

Métodos Numéricos y de Simulación

TEMA 2. Introducción a SIMULINK



Departamento de Electrónica y Electromagnetismo
Universidad de Sevilla

Indice

- ¿Qué es la Simulación?
- ¿Qué es Simulink?
- Cómo crear un modelo
- Cómo ejecutar la simulación de un modelo
- Ejemplos
- Matlab y Simulink
- Tiempo y programación en Simulink

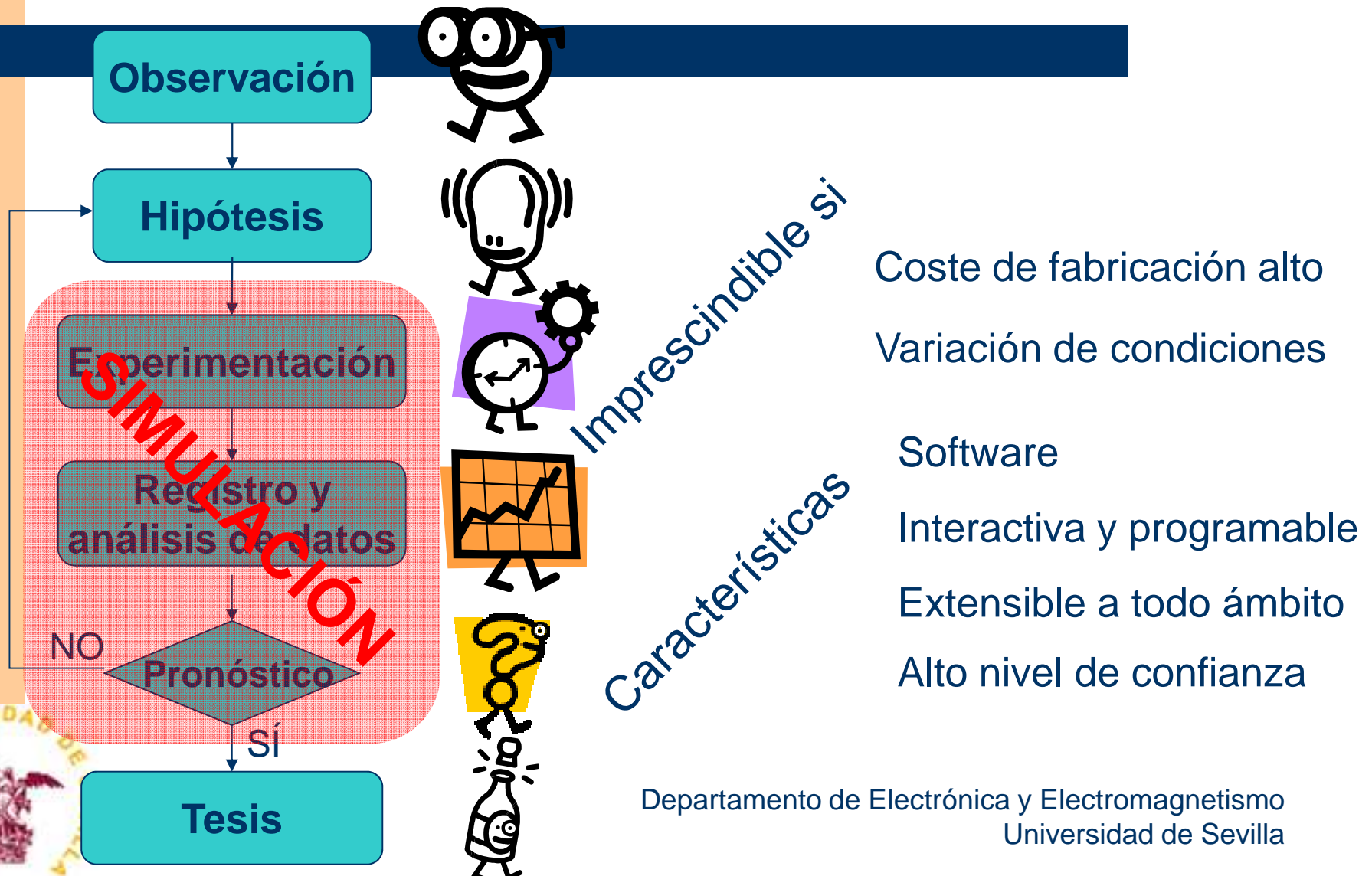


Definición de Simulación

- Mecanismo que permite **reproducir el comportamiento** de un modelo sin necesidad de experimentación
- Proceso para **diseñar y desarrollar un modelo** computerizado de un sistema o proceso y **conducir experimentos** con este modelo con el propósito de entender el comportamiento del sistema o evaluar varias estrategias con las cuales se puede operar el sistema



Simulación en el Método Científico



Fases de la simulación

- Definición del sistema
- Formulación del modelo
- Colección de datos
- Implementación del modelo en el ordenador
- Verificación
- Validación
- Experimentación
- Interpretación
- Documentación

MODELO



Ventajas de la simulación

Mejor comprensión
del sistema

Mucho
software
disponible

Entrenamiento

Alternativo a montaje
experimental y a
análisis teórico

Efecto de cambios
en el sistema

BAJO
COSTE



Inconvenientes de la simulación

- Resultados de precisión dudosa (dependientes del modelo)
- Nos puede alejar de la realidad
- Crea dependencia, según las aplicaciones

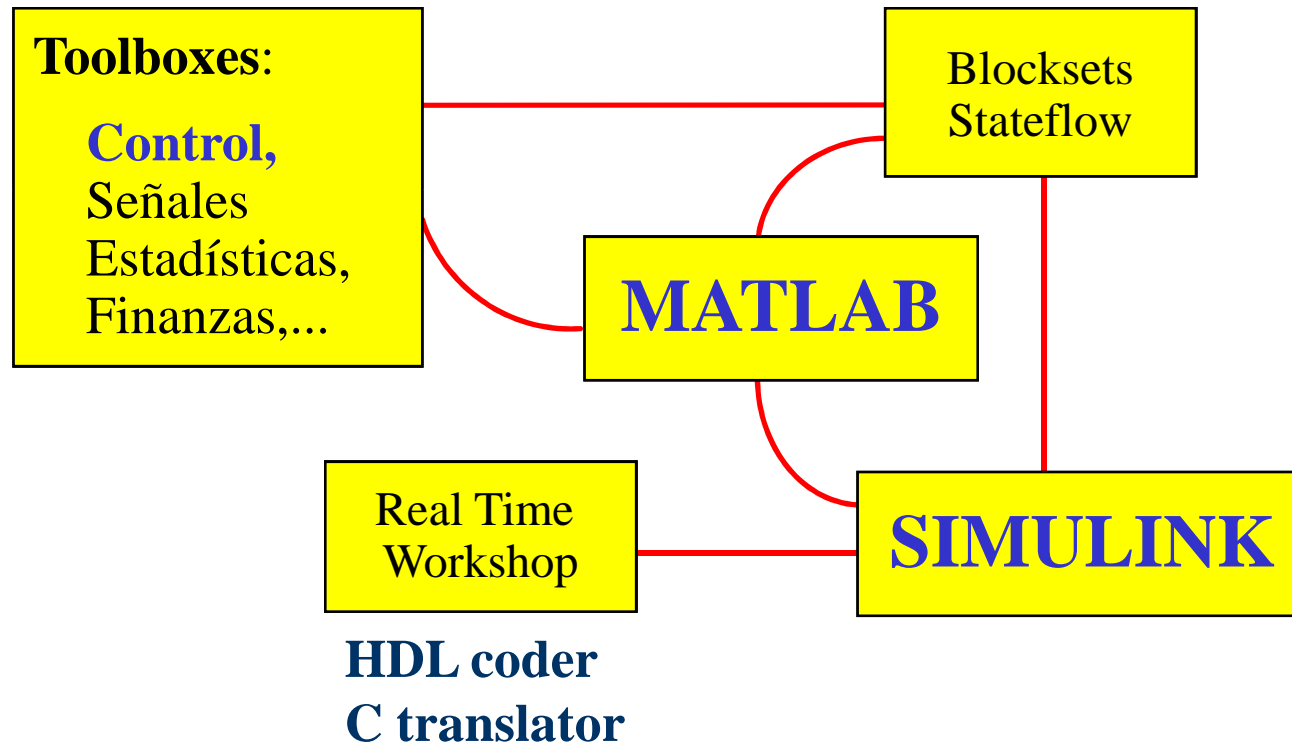


¿ Qué es Simulink ?

- Es una herramienta interactiva
- Sirve para modelar y analizar sistemas dinámicos
- Basada en diagramas de bloques
- Acoplada con MATLAB



Estructura de SIMULINK



Pasos para el uso de Simulink

- Definición de un modelo o representación matemática
- Definición de los parámetros del sistema
- Elección de un método de integración apropiado
- Ajuste de las condiciones de ejecución de la simulación



Recomendaciones previas

- Tener una descripción matemática del modelo
- Las ecuaciones deben ser manipuladas para eliminar posibles lazos algebraicos
- Tener conocimiento de cuáles variables son independientes y cuáles son dependientes
- Reescribir las ecuaciones integrales con las variables de estado dependientes expresadas como alguna integral de una combinación de variables independientes y variables dependientes



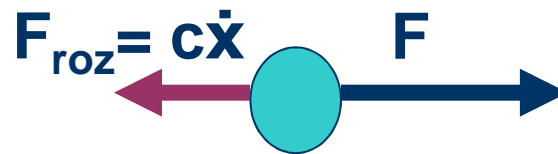
Crear un Modelo

- Una **teoría** es una declaración de un principio abstracto de una observación
- Un **modelo** es una representación de una teoría que puede ser usada para predicción, control, etc.
- Un modelo puede ser **real y no simple** para ser entendido y fácilmente manipulado
- El modelado consiste en un **proceso de análisis y síntesis** para encontrar una descripción matemática conveniente que abarque las características dinámicas relevantes de los componentes, preferiblemente en términos de parámetros que puedan ser determinados en la práctica



Ejemplo simple

- Posición (x) de una masa (m), sometida a una Fuerza (F) y que se ve sometida a una fuerza de rozamiento proporcional a su velocidad (\dot{x})
- $F - c\dot{x} = m\ddot{x}$
- Lineal, dinámico, determinista, continuo,... si masa puntual



Cómo crear un modelo

- Edición de un modelo
 - Abrir la ventana de un nuevo modelo
 - Añadir bloques
 - Conectar los bloques
 - Parametrizar los bloques
 - Cambiar el tamaño de los bloques, modificar etiquetas, añadir anotaciones...
- Guardar un modelo (formato .slx)
- Abrir un modelo desde Matlab

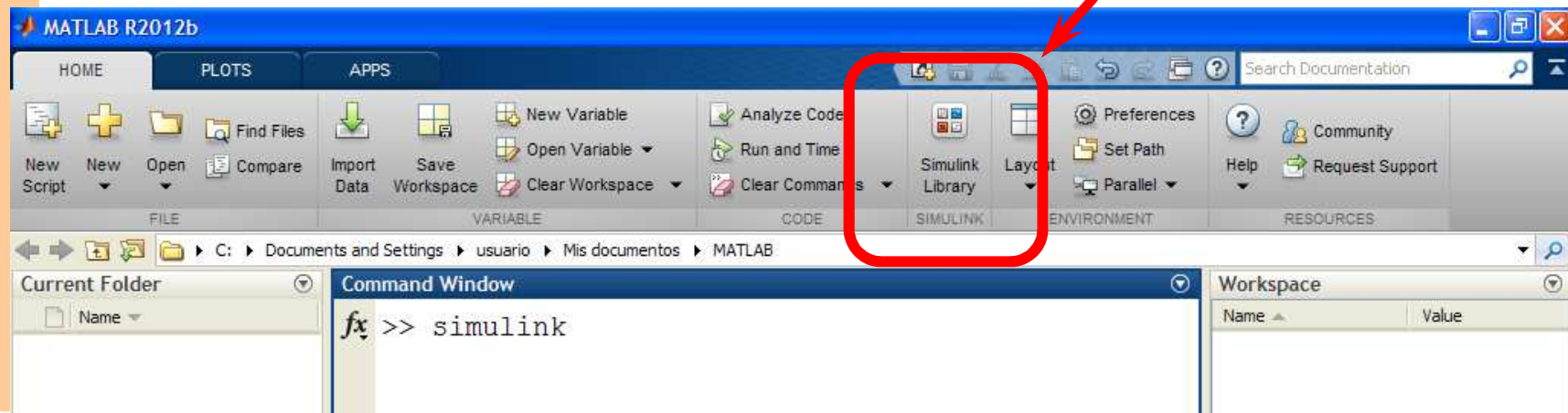


Cómo cargar SIMULINK

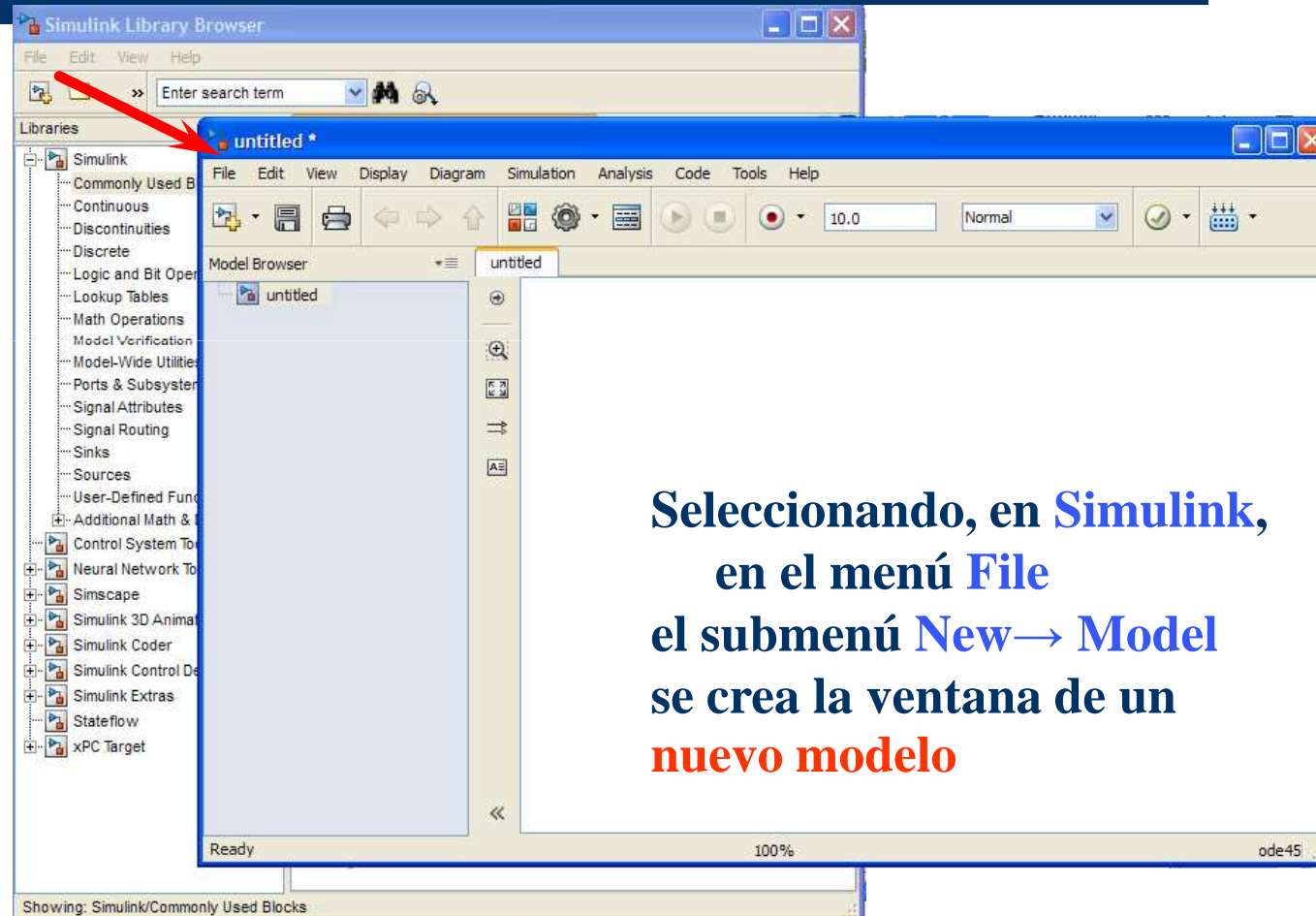
Escribiendo **simulink**
se inicia el programa.

» **simulink**

Activando en la
ventana de MATLAB
el icono

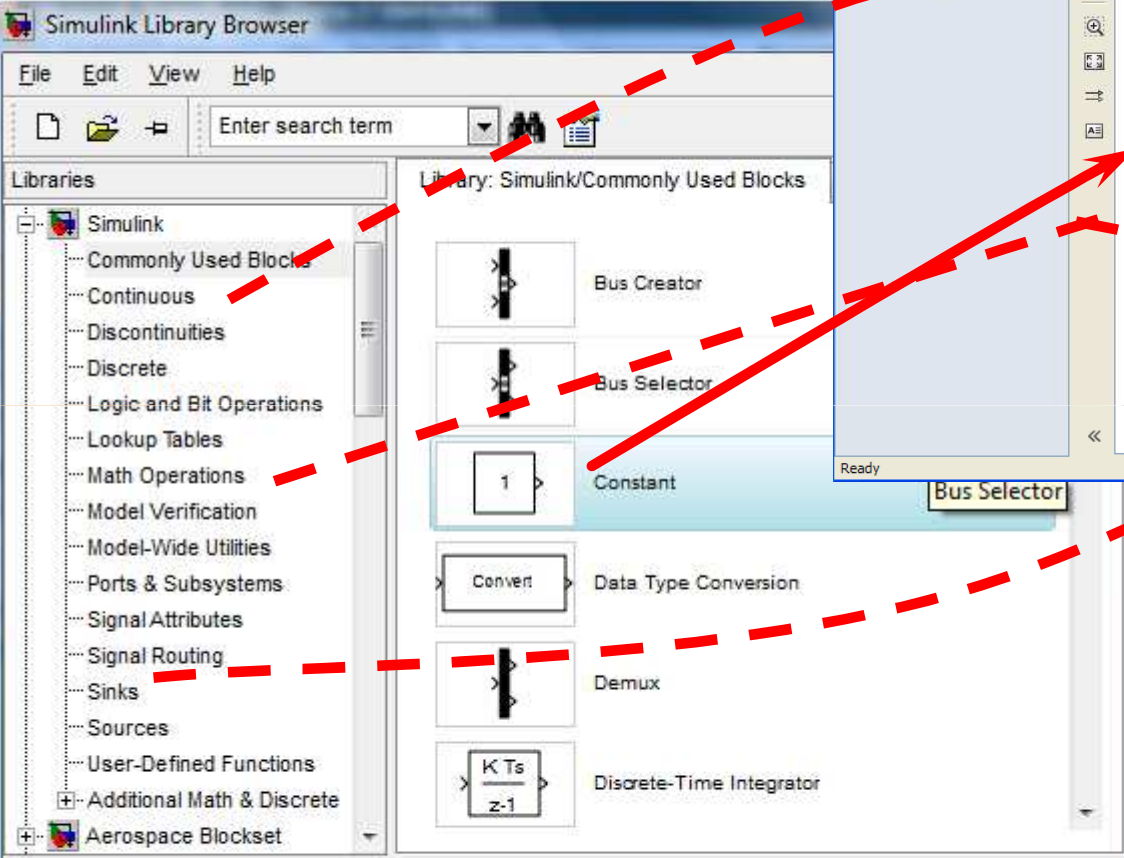


Abrir la ventana de un nuevo modelo



Seleccionando, en **Simulink**,
en el menú **File**
el submenú **New** → **Model**
se crea la ventana de un
nuevo modelo

Añadir bloques



Simulink Library Browser

File Edit View Help

Enter search term

Libraries

- Simulink
 - Commonly Used Blocks
 - Continuous
 - Discontinuities
 - Discrete
 - Logic and Bit Operations
 - Lookup Tables
 - Math Operations
 - Model Verification
 - Model-Wide Utilities
 - Ports & Subsystems
 - Signal Attributes
 - Signal Routing
 - Sinks
 - Sources
 - User-Defined Functions
 - Additional Math & Discrete
 - Aerospace Blockset

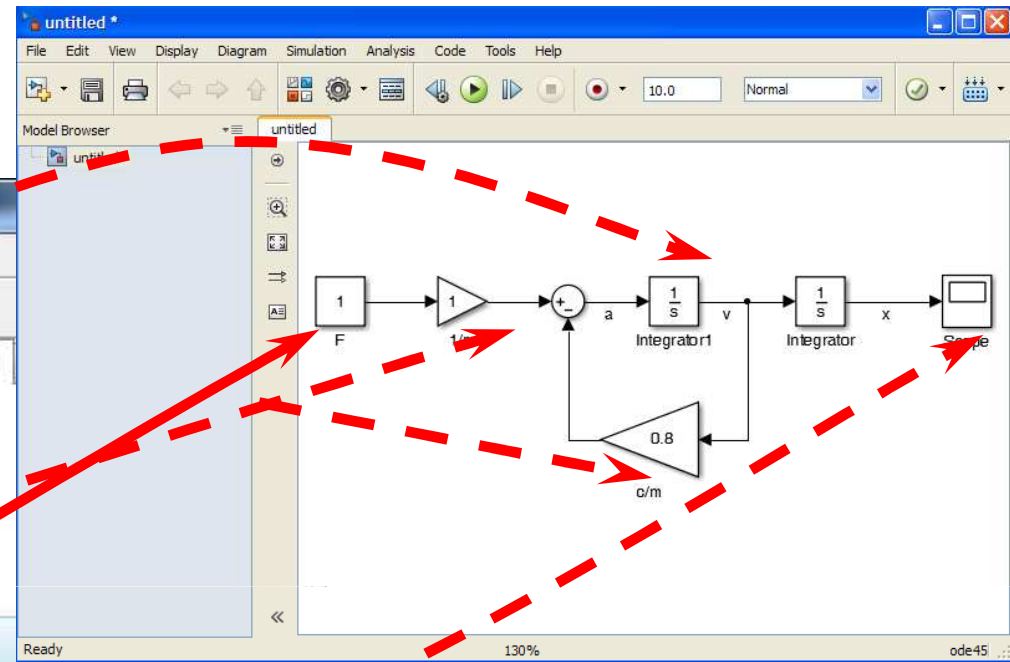
Library: Simulink/Commonly Used Blocks

- Bus Creator
- Bus Selector
- Constant
- Convert Data Type Conversion
- Demux
- Discrete-Time Integrator

Block Description

Constant: Output the constant specified by the 'Constant value' parameter. If 'Constant value' is a vector and 'Interpret vector parameters as 1-D' is on, treat the constant value as a 1-D array. Otherwise, output a matrix with the same dimensions as the constant value.

Showing: Simulink/Commonly Used Blocks



Arrastrar los bloques a la ventana del modelo

Modificar tamaño, nombre, rotar, hacer anotaciones...

Parametrizarlos (bot.izdo.2)

Descripción del bloque

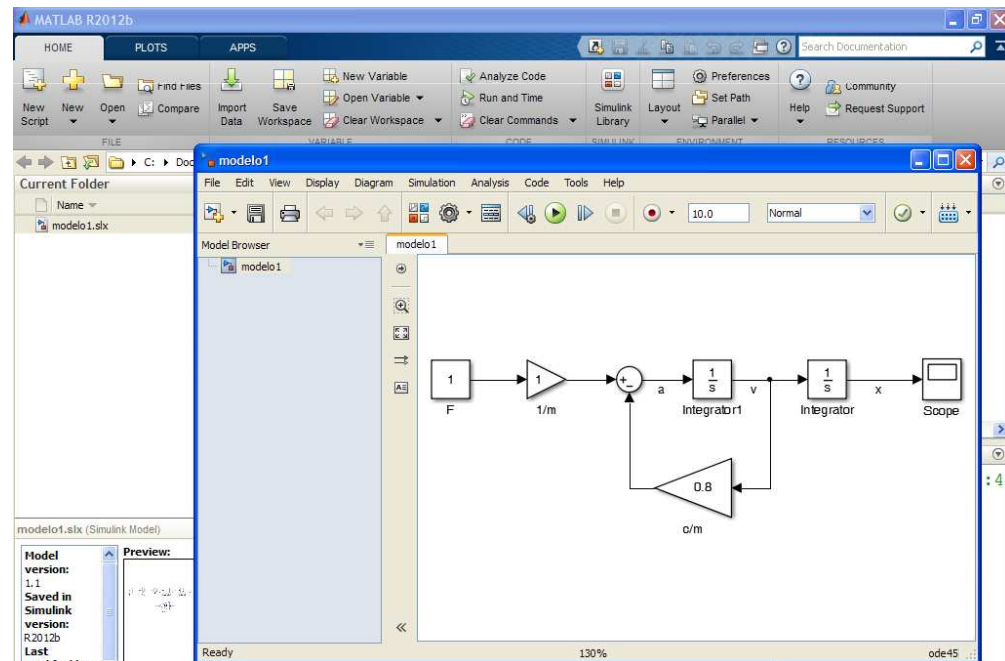
Departamento de Electrónica y Electromagnetismo
Universidad de Sevilla

Guardar el modelo

Save as: Crea el fichero del modelo: `modelo1.slx`

Abrir el modelo desde Matlab: `modelo1` en línea de comando

Simularlo con Simulink



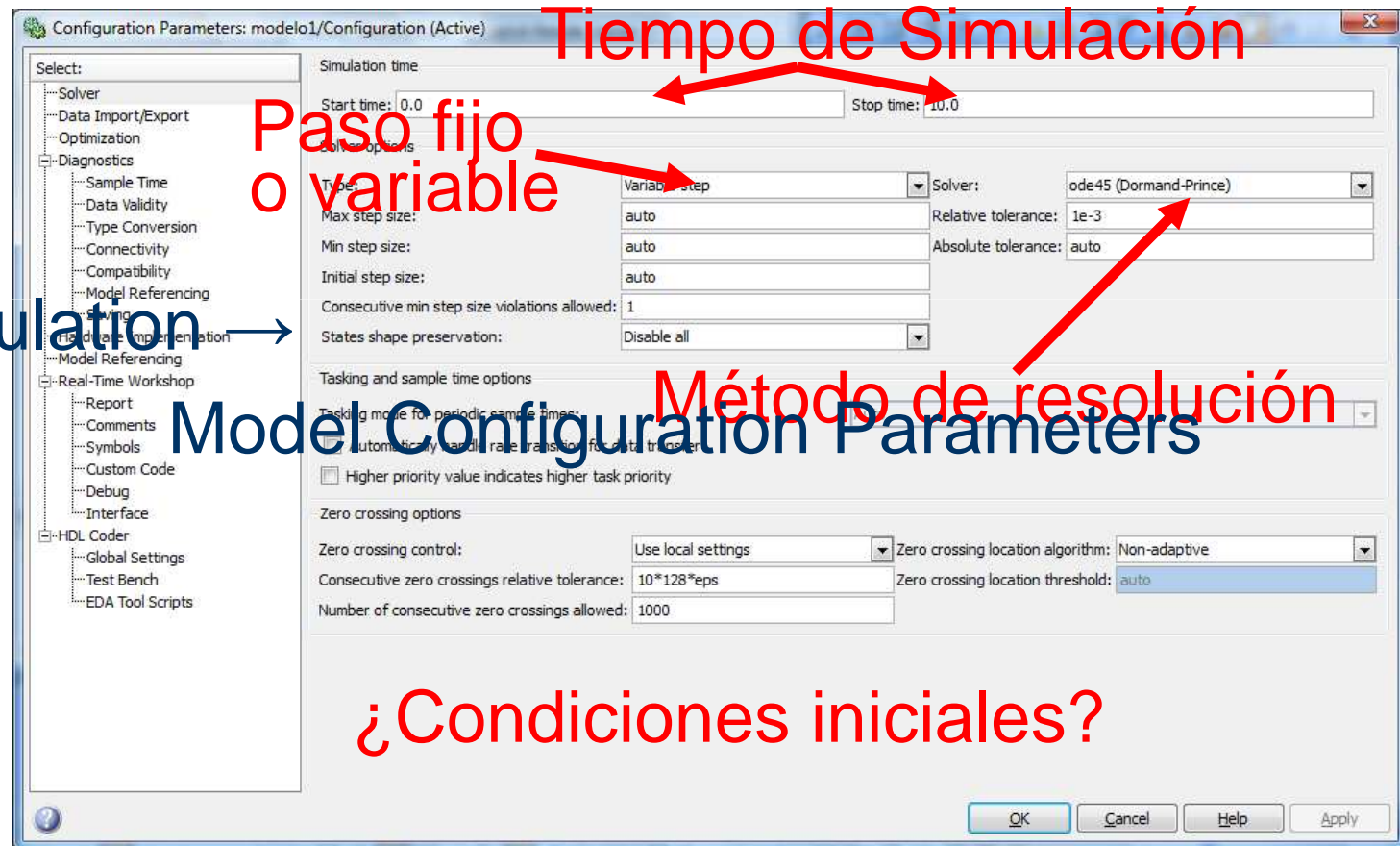
Ejecutar una simulación

- Poner los parámetros de la simulación
- Ejecutar una simulación desde la ventana del modelo
- Poner y sacar valores en/desde los modelos
 - Utilizar en Matlab los valores obtenidos en la simulación
 - Variables definidas en Matlab y Simulink
- Simular desde la línea de comandos



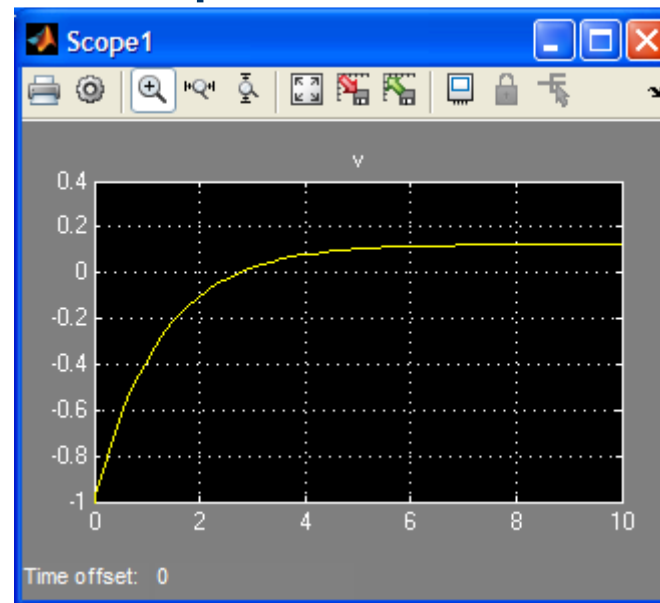
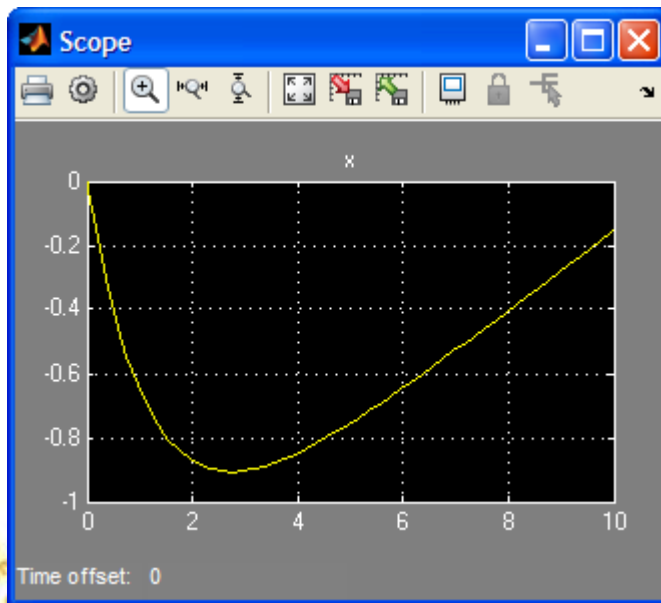
Fijar los parámetros de la simulación

- Simulation →



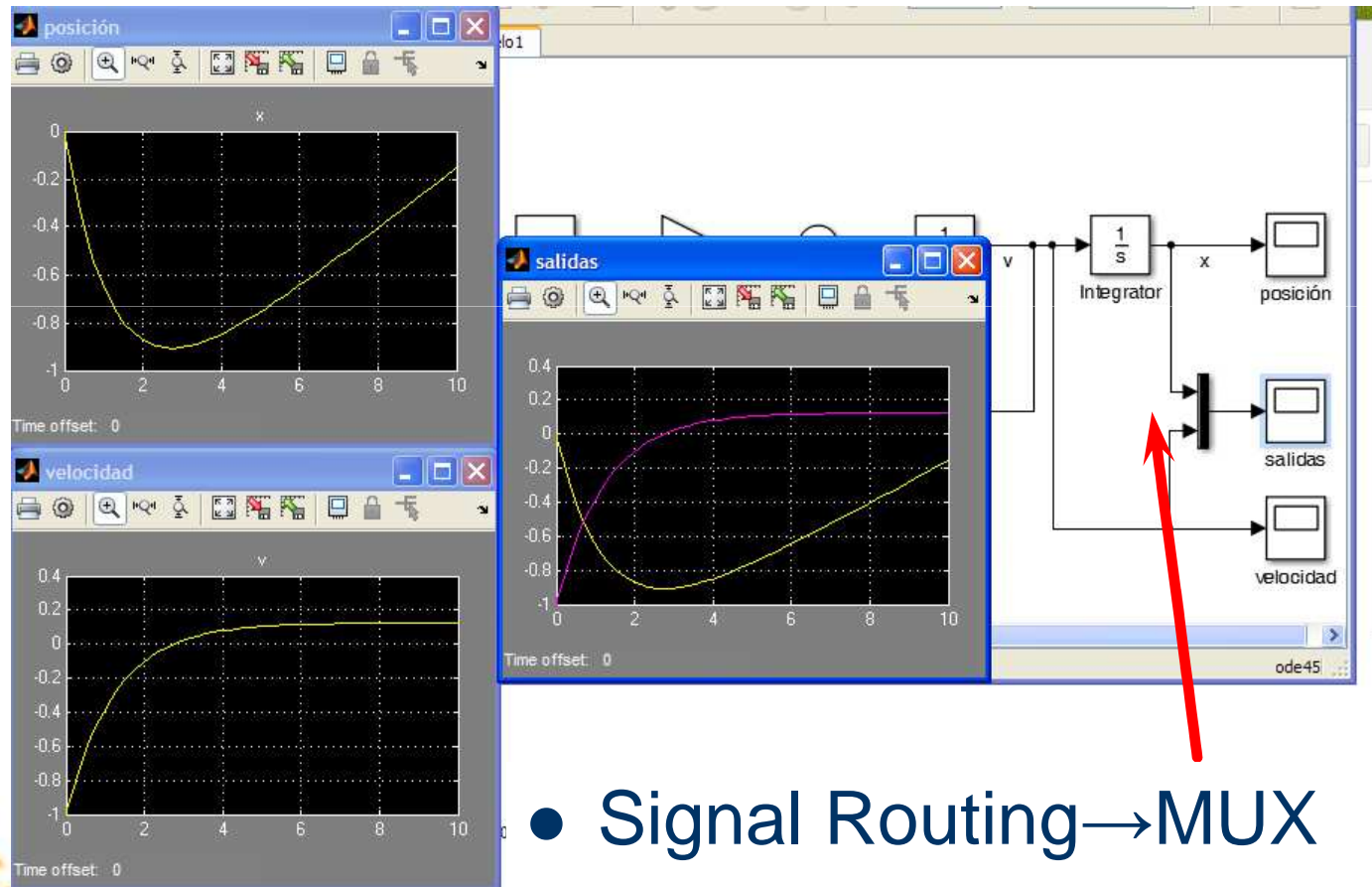
Casos simulados (I)

- Considerar $F=0.1\text{ N}$, $m=1\text{ Kg}$, $c=0.8\text{ Kg/s}$,
- Condición inicial, $x=0$, $\dot{x}=-1\text{ m/s}$
- Parámetros de simulación por defecto



Departamento de Electrónica y Electromagnetismo
Universidad de Sevilla

Casos simulados (I)



- Signal Routing → MUX

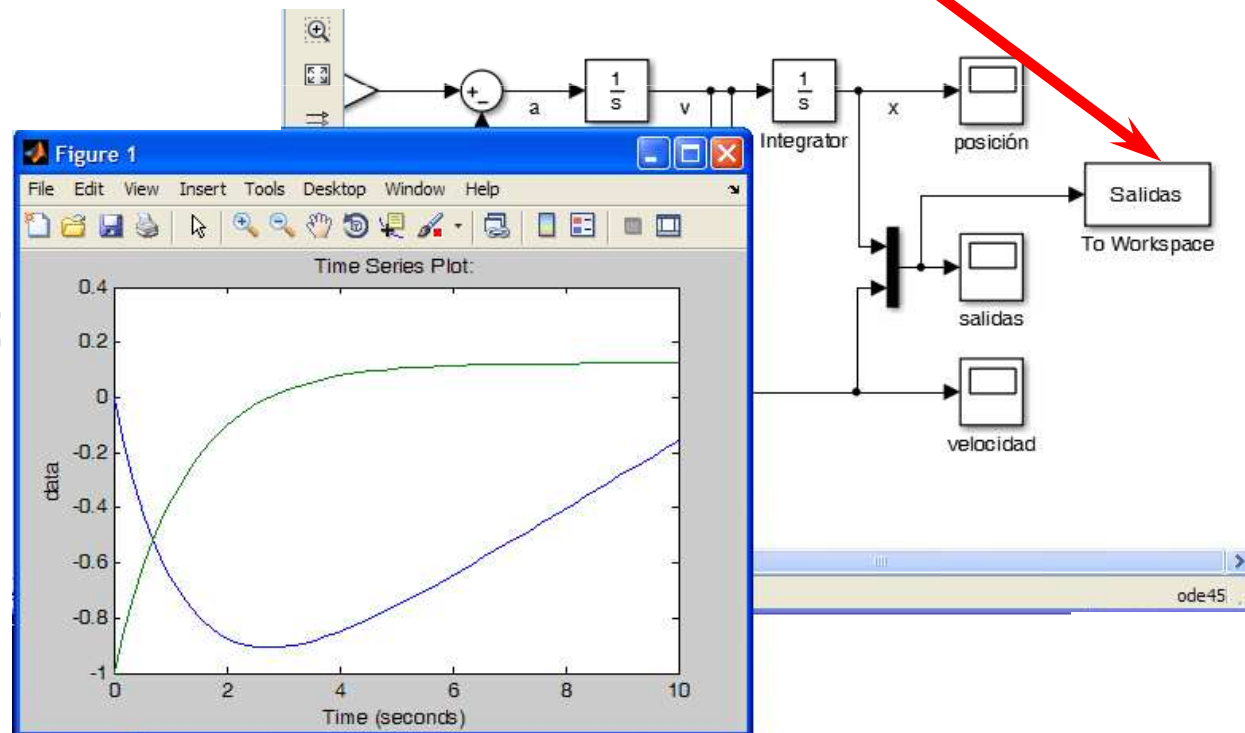
Casos simulados (I)

- Pasar la información a MATLAB

Sim → Model Conf Param → data Import/Export

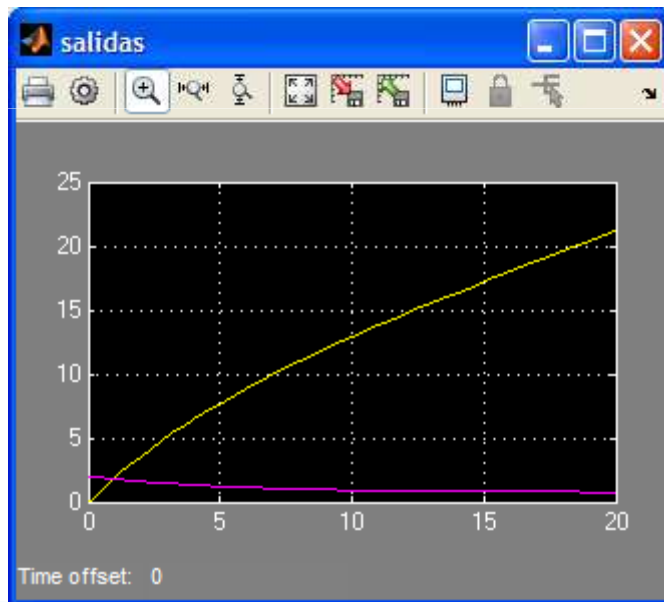
>whos

>plot (Salidas);

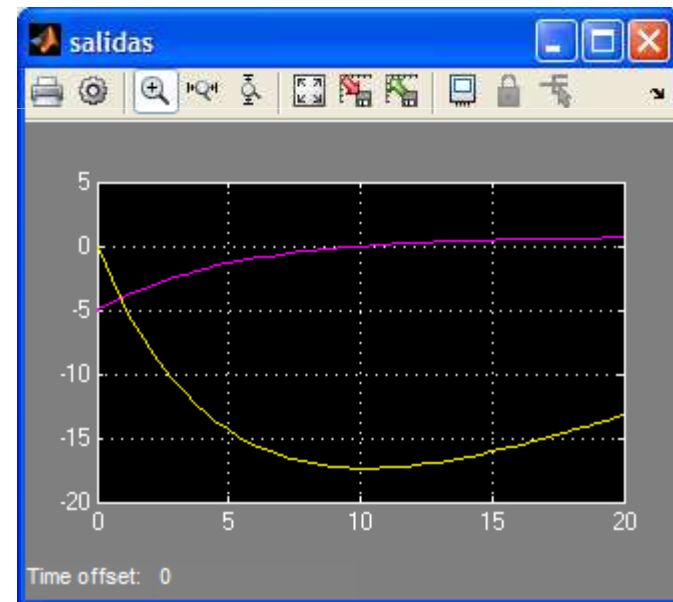


Casos simulados (II)

- Considerar $F=0.3\text{N}$, $m=2\text{Kg}$, $c=0.4\text{Kg/s}$,
- Tiempo de simulación: 20s



$$x=0, \dot{x}=2 \text{ m/s}$$



$$x=0, \dot{x}=-5 \text{ m/s}$$

Métodos de integración

- **ode45**: Método basado en Dormand-Prince , un paso Runge-Kutta y es recomendado como un primer método
- **ode23**: Método basado en Bogacki-Shampine, un paso Runge-Kutta. Más eficiente que ode45 cuando la tolerancia es amplia. Variantes s, t y tb
- **ode113**: Método multipaso, de orden variable Adams-Bashforth-Moulton. Recomendable cuando la función evaluación consume tiempo y la tolerancia es poca
- **ode15s**: Método multipaso, de orden variable basado en la fórmula de diferenciación “backward”



Parámetros de control relevantes

- **tolerance:** Usado para la rutina de integración para controlar la cantidad de error relativo a cada paso
- **minimum step size:** Usado para iniciar y reiniciar la integración al comienzo de una ejecución y después de una discontinuidad
- **maximum step size:** Límite de la longitud del paso para encontrar una apariencia suave en el plot de salida



Crear subsistemas

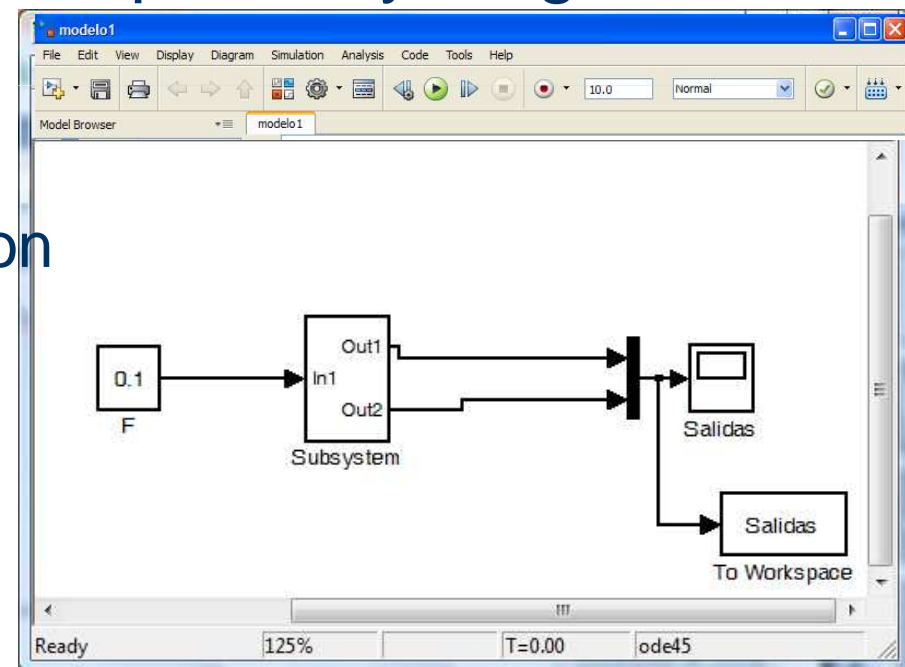
- Nivel de entrada-salida
- Selección con el bot. izq. ratón y luego:

Diagram →

Susbs&ModelRef →

create from selection

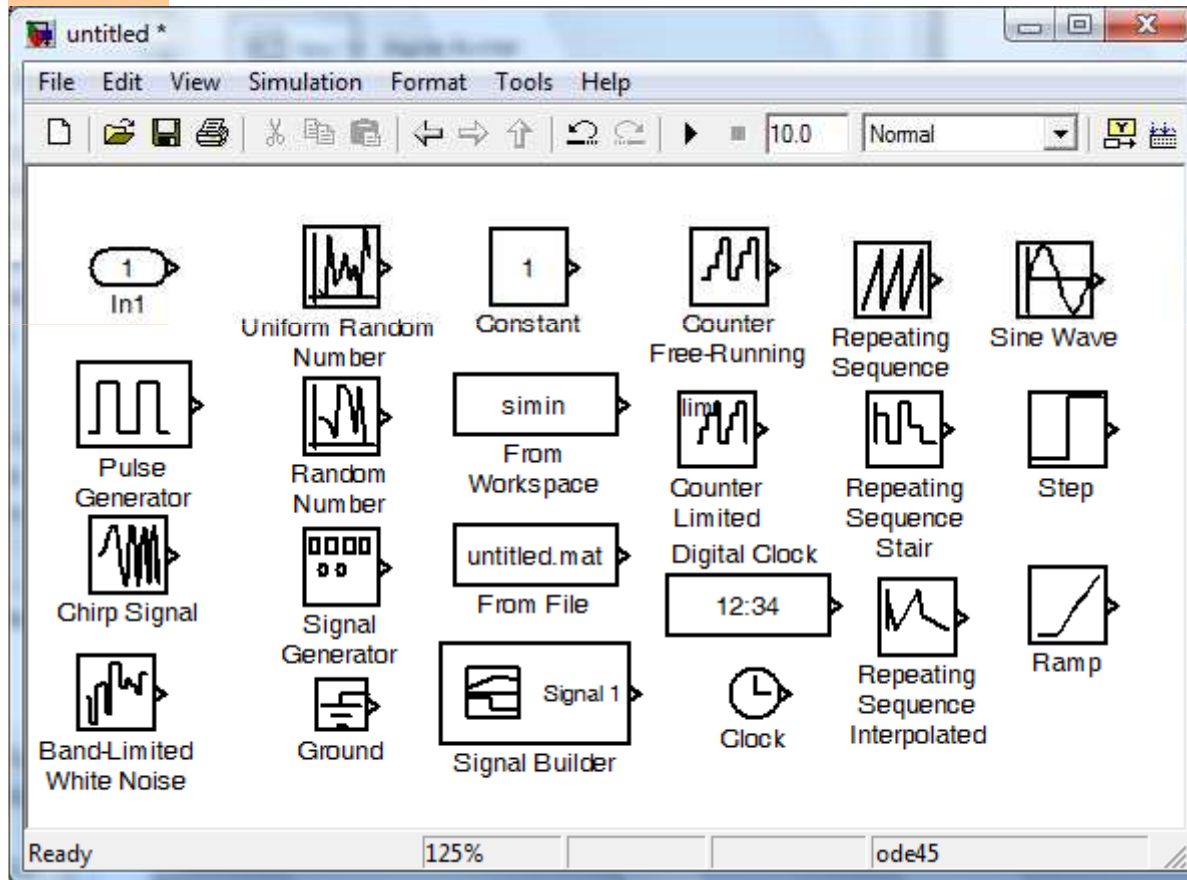
Se puede invocar el
subsistema con bot.izq.2



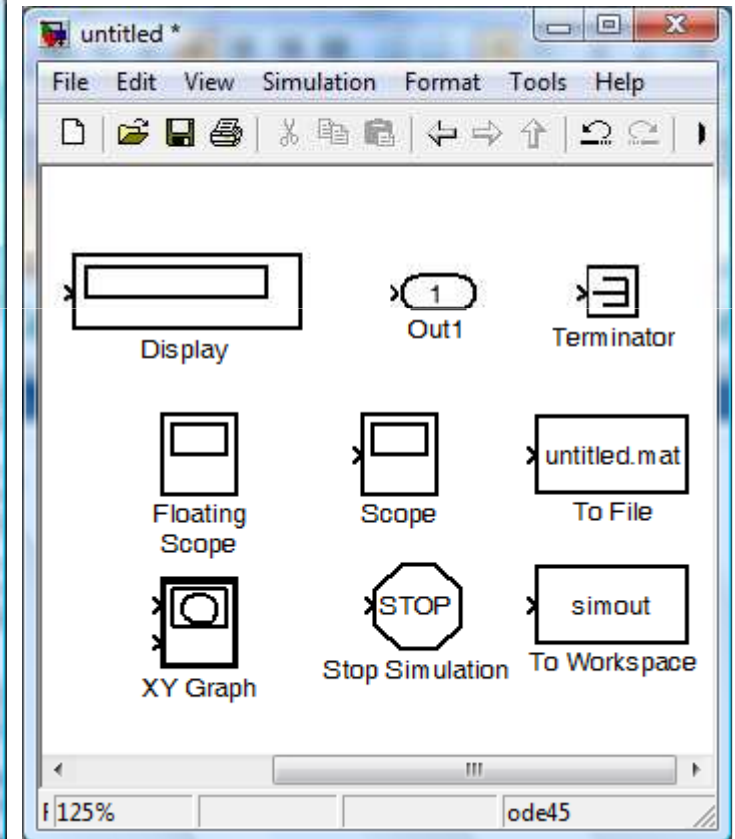
Departamento de Electrónica y Electromagnetismo
Universidad de Sevilla



Cómo introducir/observar información



Sources



Sinks

Transferencia datos Matlab-Simulink

- Matlab y Simulink comparten el Workspace
 - Definición de variables y parámetros
 - Uso de From/To WorkSpace
 - Uso de Signal Properties y Data Import/Export
- Escritura en/desde ficheros .mat
- Escritura en un fichero de matlab de las variables/datos



Ejecución de Simulink desde Matlab

- Uso de conjunto específico de comandos (ver la guía “Simulink como comandos de Matlab”). Destacan:

```
>> sim % se lanza la simulación
```

```
>> [t, x, y] = ode45 (modelo, lista de parámetros, ...) % se simula  
un modelo con parámetros. El tiempo t, el estado x y la salida y son  
vectores retornados por la simulación
```

```
>> simplot % se muestran los resultados
```

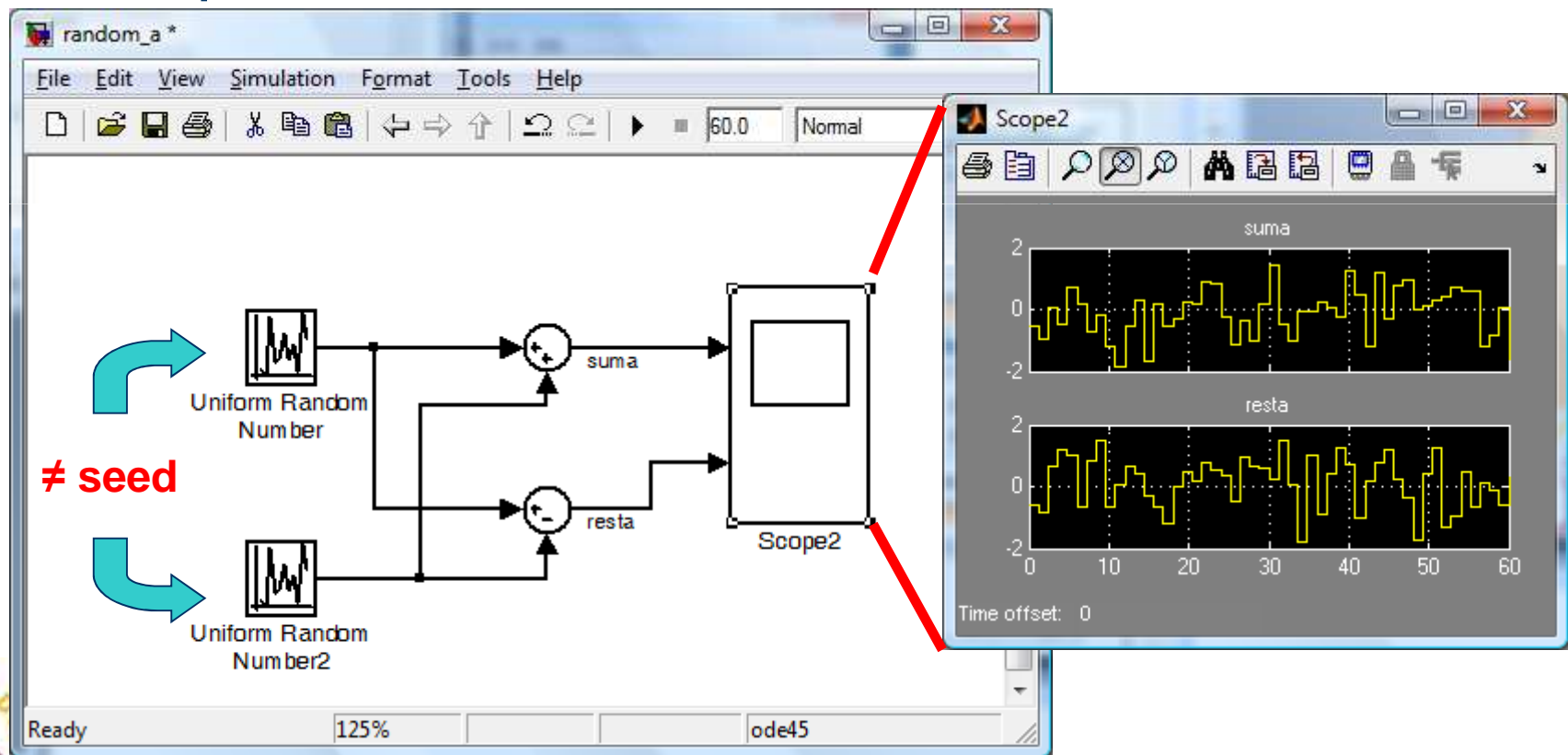
```
>> set_param % se configuran los parámetros
```

- Se pueden construir scripts completos, para ajustar los parámetros y automatizar las simulaciones



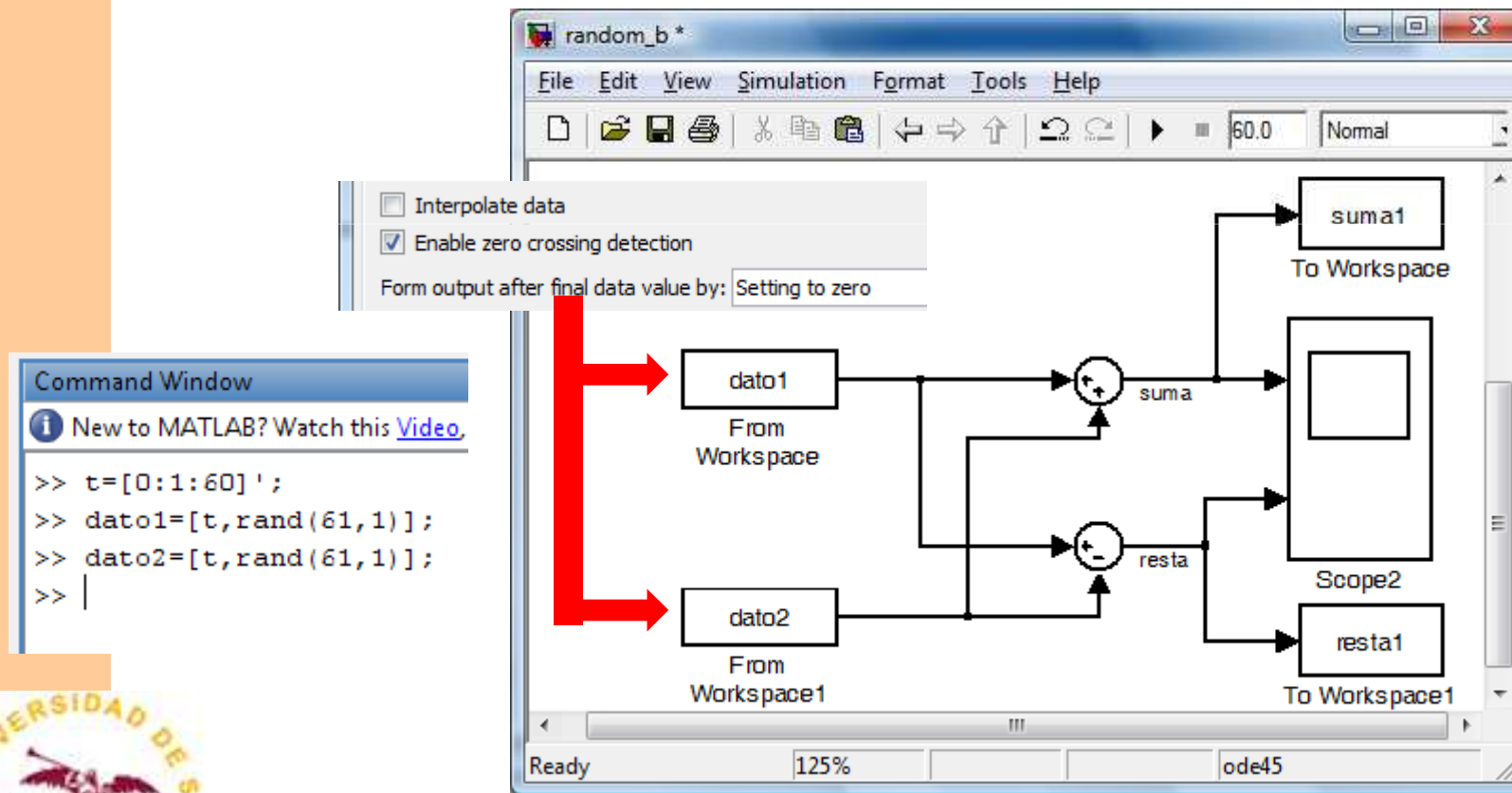
Ejemplo: Operar 2 fuentes de n^{os} aleatorios

- Completamente en Simulink



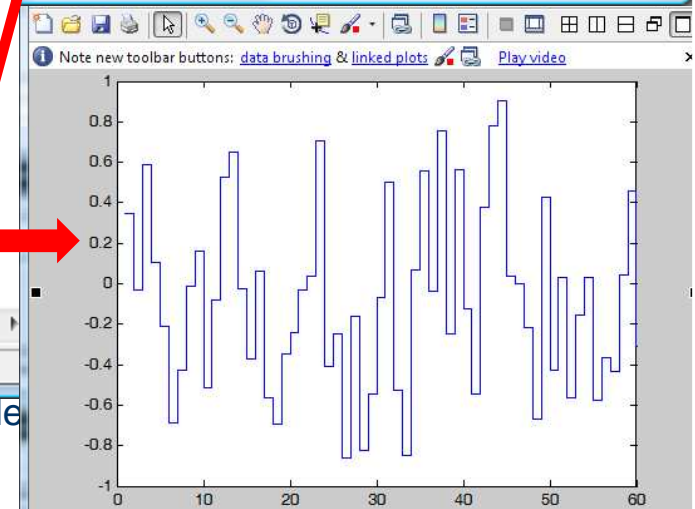
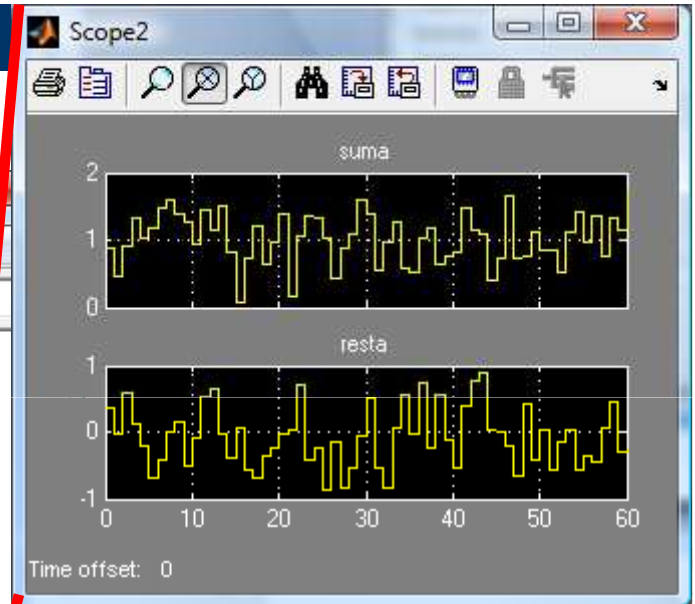
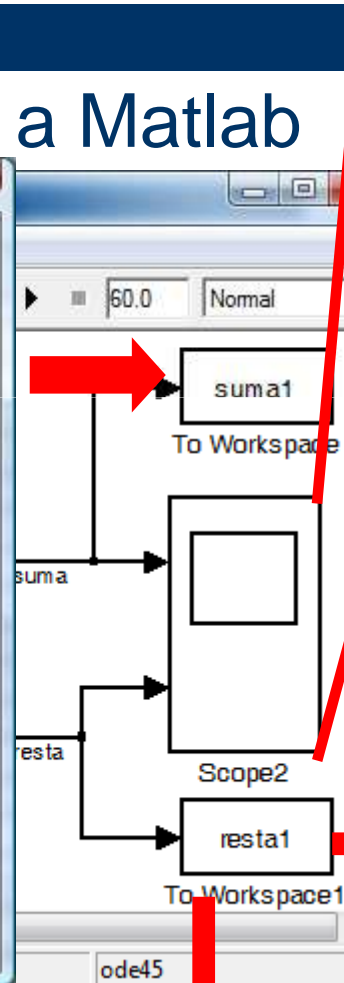
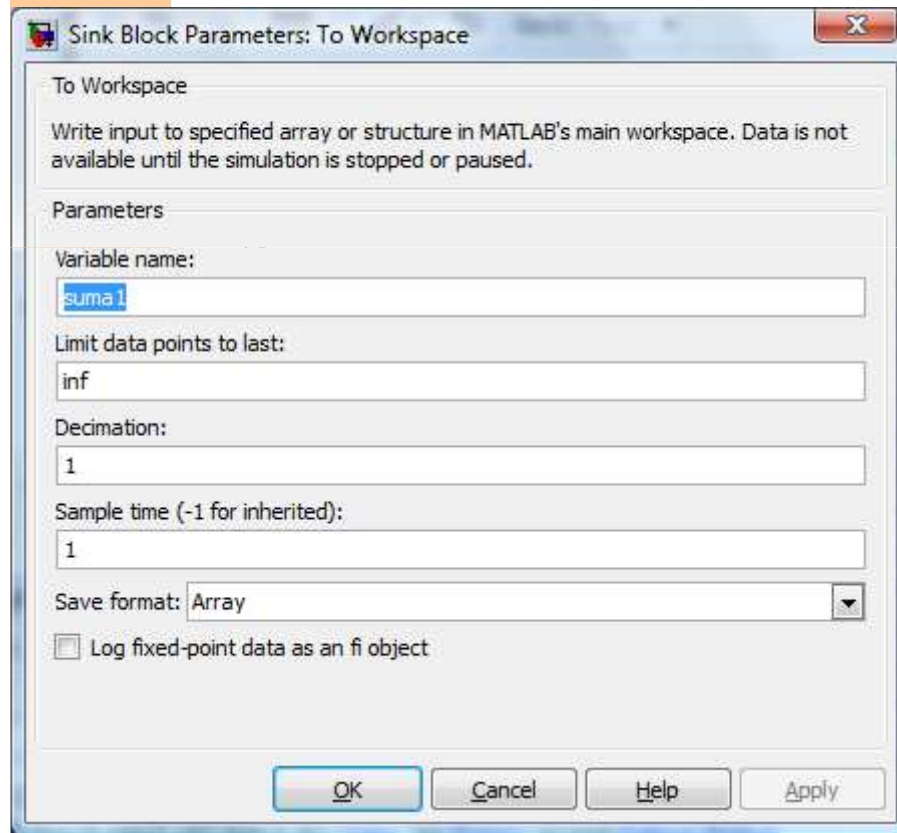
Ejemplo: Operar 2 fuentes de $n^{\circ}s$ aleatorios

- Generando los datos en Matlab



Ejemplo: Operar 2 fuentes de n^{os} aleatorios

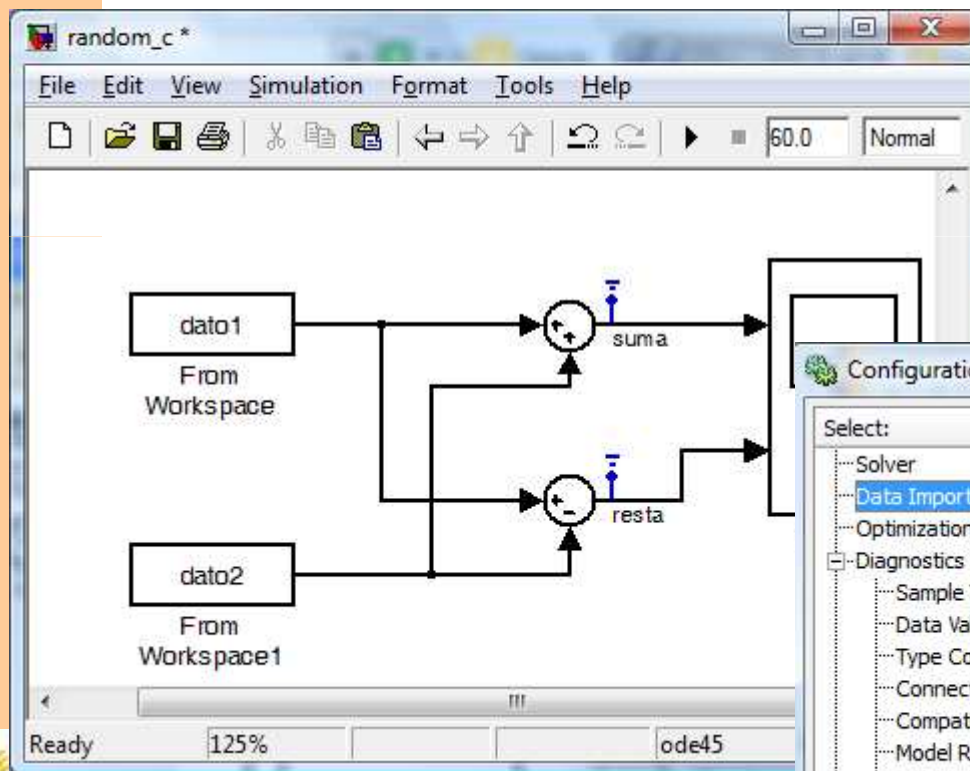
- Pasando los datos a Matlab



Departamento de
>> stairs (resta1)

Ejemplo: Operar 2 fuentes de n^{os} aleatorios

- Pasando los datos a Matlab (II)

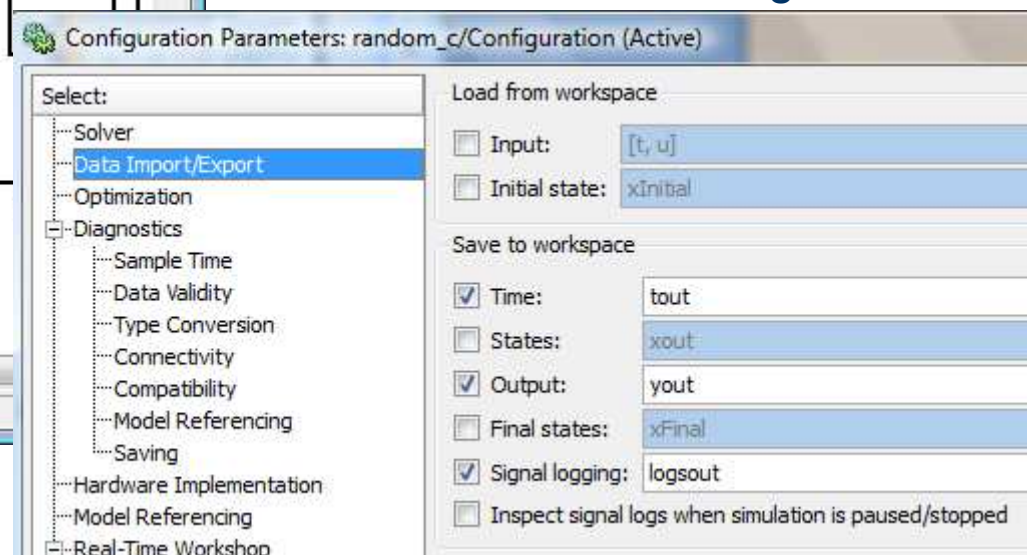


En la línea (bot.dcho.1)

Signal Properties>

☒ Log Signal Data

Simulation>Model Config. Params.



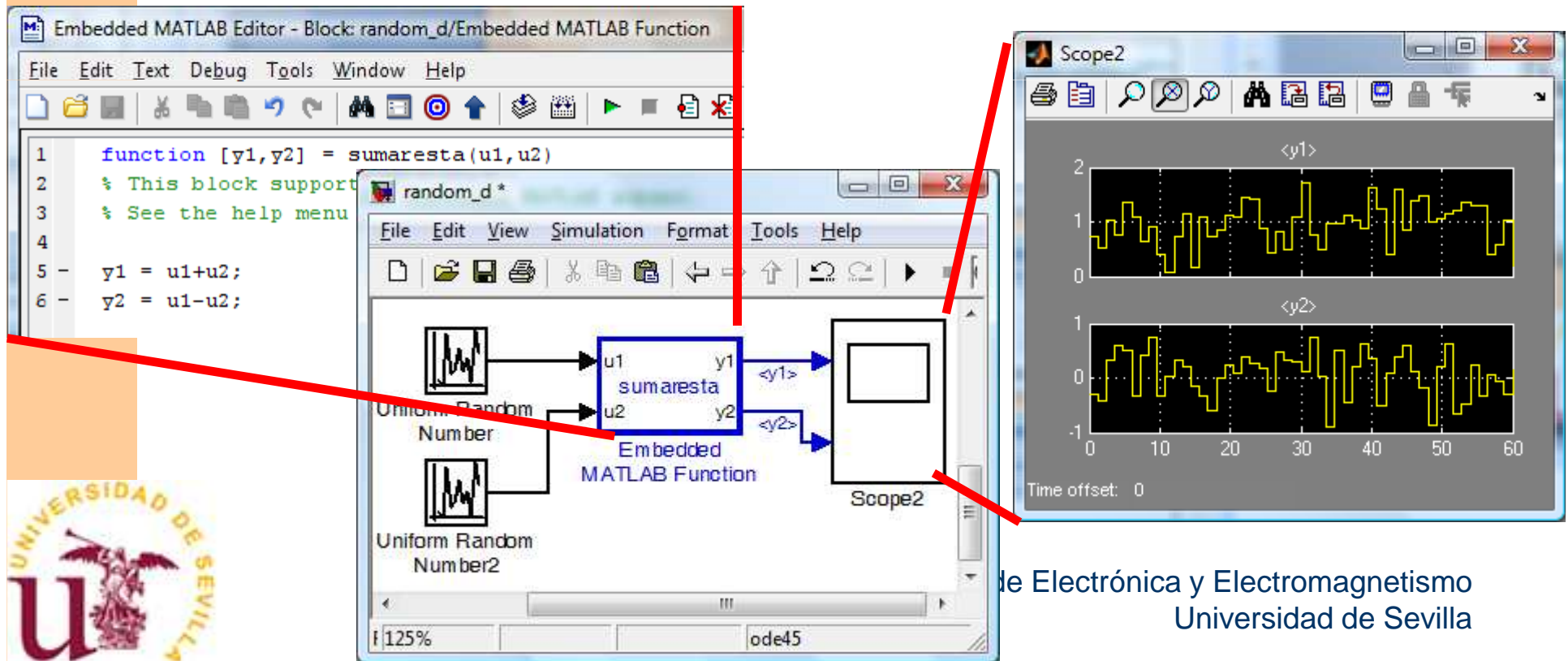
>> plot (logout.suma)

universidad de Sevilla



Inclusión de código Matlab en Simulink

- Uso de “MATLAB fcn” o “Embedded MATLAB Function” en la librería *User-Defined Functions*



El tiempo en Simulink

- Esencial en la simulación dinámica: **no es tiempo real**
- **Sample time**: parámetro que indica cuándo el bloque produce salidas y/o cambia su estado interno (Discrete, Continuous, Fixed in minor step, Inherited, Constant, Variable, Hybrid, Triggered, Asynchronous). [SampleTime_ejemplo.mdl](#)
- Tiempo continuo vs Tiempo discreto



- \neq forma de entender el tiempo y la simulación



Simulación en tiempo continuo y discreto

- Se evalúa $f(t)$:
 - como función continua de t
 - como función definida solo en ciertos instantes
- Un mismo bloque puede simularse como continuo o discreto. [ContinuoDiscreto_ejemplo.mdl](#)
- Caso continuo es discreto con $\Delta t \rightarrow \text{timestep}$
- Simulación discreta es conducida por eventos “Event Driven” \rightarrow Librería *Discrete* (Transf. Z)
- Implementación con “bloques de retraso” ($1/z$)
- Distintos solvers



Ejemplo simulación tiempo discreto

- Simular la evolución de la población (P), tras varias generaciones, con distintas tasas de crecimiento (r):

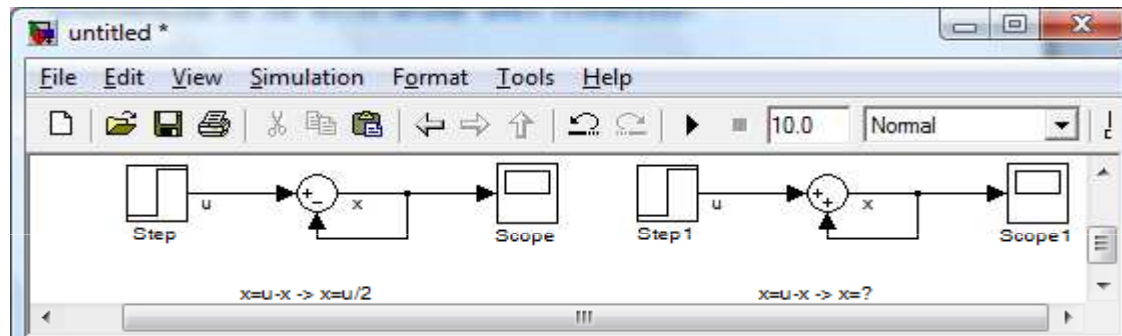
$$P_{\text{next}} = [1 + r(1-P)] \cdot P$$

- El paso entre generaciones lo marca el bloque Unit Delay (1/z)
 - Modelo en [population_fractal.mdl](#)
 - <http://library.thinkquest.org/26242/full/ap/ap16.html>



Lazos algebraicos

- La salida de un bloque es entrada al mismo.



- Si Simulink no resuelve la situación, da error.
- Como norma general hay que eliminarlos:
 - Incluir en el lazo *Unit Delay*, *Memory* o condiciones iniciales (bloque *CI* en *Signal Attributes*)
 - Transformar ecuaciones algebraicas en ODEs
 - ...

Estructuras de programación

- Los modelos de Simulink pueden incluir estructuras de programación convencionales
 - Llamadas a subsistemas
 - If-else (`if_else_subsystem_ejemplo.mdl`)
 - Bucles For/While (`While_ejemplo.mdl`)
 - Habilitación de subsistemas (`enabled_triggered_ejemplo.mdl`)
 - ...
- Librería Port&Subsystems



Tutoriales y Demos

- http://www.mathworks.es/academia/student_center/tutorials/simulink-launchpad.html
- Ayuda de Simulink

