Cuando los comandos forman parte de un programa, es conveniente emplearlo para evitar así desbordar la pantalla con información innecesaria.

5.1.3.4.- Operaciones básicas con Matlab

La siguiente tabla muestra las operaciones básicas aritméticas y lógicas que podemos realizar con Matlab

Expresión en Matlab	Operación
+	Suma aritmética
-	Resta aritmética o cambio de signo
*	Multiplicación aritmética
/	División
^	Elevar un número a una potencia
<	Relación “menor que”
>	Relación “mayor que”
<=	Relación “menor o igual que”
>=	Relación “mayor o igual que”
==	Relación “igual que”
~=	Relación “distinto que”
&	Producto lógico
	Suma lógica
~	Negación

Tabla 5.1: Operaciones aritméticas y lógicas en Matlab

Todas estas operaciones se aplican indistintamente a escalares, vectores y matrices.

5.1.3.5.- Funciones en Matlab

Buena parte de las operaciones que se realizan con Matlab son llamadas a funciones. Las funciones procesan información, por lo que poseen datos de entrada y de salida, que pueden ser matriciales. Los datos de entrada se especifican entre paréntesis; si son varios, éstos van separados por comas.

Las funciones son programas escritos por el usuario o incorporados en el paquete básico de Matlab. Entre estas últimas destacan las siguientes:

Nombre	Función
sin	Seno
sinh	Seno hiperbólico
cos	Coseno

cosh	Coseno hiperbólico
tan	Tangente
tanh	Tangente hiperbólica
cot	Cotangente
coth	Cotangente hiperbólica
sec	Secante
sech	Secante hiperbólica
csc	Cosecante
csch	Cosecante hiperbólica
asin	Arcoseno (inversa del seno)
asinh	Arcoseno hiperbólico (inversa del seno hiperbólico)
acos	Arcocoseno (inversa del coseno)
acosh	Arcocoseno hiperbólico (inversa del coseno hiperbólico)
atan	Arcotangente (inversa de la tangente)
atan2	Arcotengente de cuatro cuadrantes

Tabla 5.2: Funciones Elementales de Matlab: Trigonometría

Nombre	Función
exp	Exponencial
log	Logaritmo natural (base e)
log2	Logaritmo base 2
log10	Logaritmo base 10
sqrt	Raíz cuadrada

Tabla 5.3: Funciones Elementales de Matlab: Exponenciales

Nombre	Función
fix	Redondear hacia cero
floor	Redondear hacia menos infinito
ceil	Redondear hacia más infinito
round	Redondear hacia el entero más próximo
mod	Módulo de la división entera
rem	Resto de la división entera

Tabla 5.4: Funciones Elementales de Matlab: Ajuste y redondeo

Nombre	Función
inv	Matriz inversa
det	Determinante
eig	Autovalores
'	Matriz traspuesta
eye	Crear una matriz identidad dado por el número de filas/columnas

zeros	Crear una matriz de ceros dado por el número de filas/columnas
ones	Crear una matriz de unos dado por el número de filas/columnas
length	Longitud de un vector
size	Dimensiones de una matriz

Tabla 5.5: Funciones Elementales de Matlab: Operaciones matriciales

5.1.3.6.- Almacenamiento en Archivo

Matlab permite almacenar en el disco las variables del espacio de trabajo. De este modo es posible parar una sesión de trabajo y continuar en otro momento sin tener que repetir los cálculos. La orden más común para salvar los datos es **save**, que puede usarse de varias maneras. En la siguiente tabla mostramos un resumen:

Orden	Operación realizada
save	Crea el archivo de nombre <i>matlam.mat</i> en la carpeta actual. Dicho archivo contiene todas las variables que existen en ese momento en el entorno de Matlab.
save <i>nombre_archivo</i>	Crea el archivo de nombre <i>nombre_archivo.mat</i> en la carpeta actual. Dicho archivo contiene todas las variables que existen en ese momento en el entorno de Matlab.
save <i>nombre_archivo x, y, z</i>	Crea el archivo de nombre <i>nombre_archivo.mat</i> en la carpeta actual. Dicho archivo contiene solamente las variables <i>x</i> , <i>y</i> y <i>z</i> .

Tabla 5.6: Tabla resumen del comando save

Para recuperar las variables almacenadas en un fichero previamente creado, emplearemos principalmente la orden **load**. La siguiente tabla ilustra tres operaciones típicas de recuperación de datos

Orden	Operación realizada
load	Lee todas las variables del archivo <i>Matlab.mat</i> de la carpeta actual. Si alguna de las variables del disco tiene nombre coincidente con otra que previamente existe en Matlab se producirá la destrucción de la variable existente para dejar su sitio a la variable del disco

<code>load nombre_archivo</code>	Igual que en el caso anterior pero leyendo del archivo <code>nombre_archivo.mat</code> de la carpeta actual
<code>load nombre_archivo x, y, z</code>	Igual que en el caso anterior pero leyendo únicamente las variables <code>x, y, z</code>

Tabla 5.7: Tabla resumen del comando load

5.2.- Simulink

5.2.1.- Introducción

Simulink es una aplicación que permite construir y simular modelos de sistemas físicos y sistemas de control mediante diagramas de bloque. El comportamiento de dichos sistemas se define mediante funciones de transferencia, operaciones matemáticas, elementos de Matlab y señales predefinidas de todo tipo.

Simulink dispone de una serie de utilidades que facilitan la visualización, análisis y guardado de los resultados de simulación. Simulink se emplea profusamente en ingeniería de control.

Principalmente se trata de un entorno de trabajo gráfico con el que se especifican las partes de un sistema y su interconexión en forma de diagrama de bloques.

5.2.2.- Uso de Simulink

En primer lugar, lanzaremos la aplicación escribiendo *Simulink* en la línea de comandos de Matlab, a bien abriendo desde el explorador de Windows cualquier fichero con extensión *.mdl*. En el primer de los casos se abrirá la ventana de la Figura 5.2

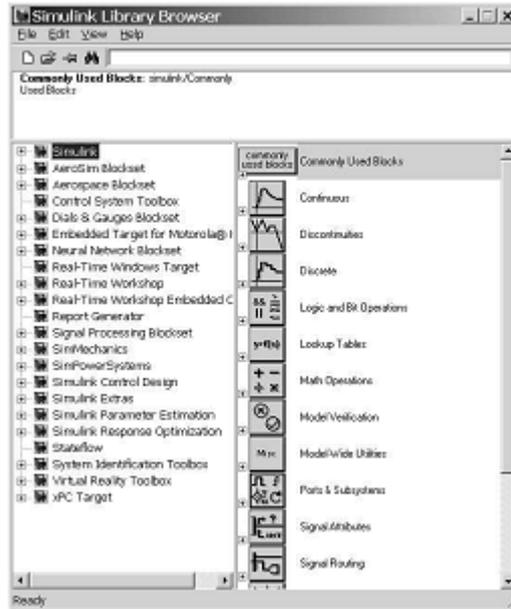


Fig. 5.2: Ventana navegación de bloques Simulink

Esta ventana inicial no está destinada a crear modelos de simulación; su función principal consiste en navegar por la enorme librería de bloques disponibles para el modelado.

En ella distinguiremos dos partes: la izquierda contiene una visión en forma de árbol de todos los Toolboxes instalados que contienen bloques Simulink. La amplitud de este árbol dependerá de las opciones que hayamos activado al seleccionar Matlab.

La parte de la derecha de la ventana de **Fig. 5.2** muestra los bloques Simulink contenidos en el Toolbox o nodo de la parte izquierda de la ventana. Estos bloques se deben arrastrar sobre el espacio de trabajo Simulink para la creación del modelo que queremos simular.

Por último, cabe indicar que en la parte superior de la ventana de inicio de Simulink hay varias herramientas como la búsqueda de un bloque determinado a partir de su nombre, que nos pueden resultar bastante útiles.

5.2.3.- El espacio de trabajo Simulink

Si pulsamos en el icono superior izquierdo de la ventana de **Fig. 5.2** (página en blanco), se abre una ventana blanca sobre la que iniciaremos la creación de un modelo de simulación. Dicha ventana se puede ver en la Figura 5.3

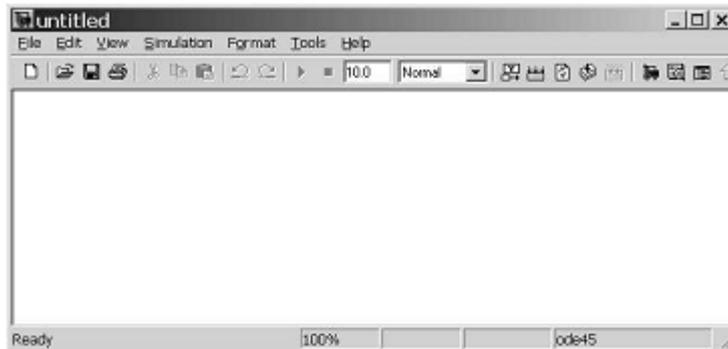


Fig. 5.3: Espacio de trabajo de Simulink

En el espacio de trabajo de Simulink crearemos un modelo insertando los bloques correspondientes. Una vez localizado el bloque que nos interesa, arrastraremos dicho bloque hacia el espacio de trabajo de Simulink. El arrastre de bloques se realiza seleccionando el icono del bloque con el botón izquierdo del ratón, y manteniendo éste pulsado se desplazará el cursor hasta la ventana del modelo.

Una vez configurados los distintos bloques con los parámetros que necesitemos, tendremos que conectarlos entre ellos. Para ello emplearemos las siguientes operaciones:

Operación	Procedimiento
Conectar Bloques (I)	Para conectar las salidas de un bloque a la entrada de otro, hacer click con el botón izquierdo del ratón en el bloque origen. Pulsar y mantener la tecla CTRL y hacer de nuevo click sobre el botón destino
Conectar Bloques (II)	También se puede extraer un cable de señal haciendo click en el saliente derecho del bloque origen y prolongar la señal (pulsando y manteniendo el botón izquierdo del ratón) hasta llegar a la parte izquierda del bloque destino.
Bifurcar cables	Un cable de señal (que lleva la salida de un bloque hacia otro bloque), puede bifurcarse para distribuir la señal a varios bloques pulsando con el botón derecho en cualquier punto del cable

Tabla 5.8: Formas de conectar los bloques

5.2.4.- Fuentes y Sumideros de señal

En las simulaciones han de existir fuentes de señales externas ya que lo que pretendemos, en general, es ver cómo responden determinados sistemas a estímulos exteriores.

A modo de referencia, la tabla 5.9 muestra algunas fuentes de señal de uso común (nodo Simulink/Source), mientras que la tabla 5.10 muestra algunos de los bloques sumidero (Simulink/Sinks) más comunes

Elemento	Función
Clock	Marcas de tiempo de la simulación. Útil para trazar gráficas con los resultados
Sin	Señal senoidal parametrizable
Step	Señal en escalón
Constant	Señal de valor constante
Signal generator	Permite elegir entre un conjunto de señales predefinidas
Random Number	Generación de ruido blanco configurable
From Workspace	Señal generada a partir de una variable del espacio de trabajo de Matlab

Tabla 5.9: Fuentes de señal en Simulink

Elemento	Función
Scope	Gráfica 2D para visualizar las señales frente al tiempo durante la simulación
XY Graph	Gráficas 2D para visualizar un gráfico X-Y creado a partir de dos señales de entrada
To Workspace	Almacena las muestras de la señal de entrada en una variable (vector) del espacio de trabajo de Matlab

Tabla 5.10: Sumideros de señal en Simulink