









- Introducir conceptos de diseño basado en modelos
- Aprender a diseñar modelos de sistemas
- Conocer el entorno Simulink y sus aplicaciones
- Diseñar y simular modelos en Simulink



- Modelado dinámico de sistemas
- Introducción a Simulink
- Simulación de ecuaciones diferenciales
- Creación de subsistemas
- Herramientas de análisis en Simulink
- Depuración
- Extensiones de Simulink



- El modelo matemático de un sistema dinámico es descrito por un conjunto de ecuaciones
- El modelo de diagrama de bloques de Simulink es una representación gráfica del modelo matemático de un sistema dinámico
- Las ecuaciones matemáticas descritas por un modelo de diagrama de bloques son conocidas como ecuaciones algebraicas, diferenciales y/o de diferencia



- Dependiendo del sistema modelado y del propósito de la simulación, el modelo matemático puede ser:
  - Lineal o No Lineal: los modelos lineales obedecen al principio de superposición
  - Parámetros Distribuídos: son descritos por ecuaciones diferenciales parciales usualmente con el tiempo y una o más coordenadas espaciales como variables independientes
  - Estáticos y Dinámicos: los modelos estáticos no toman en cuenta la variación del tiempo mientras que los modelos dinámicos si lo hacen



- Continuos y Discretos: los modelos de tiempo continuo son descritos por ecuaciones con variables continuas en el tiempo. Los modelos discretos son descritos por ecuaciones cuyas variables dependientes son muestreadas en intervalos fijos de tiempo
- Determinístico o Estocástico: un modelo determinístico, produce el mismo resultado al ejecutarse si se cumplen las mismas condiciones iniciales. Un modelo estocástico es aquel en el cual la información pasada, no permite la formulación de una regla para determinar el resultado preciso de un experimento



- Es una herramienta interactiva (tipo "drag and drop"), para modelar, simular y analizar sistemas dinámicos, basada en diagramas de bloques
- Permite a los usuarios concentrarse en la estructura del problema, en lugar de tener que preocuparse acerca de un lenguaje de programación
- Los parámetros de cada bloque de señal y sistema son configurados por el usuario. La simulación se realiza sobre un tiempo determinado
- Simulink es una extensión de MATLAB



- Definición de un modelo y representación matemática
   Manipular ecuaciones para eliminar lazos algebraicos
- Definición de los parámetros del sistema
- Seleccionar un método de integración apropiado
- Ajuste de las condiciones de ejecución de la simulación





Para activar el entorno de desarrollo Simulink se escribe en la ventana de comandos Matlab
 > simulink J

 o se pulsa sobre el icono

📣 MATLAB R20	15a - academic use					
HOME	PLOTS	APPS				
New New Script V	Open	Import Save Data Workspace	<ul> <li>New Variable</li> <li>Dpen Variable ▼</li> <li>Clear Workspace ▼</li> </ul>	Analyze Code	Simulink Library	O Preferences Layout
	FILE	V	ARIABLE	CODE	SIMULINK	ENVIRONMENT
< 🔶 🔁 🔊	퉬 🕨 d: 🕨 Users	▶ pedro ▶ Cursos	▶ MatlabSimulink ▶ Sin	nulink 🕨		
Current Folder	C	Command Wind	ow			
Name	bloy	fx >> simuli	nk			



## Librería de bloques de Simulink

Simulink Library Browser		-	10 × 1	
🗢 🔶 Enter search term 🔹 🖎 👻 📆	- 🔄 🗕 🤇	3		
Simulink				
<ul> <li>Simulink</li> <li>Commonly Used Blocks</li> <li>Continuous</li> <li>Dashboard</li> <li>Discontinuities</li> <li>Discrete</li> <li>Logic and Bit Operations</li> <li>Logicup Tables</li> </ul>	Commonly Used Blocks	Continuous	Discontinuities	Discrete
Math Operations Model Verification Model-Wide Utilities Ports & Subsystems Signal Attributes Signal Routing Sinks	Logic and Bit Operations	Lookup Tables	Math Operations	Model Verification
Sources User-Defined Functions Additional Math & Discrete Communications System Toolbox Communications System Toolbox HDL Sup; Control System Toolbox	Misc Model-Wide Utilities	Ports & Subsystems	Signal Attributes	Signal Routing
<ul> <li>DSP System Toolbox</li> <li>DSP System Toolbox HDL Support</li> <li>HDL Coder Image Acquisition Toolbox Instrument Control Toolbox</li> <li>Neural Network Toolbox</li> <li>Simscape</li> <li>Simulink 3D Animation</li> </ul>	Sinks	Sources	User-Defined Functions	Additional Math & Discrete
4				



- Simulink proporciona un conjunto muy amplio de diagramas de bloques para describir sistemas continuos, discretos e híbridos
- Un sistema continuo es un sistema que se representa por ecuaciones diferenciales. Un sistema discreto se representa por ecuaciones en diferencia. Un sistema híbrido contiene componentes continuos y discretos
- Un diagrama de bloques describe un conjunto de relaciones que puede representar un conjunto de ecuaciones simultáneas



- La mayoría de sistemas físicos son modelados como sistemas continuos, porque pueden ser descritos mediante ecuaciones diferenciales (ecdif)
- Hay cuatro bloques primitivos que pueden modelar cualquier sistema descrito por ecdif. lineales:
  - Ganancia
  - Suma
  - Derivada
  - Integrador
- El bloque de función de transferencia es muy usado para el modelado se sistemas físicos y controladores



- Bloque ganancia
- x(t)y = kx→ y(t) k Bloque suma c = a - bа > C b Bloque derivada Notación Laplace  $y = \frac{dx}{dt} \qquad x \longrightarrow du/dt \longrightarrow y \qquad x \longrightarrow$ S → y Bloque integrador Notación Laplace ſ  $y(t) = y(t_0) + \int_{t_0}^t x(\tau) d\tau \quad x \longrightarrow$  $\rightarrow$  y x  $\rightarrow$ 1/s  $\rightarrow v$



- Bloque función de transferencia
  - Se usa frecuentemente en el diseño de sistemas de control y modelado de sistemas
  - Se define como la división de la transformada de Laplace de la salida del sistema entre la transformada de Laplace de la entrada del sistema asumiendo condiciones iniciales cero
- Ejemplo:  $m\ddot{x} + c\dot{x} + kx = F$ Notación Laplace:  $ms^{2}X(s) + csX(s) + kX(s) = F(s)$   $G(s) = \frac{X(s)}{F(s)} = \frac{1/m}{s^{2} + \frac{c}{m}s + \frac{k}{m}}$ FT







Bloque de función de transferencia:

$$\mathsf{F} \longrightarrow \boxed{\frac{1/m}{s^2 + (c/m)s + (k/m)}} \longrightarrow \mathsf{X} \qquad \mathsf{F} \longrightarrow \mathsf{G(s)} \longrightarrow \mathsf{X}$$



- El bloque de espacio de estado es alternativo al bloque función de transferencia
- Representa de forma compacta la dinámica de un sistema
  - Permite especificar condiciones iniciales y proporciona acceso a variables internas
  - Permite modelar sistemas con múltiples entradas y múltiples salidas (MIMO)

- Concepto de espacio de estado:
  - Un vector de estado es un conjunto de variables de estado suficientes para describir el estado dinámico del sistema.
  - La forma general del modelo de espacio de estado de un sistema dinámico es:

 $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$ 

donde **x** es el vector de estado, **u** es el vector de entrada y *t* es el tiempo.

– La ecuación resultado o salida del sistema es:

 $\mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u}, t)$ 



Espacio de estado aplicado a sistemas lineales

• Ecuaciones generales:

 $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$  $\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$ 

donde:

- A Matriz del sistema
- B Matriz de entrada
- C Matriz de salida
- **D** Matriz de transmitancia directa



- Un sistema discreto puede ser representado usando ecuaciones en diferencia y operan con señales discretas (representadas por pulsos x(k), k es el número de pulso)
- Los bloques discretos básicos son:
  - Ganancia discreta
  - Suma discreta
  - Retardo unitario (unit delay)
  - Integrador discreto en tiempo



Bloque suma discreta

$$c = a - b$$
 a

• Bloque retardo unitario <sup>b</sup>  $y(k) = x(k-1) \quad x(k) \longrightarrow T \quad \longrightarrow x(k-1) \quad x(k) \longrightarrow z^{-1} \quad \longrightarrow x(k-1)$ 

> C

• Bloque integrador Backward int. Forward int.  $y(k) = y(k-1) + \int_{T(k-1)}^{Tk} u(t)dt \quad x \longrightarrow \boxed{\frac{Tz}{z-1}} \longrightarrow y \quad x \longrightarrow \boxed{\frac{T}{z-1}} \longrightarrow y$ 



- Bloque función de transferencia
  - Análogo al bloque ft continuo
  - Se define como la división de la transformada Z de la salida del sistema entre la transformada Z de la entrada del sistema asumiendo condiciones iniciales cero
- Representación:





- Similar al bloque de espacio de estado continuo siendo las variables de estado discretas
- Ecuaciones generales:

 $\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)$  $\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k)$ 

donde:

- X vector de variables de estado
- **u** vector de entrada
- ${f y}$  vector de salida



- Simulink proporciona un amplio conjunto de bloques adicionales a los lineales. Los más comunes son.
  - Producto
  - Abs
  - Lógicos (AND, OR, NOT)
  - Relacionales (<,>)
  - Sign
  - Función saturación
  - Retardo de transporte
  - Búsqueda en tabla (interpolación)



## **Bloques Simulink**





## **Bloques Simulink**

Bús	queda tabla	a Op	eracione	s m	natemáticas	Sir	nks		Lógicos	yО	p. Bit
cos(2*pl*u)	Cosine	u]	Abs	Re	Real-Imag to Complex		Display	ղիտ	Band-Limited White Noise	$\mathbb{M}$	Random Number
	Direct Lookup Table (n-D)	+ +	Add	U(:)	Reshape		Floating Scope	-/W	Chirp Signal	M	Repeating Sequence
	Interpolation Using Prelookup	solve f(z) = 0	<sup>z</sup> Algebraic Constraint	floor	Rounding Function		Out 1	Θ	Clock	$\mathbb{N}$	Repeating Sequence Interpolated
	Lookun Table		Assignment		Sign		Scope	1	Constant	հՆ	Repeating Sequence Stai
			bias	t∱√	Sine Wave Function		00000	1715	Counter Free-Running		
	Lookup Table (2-D)		Complex to Magnitude-Angle	1	Slider Gain	STOP	Stop Simulation	liJJY	Counter Limited		Signal Builder
n-0 T(u)	Lookup Table (n-D)		Complex to Real-Imag	Squeeze	Squeeze		Terminator	12:34	Digital Clock	••• ••	Signal Generator
x xdat y	Lookup Table Dynamic	÷	Divide Dat Product	+ -	Subtract	untitled.mat	To File	untitled.mat	From File	$\wedge$	Sine Wave
	Prelookup		Gain	( <del>+</del> _+)	Sum	simout	To Workspace	simin	From Workspace		Step
sin(2*pi*u)	Sine	벨 고	Magnitude-Angle to Complex	Σ	Sum of Elements		W.Gooh	4	Ground	M	Uniform Random Number
		e <sup>u</sup>	Math Function	sin	Trigonometric Function		X1 Graph	1	In1		
			Matrix Concatenate	-u	Unary Minus			ЛЛ	Pulse Generator		
		min	MinMax		Vector Concatenate				Ramp		
		u R R	MinMax Running Resettable	u+Ts	Weighted Sample Time Math						
		P(2.1)	Permute Dimensions								
		P(u) O(P) = 5	Polynomial								
		×	Product								
		$\Box$	Product of Elements								



 Haciendo doble-click sobre cualquier bloque o seleccionando con el botón derecho del ratón los parámetros del bloque se obtiene una ventana para configurar el bloque según las necesidades del modelo

	Function Block Parameters: Integrator
	Integrator
	Continuous-time integration of the input signal.
	Parameters
🙀 untitled *	External reset: none
File Edit View Simulation Format Table Hale	Initial condition source: internal
File Edit View Simulation Pormat Tools Help	Initial condition:
D   🚔 🔲 🙈 👗 ங 💼 (与 🔿 介   ♡ C   ▶ 🔳 10.0 🖌 Normal 📃 🔍 🖾 🛤 🕲 🍪 🛗 🔚 📾 📾 🛞	0
	Limit output
	Upper saturation limit:
	Lower saturation limit:
Integrator	Show estimation port
	Show state port
	Absolute tolerance:
	auto
	Ignore limit and reset when linearizing
Ready 100% ode45	Enable zero crossing detection
	State Name: (e.g., 'position')
	<u>OK</u> <u>Cancel</u> <u>Help</u> <u>Apply</u>



- Es posible desarrollar bloques para Simulink y agruparlos para la modelación y simulación de sistemas específicos
- Existen varios blocksets que se pueden añadir a Simulink
  - Aerospace Blockset
  - Communications Blockset
  - Gauges Blockset
  - Signal Processing Blockset
  - Video and Image Processing Blockset



- Para editar un modelo
  - Abrir la ventana de un nuevo modelo
  - Conectar los bloques
  - Añadir bloques
  - Cambiar el tamaño de los bloques
  - Modificar etiquetas y añadir anotaciones
  - Parametrizar los bloques y la simulación
  - Ejecutar y modificar
- Guardar un modelo (formato Simulink y .m) Abrir un modelo desde Matlab



- Un modelo Simulink típico consiste de tres elementos:
  - Entradas, Sources o inputs
    - Constantes, generadores de funciones (ondas senoidales, escalón o señales creadas en Matlab)
  - Sistema modelado, representado por el diagrama de bloques
  - Salidas, Sinks u outputs
    - Gráficos, osciloscopios, ficheros





 Para crear un modelo en Simulink se pulsa sobre el icono New model □ del Simulink Library Browser o se selecciona File → New → Model

Simulink Library Browser	
<u>File E</u> dit <u>V</u> iew <u>H</u> elp	
(□)≌ -≈ ₩	
Commonly Used Blocks: simulink/Comm Used Blocks	ionly
E 🙀 Simulink	Comr used

뒞 s	imulink Library Browser	1000	
File	Edit View Help		
	New •	Model	Ctrl+N
	Open Ctrl+O	Library	
	Close		
	Preferences		



Espacio de trabajo

• En el espacio de trabajo se colocarán los diagramas de bloque del modelo

<u>File Edit View H</u> elp	
🗅 🚔 -🛱 🏘 🔽 👘 💭 👘 untitled	<u> </u>
Commonly Used Blocks: simulink/Commonly           Used Blocks	
	]
Simulink       Commonly Used Blocks         All Continuous       Continuous         Discrete       Discrete         Discrete       Math Operations         Model Verification       Model-Wide Utilities         Discrete <td< td=""><td>1.</td></td<>	1.



## Añadir bloques

# Hacer click sobre una librería para desplegar los bloques





## **Conectar los bloques**

Para añadir un conector: Arrastrar, pulsando el botón derecho del ratón y la tecla Ctrl, desde una salida, o desde una entrada, de alguno de los bloques al otro bloque





## Cambiar el tamaño y/o mover los bloques

Tras seleccionar el bloque, aparecen en él los puntos, desde los cuales se puede arrastrar para cambiar el tamaño del bloque



Posteriormente se pueden mover el bloque para que las líneas de conexión queden rectas



### Hacer click en la etiqueta y editarla



Hacer dobleclick en el fondo y escribir el texto

## Parametrizar los bloques

untitled *	🛃 'Gráfica salida' parameters
<u>File Edit View Simulation Format Tools Help</u>	General Data kistowa Tim two vielt eliekieg og Cerrar
File       Edit       View       Simulation       Format       Tools       Help         Image:	General       Cerrar         Ceneral       Tim terristic status         Function Block Parameters: Sistema       X         Transfer Fcn       The numerator coefficient can be a vector or matrix expression. The denominator coefficient must be a vector. The output width equals the number of rows in the numerator coefficient. You should specify the coefficients in descending order of powers of s.         Parameters       X         Num       Sink Block Parameters: Salida         To Workspace       Write input to specified array or structure in MATLAB's main workspace. Data is not available until the simulation is stopped or paused.
Parameters         Step time:         Initial value:         0         Final value:         1         Sample time:         0         ✓         Interpret vector parameters as 1-D         ✓         Enable zero crossing detection	Absi       Parameters         autt       Variable name:         Simout       Imit data points to last:         Inf       Decimation:         1       Sample time (-1 for inherited):         -1       Save format: Structure         Imit data as an fi object


- Para guardar el modelo seleccionar File  $\rightarrow$  Save
- El sufijo de los modelos Simulink es .mdl
- Desde la ventana de comandos de Matlab se puede abrir el modelo escribiendo el nombre del fichero

🙀 ur	titled *			🙀 Save As	_			×
File	Edit View Simulation	Format	Tools <u>H</u> elp	Guar <u>d</u> ar en:	MATL/	AB	← 🗈 💣 📰 ▾	
	New	+		(Ass	Nombre	*	Fecha de modifica	Tipo
	Open	Ctrl+0		2	🔒 demo	deploy	03/12/2010 14:03	Carpeta
	Close	Ctrl+W	╶┭─╼┝└─┘│	Sitios recientes	🚺 pr_de	ploy	03/12/2010 14:33	Carpeta
	Save	Ctrl+S	Gráfica					
	Save As		salida	Escritorio				
	Source Control	÷	simout					
	Model Properties		Salida	Bibliotecas				
	Preferences		den	Equipo				
	Export to Web							
	Print	Ctrl+P		Red				
	Print Details				•			
	Print Setup							
	Enable Tiled Printing				Nombre:	ejem 1	<u> </u>	<u>G</u> uardar
	Exit MATLAB	Ctrl+Q			<u>T</u> ipo:	Simulink Models (*.mdl)	•	Cancelar



- Asignar 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



## Asignar parámetros de la simulación

File Edit View Simulation Format Tools Help         Start       Ctrl+T         Stop       Configuration Parameters         Configuration Parameters       Ctrl+E         Configuration Parameters       Ctrl+E         Accelerator       Rapid Accelerator         External       Select         Sistema de Primer Orden       Salida         Select       Sulda         Solver       Stat time (0.0         Stat time:       Stat time:         Data Inpot/Export       Solver options         Data Inpot/Export       Solver options         Type:       Vaiable-step Solver:       ode45 (Domand-Prince)          Model Referencing       Min step size:       auto         Min step size:       auto       Absolute tolerance; auto         Initial step size:       auto       Absolute tolerance; auto         Initial step size:       auto       Absolute tolerance; auto         Normal       Automatically handle data transfers between tasks       Higher priority value indicates higher tasks, priority
Number of consecutive min step size violations allowed:       1         Consecutive zero crossings relative tolerance:       10*128*eps         Number of consecutive zero crossings allowed:       1000



 Cuando se ejecuta una simulación, Simulink resuelve el conjunto de ecuaciones diferenciales y diferencia numéricamente usando uno de los solvers

Configuration Parameters: ejem1/Configuration (Active)						
Configuration Parameters Select: Configuration Solver Continuation Con	ejem1/Configuration     Simulation time     Start time: 0.0     Solver options     Type:     Max step size:     Initial step size:     Zero crossing control     Automatically han     Higher priority val     Solver diagnostic con     Number of consecutive	Variable-step auto auto uuto Use local settings dle data transfers betw ue indicates higher tast trols	Stop time: 10 Solver: Relative tolerance: Absolute tolerance: een tasks priority	.0 ode45 (Dormand-Prince) 1e-3 auto		
	Consecutive zero cr	ssings relative tolerance	e: 10*128*e	eps		
	e zero crossings allowe	ed: 1000				
۲ <u>ــــــــــــــــــــــــــــــــــــ</u>						
			<u>0</u> K <u>C</u> an	cel <u>H</u> elp	Apply	



- El tipo de Solver que proporciona Simulink es amplio y son los más métodos más rápidos y precisos
- Algunos de los solvers proporcionados son:
  - 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 un paso Runge
    - Kutta y pude ser más eficiente que ode45 cuando la tolerancia es amplia
  - ode113: Este es un multipaso, de orden variable Adams Bashforth Moulton PECE. Es recomendable cuando la función evaluación consume tiempo y la tolerancia es poca
  - ode15s: Es un multipaso, de orden variable basado en la fórmula de diferenciación " backward"
  - ode23: un paso basado en la fórmula de Rosembrock de orden 2.



- Otros parámetos se refieren al tiempo de simulación y el paso de integración:
  - Start time
  - Stop time
  - Tipo de solver:
    - Variable
    - Fijo
- El número de puntos, en caso de usar tiempo fijo es:

$$N^{\circ} de puntos = \frac{Stop Time - Start Time}{Step size} + 1$$



#### Parámetros de la simulación

 Otros parámetos tiene que ver con la entrada/salida de datos al modelo y desde el modelo

Configuration Parameters: ejem1/Configuration (Active)						
Select:	Load from workspace	1				
Solver <mark>Data Import/Export</mark> Optimization	Input: [t, u] Initial state: xInitial					
Diagnostics     Hardware Implementation	Save to workspace					
Model Referencing	☑ Time: tout					
È-Real-Time Workshop	States: xout					
	🔽 Output: yout					
	Final states: xFinal					
	Signal logging: logsoul					
	Inspect signal logs whe	n simulation is paused/stopped				
	Save options					
	📝 Limit data points to last:	1000 Decimation: 1				
	Format:	Array				
	Uutput options:	Refine output				
		<u>Q</u> K <u>C</u> ancel <u>H</u> elp <u>Apply</u>				



#### Parámetros de la simulación

 Los parámetros y variables de los modelos se pueden acceder desde la ventana de Comandos de Matlab





• Los parámetros y variables de los modelos se pueden acceder desde la ventana de Comandos de Matlab





 Tanto desde la ventana de Matlab como la de Simulink se "ve" el mismo Workspace o Espacio de trabajo





#### Ejecución de la simulación

 Se pulsa el icono Start ► o en el menú Simulation → Start







- También se puede usar el comando Matlab sim

   [t, x, y] = sim('model', Timespan, Opciones, ut)
   donde *model* es el nombre del diagrama de bloques.
   Timespan especifica la salida de los puntos de tiempo
   Opciones es una estructura que permite asignar los
   valores de los parámetros en la ventana de diálogo
   Simulation:Parameters
  - ut asigna la parte Load de la página Workspace I/O de la ventana Simulation:Parameters

Ejemplo:

- > [t, y] = sim('ejem1\_1',5);
- > plot(t, y)



## Solución de ecuaciones diferenciales que modelan Sistemas Continuos

Construir un modelo en Simulink que resuelva la siguiente ecuación diferencial:

$$\frac{dx}{dt} = 5\sin\left(4t\right)$$

• Condición inicial: x(0) = -2.

• Primero se dibuja un diagrama de bloques de la simulación del modelo matemático (ecuación)



Diagrama del modelo

- Input: función  $5\sin(4t)$
- Output: x(t) que es la solución de la ecuación diferencial



• A continuación, se construye el modelo con Simulink



- La siguiente tabla resume el bloque y la librería donde se encuentra para ser inluído en el modelo
  - Se arrastra el bloque de la librería hasta la ventana de trabajo

Modelo	Librería	Bloque
Input	Sources	Sink
Integrador	Continuous	Integrator
Output	Sink	Scope



### Selección de bloques para el modelo





### Selección de bloques para el modelo





#### Selección de bloques para el modelo





# Conexión de los bloques con líneas de señal

- Colocar el cursor en el puerto de salida (> a la derecha) del bloque "Sine Wave" .El cursor cambia de forma a cruz
- Arrastrar desde el puerto de salida del bloque "Sine Wave" hasta el puerto de entrada (> a la izquierda) del bloque "Integrator". Cuando el cursor se encuentra sobre el puerto de entrada cambia de forma a cruz doble
- Arrastrar desde la salida del bloque
   "Integrator" hasta la entrada del bloque
   "Scope"



Las flechas indican la dirección de la señal.



## Configurar bloques con datos del modelo

- El input del modelo es:
   5sin(4t)
- Para ello se hace doble click en el bloque "Sine Wave" y en la ventana de diálogo de los parámetros del bloque ingresar:

Amplitude = 5

Frequency = 4

Source Block Parameters: Sine Wave	x
Sine Wave	- f
Output a sine wave:	
O(t) = Amp*Sin(Freq*t+Phase) + Bias	
Sine type determines the computational technique used. The parameters in the two types are related through:	o
Samples per period = 2*pi / (Frequency * Sample time)	
Number of offset samples = Phase * Samples per period / (2*pi)	
Use the sample-based sine type if numerical problems due to running for large time (e.g. overflow in absolute time) occur.	s
Parameters	
Sine type: Time based	3
Time (t): Use simulation time	3
Amplitude:	
5	
Bias:	
0	
Frequency (rad/sec):	
4	
Phase (rad):	
0	
Sample time:	
<u> </u>	



## Configurar bloques con datos del modelo

- El valor inicial es: -2
- Para ello se hace doble click en el bloque "Integrator" y se ingresa la condición inicial = -2

Function Block Parameters: Integrator	3					
Integrator						
Continuous-time integration of the input signal.						
Parameters						
External reset: none						
Initial condition source: internal						
Initial condition:						
-2						
C Limit output						
Upper saturation limit:						
irii Lever estrestian limit						
- inf						
Show saturation port						
Show state port						
Absolute tolerance:						
auto						
Ignore limit and reset when linearizing						
Enable zero crossing detection						
State Name: (e.g., 'position')						
•						
<u>O</u> K <u>C</u> ancel <u>H</u> elp <u>Apply</u>						



•

icono Start 🕨

### Ejecutar la simulación

En la ventana de trabajo, click • en "Simulation" y seleccionar "Start"





- Hacer doble click en el bloque "Scope"
- Se visualiza el output x(t) en la ventana Scope
- Se puede mejorar la visualización utilizando los iconos de la ventana. Ej.: Autoscale y Tick labels all





## Ejemplo de Sistema Continuo de segundo orden

 El siguiente sistema muelle-masa-amortiguador se resuelve según la ecuación de movimiento (ignorando fricción):



$$m\frac{d^{2}x}{dt^{2}} + c\frac{dx}{dt} + kx = f(t)$$
  
Notación simplificada:  
$$m\ddot{x} + c\dot{x} + kx = f(t)$$

- Considerar que:
  - el sistema inicialmente está en equilibrio:  $\dot{x} = 0, x = 0$
  - función de la fuerza es un escalón con magnitud 3
  - valores de los parámetros: m = 0.25, c = 0.5, k = 1



# Diagrama de simulación del sistema continuo de segundo orden

 Si se expresa la ecuación en términos de la derivada de mayor orden

$$m\ddot{x} + c\dot{x} + kx = f(t) \longrightarrow \ddot{x} = \frac{1}{m}f(t) - \frac{k}{m}x - \frac{c}{m}\dot{x}$$

• El diagrama de bloques asociado es:





Diagrama de bloques Simulink del sistema continuo de segundo orden

- A partir del modelo se identifican los bloques necesarios en las librerías Simulink:
  - Bloque Step (1) Librería Sources
  - Bloque Sum (1) Librería Math Operations
  - Bloque Gain (3) Librería Math Operations
  - Bloques Integrator (2) Librería Continuos
  - Bloque Scope (1) Librería Sink
- Cada bloque requiere ser configurado con las ICs:
  - el sistema inicialmente está en equilibrio:  $\dot{x} = 0, x = 0$
  - input: escalón con magnitud 3
  - valores de los parámetros: m = 0.25, c = 0.5, k = 1











Arrastrar el bloque "Step"	Untitled *
desde la librería <i>"Math Oper"</i>	<u>File Edit View Simulation Format Tools Help</u>
Simulink Library Browser	
<u>File Edit V</u> iew <u>H</u> elp	
□··· Interpretation     Interpretation       □··· Interpretation     Interpretation       Interpretation     In	Step $1/m$ $\rightarrow  \rightarrow -$
······ ☆ Discontinuities Repeating Sequence	Add
	Source Block Parameters: Step
Math Operations Model Verification Model Verification Model Wide Unification Model Wide Unification	Output a step.
	Ready Step time:
····· ☆ Signal Routing Signal Generator	0 Initial value:
Sources Sine Wave	
⊕      Additional Math & Disc      Step      Step	
Uniform Random	0
Doblo click on Stanpara cambiar	Interpret vector parameters as 1-D
	Enable zero crossing detection
los parámetros <i>Step time</i> a 0 y	
Final value a 3	



## Arrastrar los bloques "*Integrator*" desde la librería *"Continuous"*



El bloque "integrator" tiene por defecto valor inicial 0. No es necesario modificarlo. Se cambian los títulos



#### Arrastrar el bloque "*Scope*" desde la librería "*Sink*"









Conectar todos los bloques. Etiquetar las señales haciendo doble click sobre las líneas.

Para hacer derivaciones de una línea se coloca el cursor sobre la línea y se mantiene pulsada la tecla Ctrl hasta conectar al otro bloque.



## Resultados del modelo en Simulink (scso)



## Comprobación de resultados

Forma Standard

$$\frac{\ddot{x}}{k/m} + \frac{c}{k}\dot{x} + x = \frac{1}{k}f(t)$$
$$\omega_n = \sqrt{\frac{k}{m}} = 2.0$$

- Frecuencia natural
- Tasa de amortiguamiento

$$\frac{2\zeta}{\omega_n} = \frac{c}{k} \quad \to \quad \zeta = 0.5$$

Ganancia estática

$$K = \frac{1}{k} = 1$$



## Uso del bloque de función de transferencia en un scso

- El siguiente sistema muelle-masa-amortiguador se resuelve según la ec. de movimiento (sin fricción):  $m\ddot{x} + c\dot{x} + kx = f(t)$
- Tomando la transformada de Laplace e ignorando las condiciones iniciales

$$ms^{2}X(s) + csX(s) + kX(s) = F(s)$$

• La función de transferencia (output/input X(s)/F(s)) es:

$$G(s) = \frac{X(s)}{F(s)} = \frac{1/m}{s^2 + \frac{c}{m}s + \frac{k}{m}} \qquad G(s) = \frac{4}{s^2 + 2s + 4}$$


# Uso del bloque de función de transferencia en Simulink para un scso



Ajustar el Numerator y Denominator coefficient a [4] y [1 2 4] del bloque "Transfer Fcn"



- El bloque espacio de estado (*state-space*) es una alternativa al bloque función de transferencia
- Las variables de estado para el sistema mma son posición y velocidad

$$\begin{aligned} x_1 &= x\\ x_2 &= \dot{x} \end{aligned}$$

• Las derivada de las variables de estado son

$$\dot{x}_1 = x_2$$
  
 $\dot{x}_2 = -\frac{k}{m}x_1 - \frac{c}{m}x_2 - \frac{1}{m}F$ 



• En notación matricial



$$y = Cx + Du$$
$$y = x_1$$
$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$
$$D = 0$$

donde:

- A = matriz del sistema
- **B** = matriz de entrada
- **C** = matriz de salida
- **D** = matriz de transmitancia directa



### Bloque de espacio de estado en Simulink para un scso



Ajustar los valores de las matrices A=[0 1;-4 -2], B=[0; 4], C=[1 0] y D=[0] del bloque "State Space"



• Se modela la amortización de un prestámo de dinero:

$$b(k) = rb(k-1) - p(k)$$

donde:

*b(k)* pago mensual *r* = *i*+1 siendo i el interés mensual, y *p(k)* pago de final de mes.

• Modelo de bloques p(k) p(k)  $z^{-1}$  b(k-1)  $k_i$ 



## Modelo de bloques de sistema discreto Simulink

• Si el balance de un préstamo es:

$$b(k) = rb(k-1) - p(k)$$

Considerar como condiciones iniciales: Balance inicial: 15000 Interés: 1% Pago mensual = 200

 Calcular el balance del préstamo despues de 100 pagos



# Modelo de bloques de sistema discreto Simulink

- A partir del modelo se identifican los bloques necesarios en las librerías Simulink:
  - Bloque Constant (1) Librería Sources
  - Bloque Sum (1) Librería Math Operations
  - Bloque Gain (3) Librería Math Operations
  - Bloques Unit delay (1) Librería Discrete
  - Bloque Scope y Display (1) Librería Sink
- Cada bloque requiere ser configurado con las ICs:
  - Bloque Unit Delay, Initial Condition: 15000, Sample Time:
     1
  - Bloque Gain: 1.01



# Modelo de bloques de sistema discreto Simulink





# Resultado de la simulación de sistema discreto Simulink





- Simulink proporciona la capacidad de crear subsistemas que equivalen a las subrutinas o funciones de los lenguajes de programación
- Los subsistemas permiten la abstracción y reusabilidad
- Los modelos pueden crecer y hacerse más complejos de comprender y mantener. Con los subsistemas se divide un gran modelo en un conjunto de jerárquico de modelos más pequeños



#### Creación de Subsistemas

- Hay dos métodos:
  - Encapsulando una porción de un modelo existente usando Edit:Create Subsystem
  - Usar el bloque Subsystem de la librería de bloques Ports
     & Subsystems

<u>File</u>	lit <u>V</u> iew	Simulation	F <u>o</u> rmat	<u>T</u> ools	<u>H</u> el	
D	Undo /	Add Line		Ctrl+7	z	
_	Can't F	Redo		Ctrl+	Y	
	Cut			Ctrl+)	ĸ	
	Сору			Ctrl+(	2	
	Paste			Ctrl+	/	
Paste Duplicate Inport						
	Delete	Delete				
	Select	All		Ctrl+/	4	
	Copy I	Model To Clip	board			
	Find			Ctrl+	F	
	Open l					
	Explore	2				
lace	Block	Properties				
	Create	Subsystem		Ctrl+(	3	
2	Mask S	Subsystem		Ctrl+N	1	
	Look U	Inder Mask		Ctrl+l	J	
C C	Link O	ptions			Þ	
	Refres	n Model Block	s	Ctrl+I	ĸ	
	Update	e Diagram		Ctrl+[	)	





# Creación de Subsistemas: Encapsulado

Seleccionar todos los bloques y señales a ser incluídos en el subsistema usando una caja envolvente

Seleccionar Edit:Create Subsystem del menu del modelo. Simulink reemplaza los bloques seleccionados por un bloque Subsystem con puertos de entrada y salida según las líneas de señales y les asigna un nombre por defecto

Cambiar el tamaño del bloque Subsystem para que las etiquetas de puertas sean legibles y mover para alinear





## Creación de Subsistemas: Encapsulado

Para ver o editar el subsistema, hacer doble-click sobre el bloque. Aparecerá una nueva ventana conteniendo el subsistema.

Además del los bloques, se añade un bloque Inport para la señal que entra al subsistema y un bloque Outport se añade para la señal que sale del subsistema. Si se cambia las etiquetas de esos puertos, cambia las etiquetas en el icono del nuevo bloque.

**Nota**: Una vez se crea el subsistema no hay operación inversa, por lo que se sugiere guardar el modelo antes de crear el subsistema.





- Si se sabe que es necesario subsistemas para crear un modelo, es conveniente construir el subsistema en una ventana de subsistema directamente
  - Para ello se arrastra el bloque Subsystem de la librería de bloques Ports & Subsystems a una ventana de modelo.
     Después hacer doble-click sobre el bloque.





• Se desea modelar un sistema de masa-resorte:



El modelo de cada móvil es:





Se construye el subsistema de un móvil según el modelo. La ganancia del bloque 1/masa se asigna 1/m1.





Se asignan las constantes de los resortes y masas desde Matlab creando un fichero .m (pe setSubsys) y se ejecuta en Matlab % Asignación de constantes para resortes y masas k1 = 1; k2 = 2; k3 = 4; m1 = 1; m2 = 3; m3 = 2;

Los parámetros de cada bloque subsystem se configura:

Móvil 1: valor del bloque Gain del bloque Resorte Izq a k1 y para el bloque Resorte Der a k2. Valor del bloque Gain del bloque 1/masa a 1/m1. Inicializar el bloque Integrator Velocidad a 0 y el Integrator Posicion a 1.

Móvil 2: valor del bloque Gain del bloque Resorte Izq a k2 y para el bloque Resorte Der a k3. Valor del bloque Gain del bloque 1/masa a 1/m2. Inicializar el bloque Integrator Velocidad a 0 y el Integrator Posicion a 0.

Móvil 3: valor del bloque Gain del bloque Resorte Izq a k3 y para el bloque Resorte Der a 0. Valor del bloque Gain del bloque 1/masa a 1/m3. Inicializar el bloque Integrator Velocidad a 0 y el Integrator Posicion a 0.



Se configura el bloque Scope para almacenar los datos de salida al workspace. Asignar Start time a 0 y Stop time a 100. Se puede obtener la gráfica desde Matlab con el comando simplot. Después de ejecutar las simulación, se obtiene:





- El enmascarado es una capacidad de Simulink que extiende el concepto de abstracción
- Con ello se trata un subsistema como si fuera un bloque simple con su propio icono y diálogo de configuración de parámetros
- Para producir un bloque enmascarado se construye el subsistema como se ha mostrado antes, se selecciona y se selecciona Edit:Mask Subsystem
- Usando el editor de máscara, se configura la documentación, diálogo de propiedades e icono



Se crea un subsistema. Usamos el desarrollado para el sistema masaresorte copiando en una nueva ventana de modelo el bloque Masa-Resorte1

Ľ

 Image: state of the state

Seleccionar el bloque y se escoje **Edit:Mask Subsystem** en el menú de la ventana de modelos para obtener la ventana de diálogo del editor Mask. Guardar el modelo (CartMask)

itlec	*	- • ×	Mask Editor : Masa-Resorte1
Edit	<u>View</u> Simulation Format	<u>T</u> ools <u>H</u> elp	
	Undo Move	Ctrl+Z	Icon Parameters Initialization Documentation
	Can't Redo	Ctrl+Y	- Device contraction
	Cut	Ctrl+X	Drawing commands
	Сору	Ctrl+C	Frame
	Paste	Ctrl+V	Visible
	Paste Duplicate Inport		
	Delete	Delete	Transparency
	Select All	Ctrl+A	Onaque
	Copy Model To Clipboard		
	Find	Ctrl+F	Rotation
	Open Block		Fixed 👻
	Explore		
	Mask Parameters		Units
	Subsystem Parameters	1	Autoscale 👻
	Block Properties	/	
	Convert to Model Block		
	Create Subsystem	Ctrl+G	Examples of drawing commands
	Mask Subsystem	Ctrl+M	
	Look Under Mask	Ctrl+U	Command port_label (label specific ports)
	Link Options	) E	Syntax port_label('output', 1, 'xy')
	Refresh Model Blocks	Ctrl+K	
	Update Diagram	Ctrl+D	



El editor Mask tiene cuatro pestañas: Icon, Parameters, Initialization, Documentation.

La página Documentation del editor sirve para establecer un tipo, colocar una descripción del bloque y una descripción más detallada que se mostrará cuando se selecciona el sistema Help de Matlab

Mask Editor : Masa-Resorte1								
Icon Parameters Initialization Documentation								
Mask type								
Masa-Resorte								
Mask description								
Este bloque modela un carro con un resorte a cada lado.								
Mask help								
Este bloque modela un carro con un resorte sin amortiguador a cada lado. La entrada son los desplazamientos desde el equilibrio de cada resorte. La salida del bloque es la posición del carro. Si sólo hay un resorte añadido al carro, asignar la constante para el otro resorte a cero. Si un resorte está fijo, asignar la entrada correspondiente a cero usando un bloque constante.								
Unmask OK Cancel Help Apply								



La página Parameters del editor Mask se usa para definir parámetros para el bloque del Aña susbsistema. Se divide Elin en dos secciones: Una Μο superior en la que se Μο define los campos de diálogo y el orden en que se muestran, asociando una variable Matlab con cada campo. La sección inferior contiene opciones para cada campo definido en la parte superior.

ſ	Mask Editor : Masa-Resorte1							
	Icon	Parameters	Initialization	Docume	ntation			
		Dialog para	meters					
adir —	->=+	Prompt	Variable	Туре	Evaluate	Tunable		
ninar —	→×							
ver arriba	->:							
ver abajo	→							
	·	Options for	selected pa	rameter				
		Popups (on	e per line):	In dialog:	Show	Enabl		
				Dialog callback:				
	Unma	isk	O	Can Can	cel Hel	p Apply		



La pestaña Parameters se usa para crear, editar y eliminar campos de diálogo asociado a variables Matlab con cada campo. Para cada campo añadido se asocia un Prompt, Variable, Type, Evaluate y Tunable. Resultado:

Function Block Parameters: Masa-Resorte1
Masa-Resorte (mask)
Este bloque modela un carro con un resorte a cada lado.
Parameters
Cons.resorte izq.:
0
Cons.resorte der.:
0
Masa:
Posición inicial:
Velocidad inicia:
<u>O</u> K <u>C</u> ancel <u>H</u> elp <u>Apply</u>

Mask Editor : Masa-Resorte1											
Icon Parameters Initialization Documentation											
Dialog parameters											
-	Prompt		Variable	Туре	Eval	Tun					
	Cons.resor	te izq.:	k_izq	edit 👻	<b>V</b>	<b>V</b>					
	Cons.resor	te der.:	k_der	edit 👻	<b>V</b>	<b>V</b>					
	Masa:		mass	edit 👻	<b>V</b>	<b>V</b>					
	Posición in	icial:	×0	edit 👻	<b>V</b>	<b>V</b>					
	Velocidad i	inicial:	x_dot0	edit 👻	<b>V</b>	<b>V</b>					
Options for selected parameter Popups (one per line): In dialog: Show Enabl Dialog callback:											
Unmask OK Cancel Help Apply											



El campo Type en Dialog parameters puede ser: edit, Checkbox y Popups . El valor asignado a la variable asociada con un campo de diálogo depende del estado de si selecciona el campo Evaluate.

Si Evaluate se selecciona la variable asociada con el campo tendrá el valor de la expresión del campo. El campo Tunable determina si el parámetro se puede modificar durante la ejecución de la simulación.

El campo Callback permite asociar con el parámetro un bloque de código que se ejecuta cuando el parámetro de diálogo se ingresa.

₫	Mask	c Editor : Masa-Resorte:	manufactor .				<b>_</b> ]	
	Icon Parameters Initialization Documentation							
		Dialog parameters						
	-	Prompt	Variable	Туре	Evaluate	Tunable		
	×	Cons.resorte izq.:	k_izq	edit	- 7			
		Cons.resorte der.:	k_der	edit	<u> </u>	<b>V</b>		
		Masa:	masa	edit	<u> </u>	<b>V</b>		
		Posición inicial:	x0	edit	<u>→</u>	<b>V</b>		Frror
		Velocidad inicial:	x_dot0	edit	▼	<b>V</b>		
		Options for selected p	arameter					Environmentions 'Mark/Callback' anilhanks of Mara Darasta
		Popups (one per line):	In dialog: 👿 Show par	ameter	V Enable p	arameter		Error evaluating MaskCaliback caliback of Masa-Resorce
							$\rightarrow$	block (mask) Masa-Resorte I'. Const. resorte debe ser
			Dialog if str2n	um(get_para	am(gcb,'k_izo	'))<0	5	positiva.
	callback: error('Const. resorte debe ser positiva')					ser positiva' <u>)</u>		
			end					
								UK
Ir							1	
	Onmask OK Cancel Help Apply						J	



La página Initialization proporciona una lista de variables asociados con los parámetros del bloque y un campo de Initialization commands. Los bloques del subsistema se debe configurar para usar las variables definidas en las páginas Initialization y Parameters. Para ello se selecciona el subsistema y se escoge **Edit:Look Under Mask** en el menú de la ventana de modelo. Hacer doble click en el bloque Gain con etiqueta Res.Izq. Y asignar Gain a k\_izq. Repetir con Res.Der. Para asignar Gain a k\_der. Lo mismo con 1/mass para asignar Gain a masa. Asignar Initial condition del integrador Velocidad a x\_dot0 y en Posicion a x0.

Mask Editor : Masa-Resorte1	CartMask/Masa-Resorte1
Icon Parameters Initialization Documentation	<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>S</u> imulation F <u>o</u> rmat <u>T</u> ools <u>H</u> elp
Dialog variab - Initialization commands	□     □     □     □     □     □     □     □     Normal
$\frac{k_{der}}{x_{adot0}} = \frac{c_{1}(t, 0, 0, 1, 2, w_{p1}, 0)}{x_{adot0}}$	1     +     -
Unmask OK Cancel Help Apply	Ready 100% ode45



Las variables definidas en la máscara de un subsistema son locales y no pueden acceder a las variables del workspace de Matlab. Un campo de entrada en el diálogo de un bloque enmascarado puede contener constantes o expresiones usando variables definidas en el workspace de Matlab. El valor del contenido del campo de entrada se asigna a la variable interna del subsistema enmascarado asociado con el campo de entrada. Esta variable se puede usar para inicializar el bloque o puede usarse para definir otras variables internas definidas en el campo Initialization commands

	Masa-Resorte (mask) Este bloque modela un carro con un resorte a cada lado.
Add Res.izg + 1 1	Parameters Cons.resorte izq.:
+	Cons.resorte der.:
Izq     Add 1     Res.der       Gain     Element-wise gain (y = K.*u) or matrix ga	Masa: m1 Posición inicial:
Gain: K_del Multiplication: [Element-wise](K*u)	1 Velocidad inicial: 0
	<u>DK Cancel H</u> elp <u>Apply</u>



La página Icon permite diseñar iconos propios para los bloques enmascarados. Consiste de seis campos. En el campo Drawing commands se puede poner comandos Matlab para dibujar y etiquetar el icono.

Mask Editor : Masa-Resorte1	
Icon Parameters Initialization Documentation	
Icon options Drawing commands	
Frame plot([0.2,0.8,0.8,0.2,0.2],	Masa-Resorte1
Visible [0.2,0.2,0.8,0.8,0.2]) plot([0,0.05,0.075,0.1,0.125,0.15,0.2],	
Transparency [0.5,0.5,0.6,0.4,0.6,0.5,0.5])	Si en la página Initialization se
Opaque	agrega la línea:
Fixed plot (0.3+0.075*x, 0.1+0.075*y,	m etig=sprintf('%1.1f kg'.mass);
Units U.7+0.075*x, 0.1+0.075*y)	
Normalized	y en Drawing Commands:
	text(0.45,0.6,'m')
Examples of drawing commands	text(0.25,0.4,m etia)
Command port_label (label specific ports)	
Syntax port_label('output', 1, 'xy')	
Unmask OK Cancel Help Ap	Masa-Resorte1



Una vez que se crea el bloque enmascarado se puede copiar a una ventana de modelo de forma idéntica cuando se copia de la librería de bloques de Simulink. Para el ejemplo se arrastra tres copias del bloque Masa-Resorte, se conecta y se asignan los parámetros con un fichero .m. El modelo así queda completo.







# Creación de Librería de Bloque

Una Librería de Bloque es un modelo especial Simulink que sirve para el mismo propósito como la librería de subrutinas en un lenguaje de programación convencional. Cuando se copia un bloque desde una librería (bloque de referencia) de bloque a un modelo, la copia mantiene un enlace al bloque de la librería. Así si cambia el bloque en la librería, ese cambio se refleja en los modelos donde se ha usado el bloque. Para crear una librería de bloque se selecciona File:New:Library en una ventana de modelo o en la librería de Simulink. Se copia los bloques deseados a la nueva librería y se guarda.





# Creación de Librería de Bloque

Pasos para crear una LibreriaPersonal con bloques Gain y MasaResorte





- Simulink dispone de herramientas de análisis para ganar en la comprensión de los modelos desarrollados, que se usan dentro del espacio de trabajo de Matlab
- Así se puede obtener la estructura del vector del espacio de estado del modelo, el número de entradas y salidas y otros parámetros importantes.
- También se puede ejecutar la simulación desde Matlab y modificar los parámetros del modelo, así como linealizar un modelo, hallar puntos de equilibrio



- Un modelo Simulink es una descripción gráfica de un conjunto de ecuaciones diferenciales, en diferencia y algebraicas
- El comando *modeL* permite determinar la estructura del vector de estado del modelo Simulink

sizes = model([],[],[],0)

[sizes,x0] = model([],[],[],0)

[sizes,x0,states] = model([],[],[],0)

donde *modeL* es el nombre del modelo Simulink



# Ejemplo: Vector de estado del modelo

Se tiene el siguiente modelo Simulink guardado en el fichero Ilamado ejmod.mdl

En la ventana de comandos Matlab se ejecuta el comando:

>> [sizes,x0,states]=ejmod([],[],[],0)
sizes =







- El comando Matlab sim permite ejecutar los modelos Simulink desde Matlab
- El comando sim se puede usan conjuntamente con el comando simset para crear y editar una estructura de datos de opciones y simget que obtiene la estructura de datos de opciones del modelo
- La sintaxis del comando sim es
- [t,x,y]=sim(model,TiemSpan,Options,utd)
   donde model es el nombre del modelo



# Ejemplo: ejecución de modelos desde Matlab

 Para ejecutar el modelo almacenado en el fichero ejmod.mdl usando la configuración por defecto, equivalente a seleccionar Simulation:Start en la barra del menú del modelo. En Matlab se ingresa el comando:

sim('ejmod')

Simulink ejecutará el modelo y si Scope está abierto mostrará la gráfica del resultado cuando la simulación se completa



# Ajuste y obtención de parámetros de simulación desde Matlab

- El comando **simset** se usa para crear y editar la estructura de opciones de la simulación
- Hay tres formas del comando simset, siendo la más común:

options=simset('RelTol',1.0E-4, 'Solver','ode4')

• Con el comando **simget** se obtiene la estructura de opciones o el valor de una propiedad:

opts=simget(model)

value=simget(model,nombre\_propiedad)

donde *model* es el nombre del modelo


- Aun cuando la dinámica de la mayoría de los sistemas físicos son no lineales, muchas técnicas útiles para el análisis y diseño de sistemas de control se basan en modelos lineales
- Ejemplos: diagramas de Bode y de lugar de raíces
- Las ecuaciones lineales del espacio de estado son:  $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$  $\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$

donde **x** es el vector de estado **y** u es el vector de entrada



- Simulink proporciona tres comandos para extraer la aproximación lineal del espacio de estado a partir de los modelos Simulink
  - linmod para sistemas continuos
  - linmod2 alternativo a linmod para reducir errores de truncamiento
  - dlinmod para sistemas continuos y discretos
- La sintaxis del comando linmod es
- [A,B,C,D]=linmod(model,X,U,par,xpert,upert)

donde model es el nombre del modelo



# Ejemplo: linealización de un modelo

Se tiene el siguiente sistema no lineal:  $x_1 = -x_2$  $\dot{x}_2 = x_1 + x_2^3 - 3x_2$ cuyo modelo (nolinear\_mod.mdl) en Simulink es: s s x1 Out x2 Gain x1 Integrator Integrator f(u) 2 x2 Out Fcn pow(u[1],3)-3\*u[1] En la ventana de comandos Matlab se ejecutan los comandos: >> [A,B,C,D]=linmod('nolinear\_mod'); >> disp(A) -1.0000()1.0000 -3.0000 >> disp(eig(A)) indica que el sistema es estable -0.3820-2.6180



- El comando trim encuentra el punto de equilibrio de un sistema mediante la búsqueda numérica del punto donde el máximo valor absoluto de la derivada de estado es minimizado
- La sintaxis del comando es

[x,u,y,dx] =

trim(model,x0,u0,y0,ix,iu,iy,dx0,idx,options,t)
donde model es el nombre del modelo y el resto opcionales

• Para la optimización de parámetros en la dinámica y control de sistemas se usa el Toolboox de Optimizac.



- Los callbacks son comandos Matlab que se ejecutan automáticamente cuando ocurren ciertos eventos (abrir un modelo, doble-click en un bloque, etc.)
- Se pueden añadir también callbacks para añadir animaciones
- Los callbacks están relacionados con el Handle Graphics de Matlab
- Los callbacks se pueden asociar con un modelo o con un bloque particular



- Permiten definir bloques personalizados usando código en Matlab, C, C++, Ada o Fortran
- Se pueden usar en varias situaciones: si existe código que modela una porción o la totalidad de un sistema, para describir el modelo dinámico de forma algorítmica en lugar de gráfica, para aumentar la eficiencia de una simulación y como mecanismo para añadir animaciones a un modelo Simulink
- Una función S se incluye en un modelo Simulink usando el bloque S-Function que se encuentra en la librería de bloques User-Defined Functions



- Simulink dispone de herramientas y técnicas para identificar y corregir errores de programación que pueden ser:
  - debidas a la construcción del modelo (sintaxis)
    - líneas de señal sin conectar, bloques de entrada y salida sin conectar y datos de configuración de bloque sin asignar
  - de implementación
    - overflows que causan errores de ejecución y lazos algebraicos que ralentizan la ejecución



# Encontrando errores de construcción del modelo

- Simulink puede detectar automáticamente errores durante el paso de compilación del modelo y lo reporta mediante la ventana de diálogo Simulation Diagnostics (simulación interrumpida) o mediante mensajes warning en la ventana de Matlab (simulación continúa)
- Para asegurar que todos los errores sean detectados se configura las opciones en Simulation:Configuration Parameters:Diagnostics



#### Encontrando errores de construcción del modelo



#### Simulink



### Encontrando problemas de implementación

- Un problema de implementación es una característica del modelo no deseada que puede no impedir su ejecución pero sea incorrecta o ineficiente
- Tipos de errores:
  - Errores de ejecución
  - Ejecución lenta
    - Oscilación por cruce de cero
    - Lazos algebraicos
    - Stiffnes
    - Complejidad del modelo
  - Resultados incorrectos

- Un debugger interactivo es una herramienta que permite controlar la ejecución de un programa y monitorizar los valores de las variables
- Para ejecutar el debugger seleccionar en la bbra de menús de la ventana del modelo Tools:Simulink Degugger
- La barra de herramientas de la ventana del debugger contiene una serie de botones (13) que permiten controlar el proceso de depuración



# Depurador (debugger) de Simulink

A Simulink Debugger: untitled	
날 🕶 🖿 🗰 2억 2월 4월 4월	La 2
Break Points Break/Display points Blocks	Outputs Sorted List Status
Remove selected point Break on conditions Zero crossings Step size limited by state Solver Error NaN values	% It stops before the initialization stage of the simulat % loop. The command line debugger is also active in the % Command Window. It is recommended that debugging action % be issued in the graphical debugger or the command line % debugger, but not both to avoid synchronization problem % % Use the Help button for additional help. %
Break at time :	



#### Botones del debugger





# Depurador (debugger) de Simulink

📣 Simulink Debugger: Orden2	
[ 음구 음고 음고 [ 등 다이 🕨 🖛 📕 💻 🖛	Œ ਦੂ⊡ 2⊡2 🔆 0 — 🚺 5 📍 Close
Break Points Simulation Loop	Outputs Sorted List Status
Break/Display points Blocks	[TM = 0 ] 0:1 Scope.Start 'Ord (sldebug 04):
Orden2/Scope 📃 🔽	Data of 0:1 Scope block 'Orden2/Scope': No data to display
Remove selected point	Trace: Data of 0:1 Scope block 'Orden2/Scope': U1 = [0]
Break on conditions	TM = 0 ] 0:2 Constant.Start '
Zero crossings	Data of 0:2 Constant block 'Orden2/Constant': No data to display
Solver Error	% [TM = 0 ] 0:3 Gain.Start 'Orde
Break at time :	(sldebug @6):



- Stateflow es una extensión de Simulink que proporciona un entorno muy potente para añadir máquinas de estado finito a los modelos de Simulink
- Se fundamenta en los diagramas Statechart que representan las máquinas de estado finito y los diagramas de transición de estado
  - Para incluir un bloque Chart en un modelo Simulink se selecciona el blockset Stateflow y arrastrar el bloque Chart en el modelo
  - Haciendo doble click en el bloque Chart se obtiene la ventana de Stateflow



- Real-Time Workshop es una extensión de Simulink que convierte los modelos Simulink en código ejecutable.
- RTW se puede configurar para producir código ejecutable para el computador en el que Simulink se ejecuta, otro computador, una tarjeta de procesado digital de señales o un controlador embebido
- El código producido se ejecuta más rápido que el modelo Simulink y su ejecución se puede controlar de forma remota



- Mastering Simulink, Dabney, Harman Prentice Hall
- System Simulation Techniques with MATLAB and Simulink, Xue, Chen - John Wiley & Sons