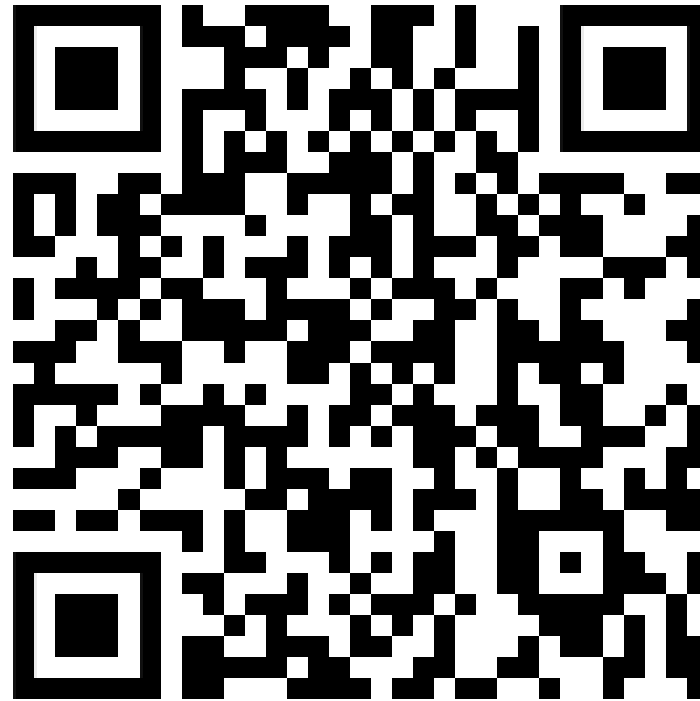


Fundamentos de MATLAB & Simulink

8 de marzo de 2023

Marcos Pesante

Código y Presentación (GitHub)

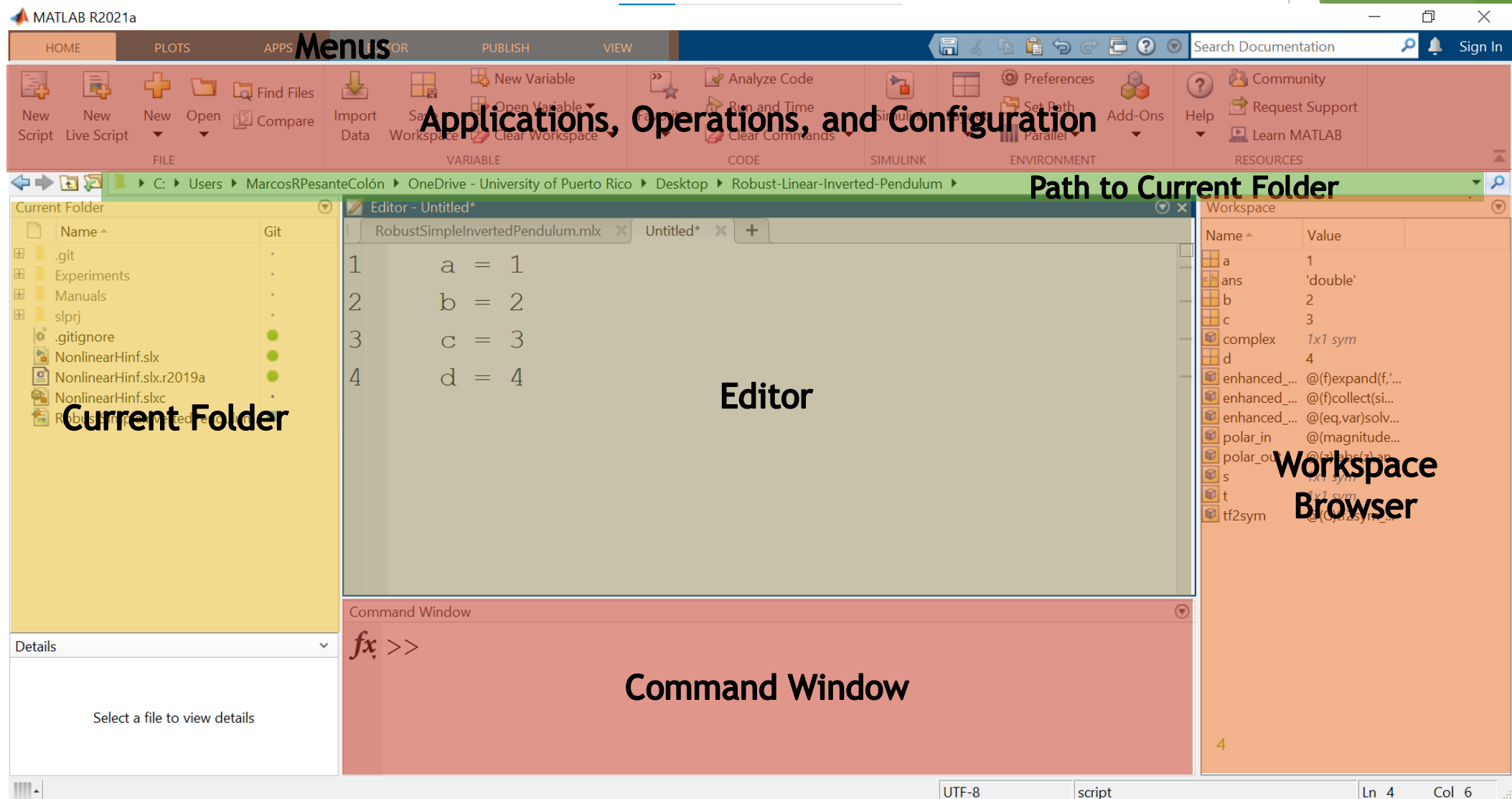


<https://github.com/M4rqu1705/Fundamentos-de-MATLAB-Simulink>

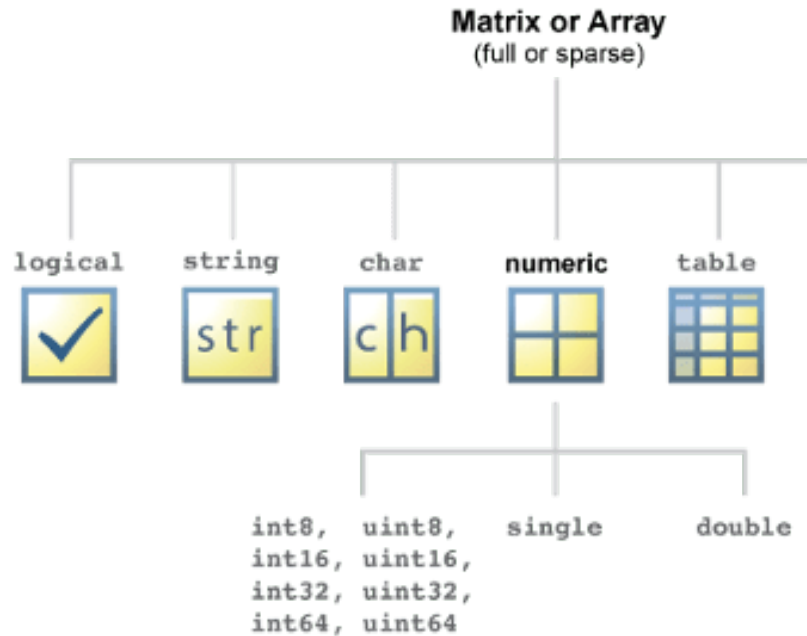
¿Qué es MATLAB?

- ▶ MATrix LABoratory
- ▶ Calculadora de matrices → Ambiente de:
 - ▶ Computación Numérica
 - ▶ Programación
 - ▶ Análisis de datos
 - ▶ Desarrollo de algoritmos
 - ▶ Creación de aplicaciones
 - ▶ Modelaje y Simulación
 - ▶ Más ...
- ▶ Aplicado a ciencia, economía e ingeniería
- ▶ Paquetes de herramientas al que llaman “toolboxes”

Elementos de la Interfaz Gráfica



Tipos de Datos



- ▶ Numéricos
 - ▶ Enteros, reales y números complejos
 - ▶ De tipo “double” por default.
- ▶ “Logical” (booleanos)
 - ▶ Valores ciertos (1) o falsos (0)
- ▶ Caracteres y Strings
 - ▶ Secuencias de caracteres
 - ▶ Utilizados para representar texto
- ▶ Matrices y Vectores

```
magic(3)
```

```
ans = 3x3
```

```
      8      1  
      3      5  
      4      9
```

```
ans(1,3)
```

```
ans = 6
```

Matrices y Vectores

- ▶ Creación de Matrices y Vectores
 - ▶ Manualmente
 - ▶ Funciones Especializadas
(zeros, ones,
eye, rand, ...)
 - ▶ Operador :
- ▶ Índices
 - ▶ Posición (fila, columna)
 - ▶ Índice individual
 - ▶ Valores lógicos
 - ▶ LOS ÍNDICES COMIENZAN EN 1

Operaciones Aritméticas

► Operaciones en orden de precedencia

1. Paréntesis ()

2. Transpuesta (.') y Exponentes (^)

► Para elevar una matriz A^n , A debe ser una matriz cuadrada y n un escalar.

3. Multiplicación (*) y División (/)

► Para multiplicar $A \times B$, A debe tener tantas columnas como B tiene filas

4. Suma (+) y Resta (-)

► Matrices deben tener mismas dimensiones (size)

► Para exponentes, multiplicaciones y divisiones, añadir un punto antes de cada operador permite llevar a cabo estas operaciones “element-wise” (a cada elemento individual de la matriz)

Seno	<u>sin</u>	<u>sind</u>	<u>asin</u>	<u>asind</u>		
Coseno	<u>cos</u>	<u>cosd</u>	<u>acos</u>	<u>acosd</u>		
Tangente	<u>tan</u>	<u>tand</u>	<u>atan</u>	<u>atand</u>	<u>atan2</u>	<u>atan2d</u>
Conversión Radianes/Grados	<u>deg2rad</u>			<u>rad2deg</u>		
Exponencial y Logaritmos	<u>exp</u>		<u>log</u>		<u>log10</u>	
Números Complejos	<u>abs</u>	<u>angle</u>	<u>conj</u>	<u>real</u>	<u>imag</u>	
Misceláneo	<u>sign</u>	<u>sqrt</u>	<u>nthroot</u>		<u>integral</u>	<u>diff</u>

Otros Operadores Útiles

¡Auxilio!

- ▶ Para repasar o investigar comandos, existen 2 alternativas principales:
 - ▶ ``help`` - Provee información resumida del comando y enlaces adicionales desde el Command Window
 - ▶ ``doc`` - Abre el navegador de ayuda, que además de proveer la misma información que ``help`` provee más detalles y ejemplos.
- ▶ Posibles alternativas para aprender sobre comandos son:
 - ▶ Search Engine - Utiliza funciones de búsqueda dentro del “help browser”
 - ▶ Foros de Mathworks Answers - La comunidad responde preguntas.
 - ▶ Google - No solo busca en los foros de MathWorks, pero además puede encontrar libros y páginas web de profesores que pueden clarificar dudas.

Higiene del Workspace

- ▶ `clc` - Remueve todo el texto del Command Window.
- ▶ `home` - Mueve el cursor del Command Window hacia arriba.
- ▶ `clear`
 - ▶ Borra variables del ambiente de trabajo
 - ▶ `clear all` borra TODAS las variables y funciones del ambiente de trabajo.
- ▶ `close`
 - ▶ Cierra figuras que estén abiertas
 - ▶ `close all` cierra TODAS las figuras del ambiente de trabajo.
- ▶ Los comentarios (`%`) permiten describir código sin afectarlo.
- ▶ Los puntos y comas (`;`) suprimen la salida de un comando.

Scripts

- ▶ Archivos donde se organizan secuencias de comandos de MATLAB.
- ▶ Corren todos los comandos o solo secciones en orden que aparecen.
- ▶ Scripts / M-Files
 - ▶ Archivo de texto más sencillo (y popular).
 - ▶ Ayuda a automatizar tareas o reducir la repetición en el Command Window.
- ▶ Live Scripts
 - ▶ Archivos multimedia con el mismo propósito que los m-files.
 - ▶ Permite incluir texto, ecuaciones, enlaces, imágenes y elementos interactivos.
 - ▶ Más fácil de usar y presenta los resultados de forma más visual y limpia.
- ▶ Forma más común de elaborar programas avanzados y compartirlos.

Variables

- ▶ Se utilizan para almacenar entradas, valores intermedios y salidas.
- ▶ Sus nombres comienzan con letras, seguidas por más letras, dígitos o underscores (_). Además, son sensibles a letras mayúsculas.
- ▶ Se pueden editar de múltiples formas:
 - ▶ Reasignar un valor nuevo a la variable existente.
 - ▶ Doble-click el valor en el Workspace Browser y cambiar.
 - ▶ Doble-click el nombre en el Workspace Browser y editar en Variable Editor.
- ▶ ¡LAS VARIABLES SE BORRAN AL CERRAR MATLAB!
 - ▶ Se puede mitigar esto creando scripts o guardándolas en un MAT-File.

Gráficas - Funciones

- ▶ Se pueden hacer gráficas de funciones en 2 y 3 dimensiones.
- ▶ Pasos generales para graficar funciones en MATLAB son:
 1. Preparar un dominio de valores (variables independientes)
 2. Preparar el resultado (variable dependiente) en función del dominio.
 3. Graficar utilizando el comando apropiado.
 4. Añadir título y leyenda y etiquetar ejes
- ▶ ``figure`` - Crea y configura múltiples ventanas separadas.
- ▶ ``subplot`` - Hace más de una gráfica dentro de una ventana.
- ▶ ``hold`` - Grafica múltiples funciones compartiendo los mismos ejes.
- ▶ ``grid`` - Añade un cuadriculado de referencia para los ejes.

Gráficas - Otros Comandos Útiles

► 2 Dimensiones

- ``plot``
- ``area``
- ``loglog``
- ``semilogx``,
``semilogy``
- ``histogram``
- ``scatter``
- ``bar``
- ``polarplot``
- ``pie``

► 3 Dimensiones

- ``plot3``
- ``surf``
- ``contour``
- ``heatmap``
- ``histogram2``
- ``scatter3``
- ``bar3``
- ``quiver3``
- ``streamline``

► Atributos de figuras

- ``xlim``
- ``ylim``
- ``zlim``
- ``axis``
- ``box``
- ``xtickangle``

Estructuras de Control: If

- ▶ Añade “inteligencia” al programa al condicionalmente correr comandos.
- ▶ Las condiciones deben poder ser evaluadas como valores lógicos.
 - ▶ Operadores Relacionales
 - ▶ Operadores Lógicos
- ▶ Las indentaciones ayudan a identificar qué está dentro del `if` y qué no. No son obligatorias.

```
if a == 1
    % a es exactamente 1
elseif a > 1
    % a no es 1 y es
    % mayor que 1
else
    % a no cumple con las
    % condiciones anteriores
end
```

Estructuras de Control: For Loops

- ▶ Los “loops” en general reducen la repetición de código.
- ▶ Hacen posible crear algoritmos más avanzados.
- ▶ `for` loops se especializan en:
 - ▶ Repetir un proceso una cantidad determinada de veces.
 - ▶ Visitar los elementos de una matriz.
 - ▶ Rastrear un índice de iteración para cualquiera de los anteriores.

```
a = [];  
for i = 1:10  
    a = [a, i^2];  
end  
a
```

```
a = 1×10  
    1     4     9    16 ...
```

```
for i = 1:2:length(a)  
    a(i) = 0;  
end  
a
```

```
a = 1×10  
     0     4     0    16 ...
```


Estructuras de Control: While Loops

- ▶ Los `while` loops corren solo si una condición lógica es cierta.
- ▶ A diferencia del `if`, repetirá el código hasta que su condición sea falsa.
- ▶ “while loops” se especializan en casos donde la cantidad de iteraciones no es conocida.
 - ▶ De no garantizar que su condición sea falsa eventualmente, el “loop” podría correr eternamente.

```
num = 65;  
cuenta = 0;  
while num > 1  
    num = num / 10;  
    cuenta = cuenta + 1;  
end  
cuenta
```

cuenta = 2

BREAK (10 mins)

Proyecto MATLAB

1. Descargar los datos del repositorio de GitHub. Extraerlos a algún folder accesible.
2. Cambiar el “Path to Current Folder” al folder con los datos.
3. Cargar los datos de la estación meteorológica en ``weather_data.mat`` usando el comando de ``load``.
 - De otra forma, se puede doble-click el documento en el “Current Folder”
4. Graficar los datos.
5. Utilizar los conceptos aprendidos de “loops” y condicionales para remover datos defectuosos.

Proyecto MATLAB: Retos Adicionales

1. Utilizar índices y operaciones con matrices para hacer lo mismo en menos líneas de código.
2. Repetir el mismo procedimiento del proyecto, pero con datos de consumo.
 - ▶ Existen datos donde la potencia aparente no concuerda con los cálculos de la potencia real y la reactiva.
 - ▶ Existen datos donde la potencia real es negativa. ¿Por qué?

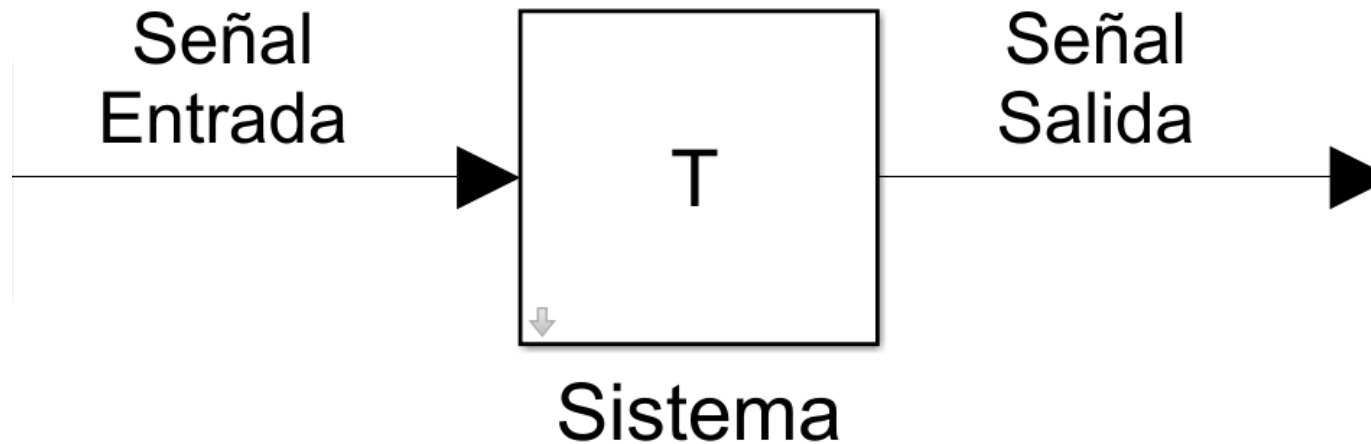
Simulink

¿Qué es Simulink?

- ▶ Ambiente de diagramas de bloques.
- ▶ Utilizado para
 - ▶ Modelaje, simulación y análisis de sistemas
 - ▶ Diseño a base de modelos
 - ▶ Desarrollo y pruebas continuas
 - ▶ Implementación de programas en sistemas embebidos
 - ▶ Integración con MATLAB
- ▶ Complementa a MATLAB de una forma más visual.
- ▶ Hace funcionalidades de MATLAB más accesibles para principiantes.

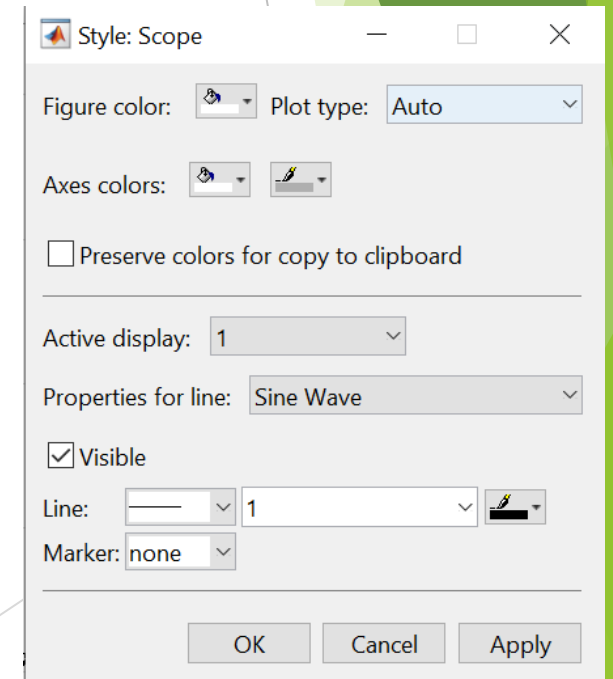
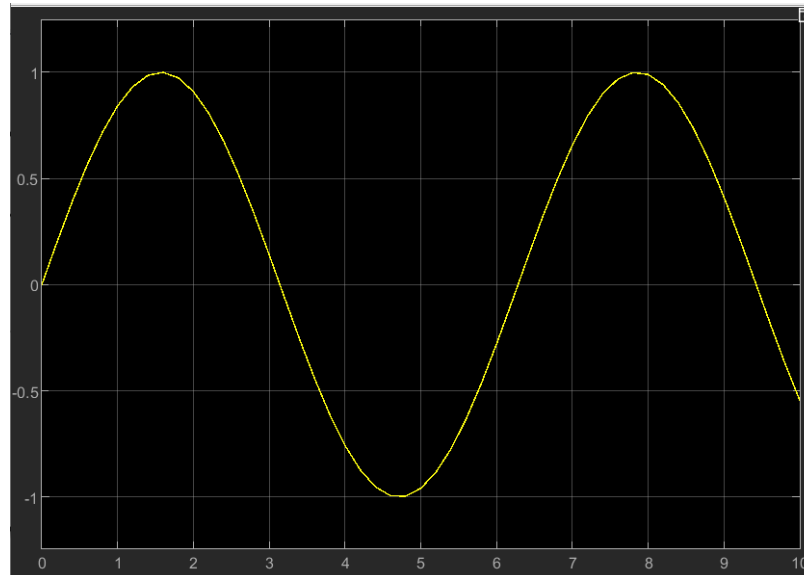
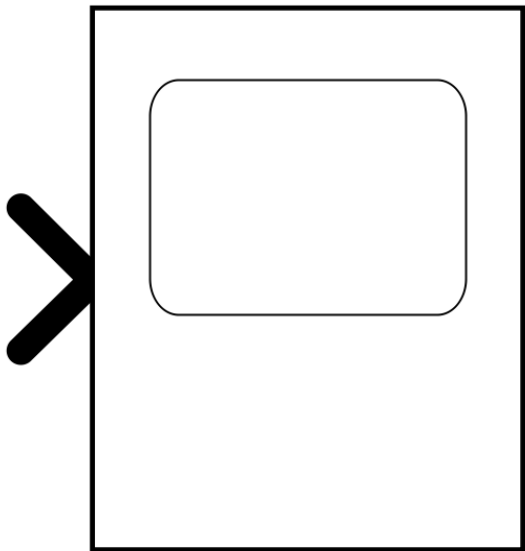
Diagramas de Bloques

- ▶ Cada línea representa una señal
- ▶ Cada bloque representa un sistema o un componente del mismo.
- ▶ Los sistemas y operaciones transforman las señales



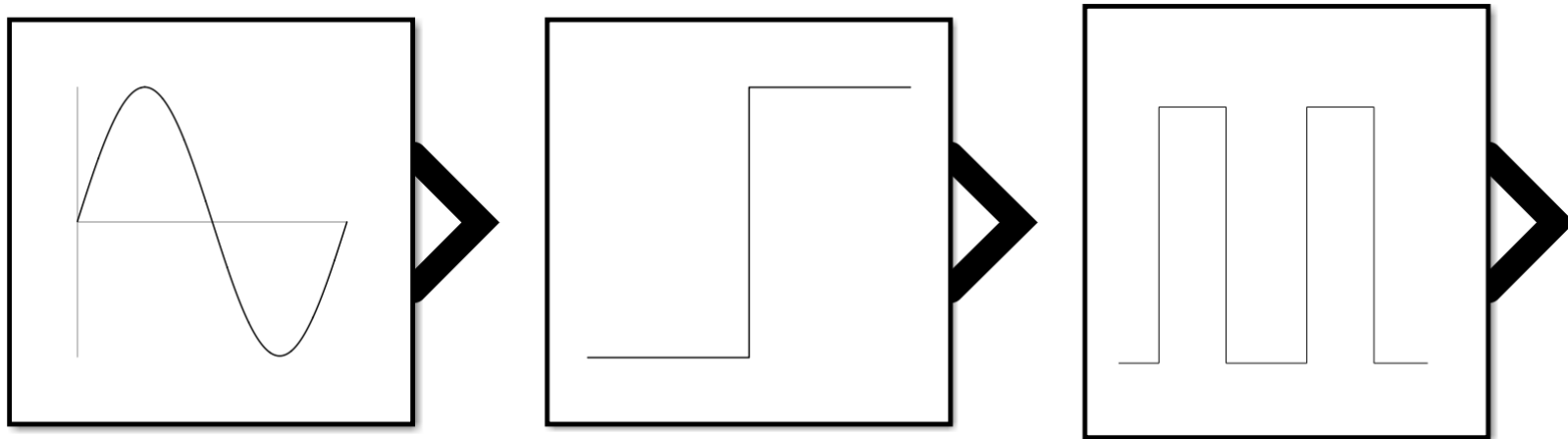
Scope

- ▶ Grafica señales que recibe mediante su puerto de entrada.
- ▶ Capaz de modificar el tipo de gráfica, ejes, y esquema de colores.
- ▶ Muy parecido a las figuras en sus funcionalidades.



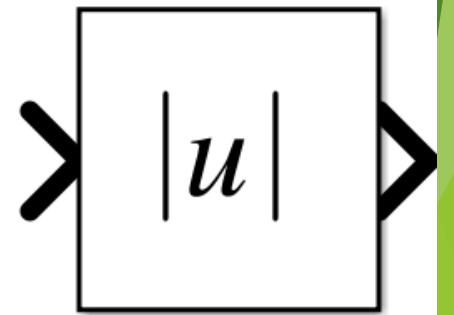
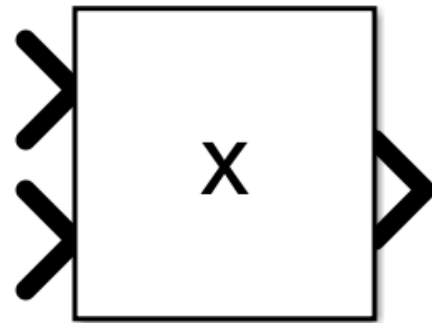
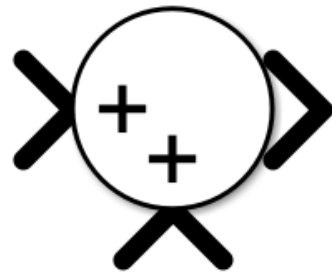
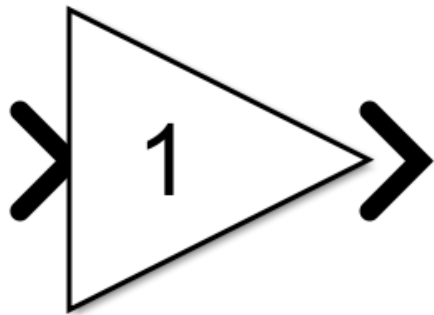
Fuentes

- ▶ Proveen entradas para la simulación
- ▶ Existe una variedad de señales dentro de Simulink → Sources
- ▶ Son suficientemente flexibles para configurar sus propiedades o hasta crear nuevas señales.



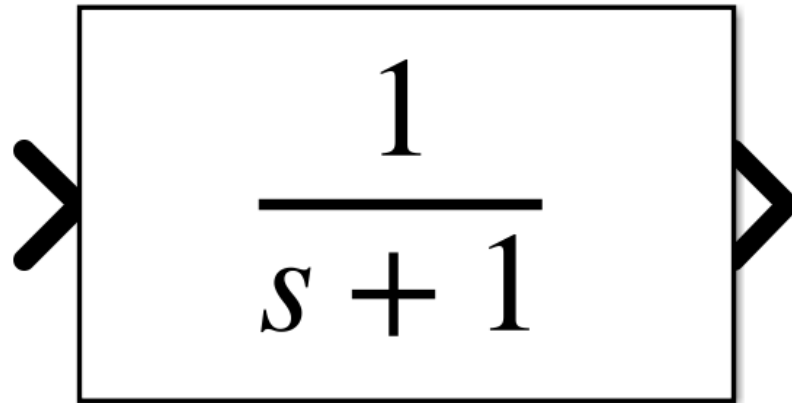
Operaciones Matemáticas

- ▶ Incluyen las operaciones matemáticas discutidas para MATLAB
 - ▶ Gain: Multiplica una señal
 - ▶ Sum: Suma múltiples señales
 - ▶ Product: Multiplica señales escalares o matrices
 - ▶ Absolute Value: Calcula el valor absoluto de su señal de entrada.



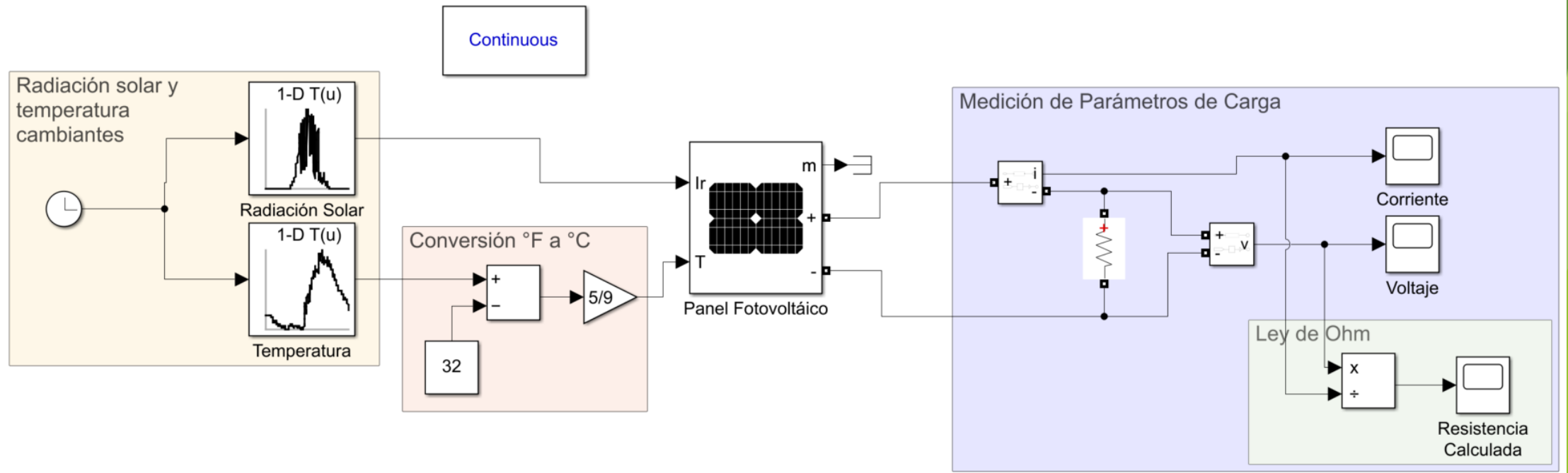
Función de Transferencia

- ▶ Transformada de Laplace de la salida a la entrada del sistema (presumiendo condiciones iniciales iguales a 0)
- ▶ Útil para abstraer detalles del comportamiento de circuitos y sistemas LTI a base de sus entradas y sus salidas.



Proyecto

- Utilizar SimScape y el Specialized Power Systems Toolbox para simular un sistema fotovoltaico y una carga eléctrica resistiva.
- Corroborar el comportamiento dictado por la Ley de Ohm.



Toolboxes y Referencias Adicionales

- ▶ Symbolic Toolbox
- ▶ Control Systems Toolbox
- ▶ SimScape
- ▶ Specialized Power Systems Toolbox
- ▶ MATLAB Coder Toolbox
- ▶ Deep Learning Toolbox