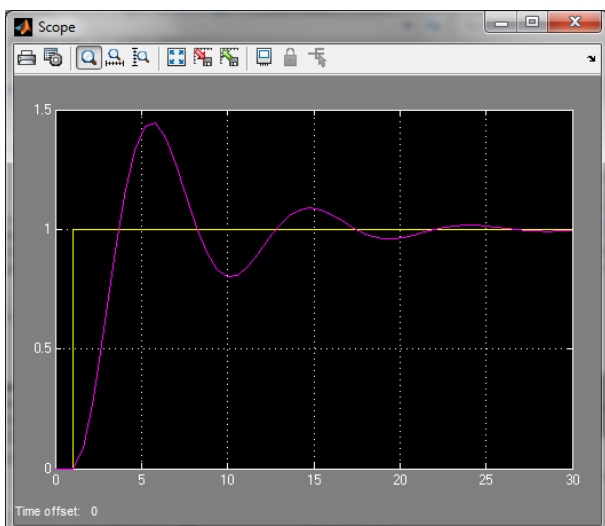
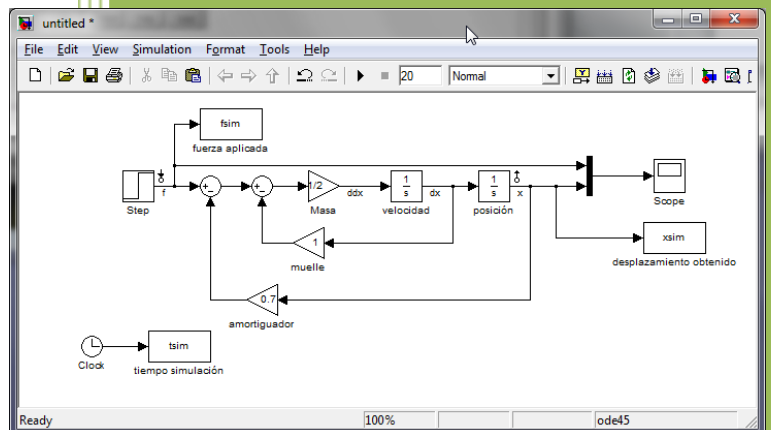
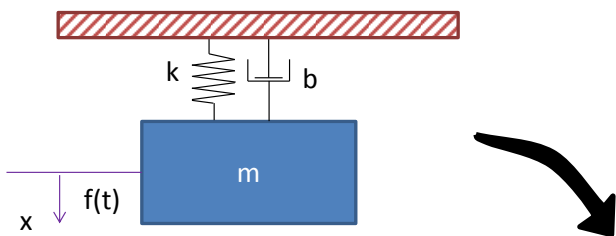


# Simulación de sistemas de control continuos con SIMULINK



## 0 Contenido

0	Contenido .....	2
1	MATLAB® y SIMULINK® .....	2
2	Uso de SIMULINK .....	3
2.1	Inicio del software .....	3
2.2	Creación de un modelo.....	3
2.3	Control de la simulación y visualización de resultados. ....	5
2.4	Biblioteca de componentes .....	6
2.5	Abrir un modelo existente .....	8
3	EJEMPLO 1. Amortiguación de un vehículo .....	9
4	EJEMPLO 2. Tanque calefactado.....	15
5	EJEMPLO 3. Introducir una FDT en simulink.....	18
6	EJEMPLO 4. Calentador. Modelo no lineal y lineal.....	19
7	Ejercicio 5. Creación de un modelo de simulación .....	21

## 1 MATLAB® y SIMULINK®

MATLAB (MATrix LABoratory) es un lenguaje de alto nivel y un entorno interactivo para cálculo numérico, visualización y programación. Permite al ingeniero analizar datos, desarrollar algoritmos y crear modelos y aplicaciones propias. Sus puntos fundamentales son:

- Entorno interactivo para la exploración, diseño y solución de problemas.
- Amplia librería de funciones para cálculo científico que permiten trabajar de forma natural con problemas de álgebra lineal, estadística, análisis de Fourier o cálculo numérico.
- Biblioteca de funciones gráficas para la visualización en 2D y 3D de datos. Incluye herramientas que permiten la generación de gráficos personalizados.
- Lenguaje de programación de alto nivel.
- Una gran variedad de librerías adicionales (toolboxes) especializadas en campos específicos de la ingeniería y la ciencia.

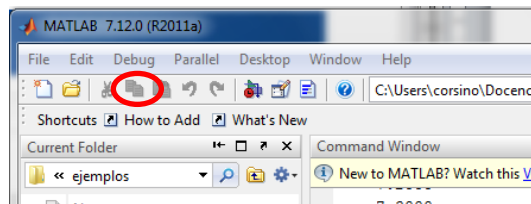
SIMULINK es una herramienta incluida en MATLAB para el modelado, simulación y análisis de sistemas dinámicos. El interfaz principal es una herramienta gráfica para la descripción del sistema mediante diagramas de bloques. Además, permite su extensión mediante la creación de librerías con nuevos tipos de bloques. Está altamente integrado con el resto de MATLAB. Es una herramienta muy usada en ingeniería de control y procesamiento de señal para la simulación de sistemas y el diseño basado en modelos.

## 2 Uso de SIMULINK

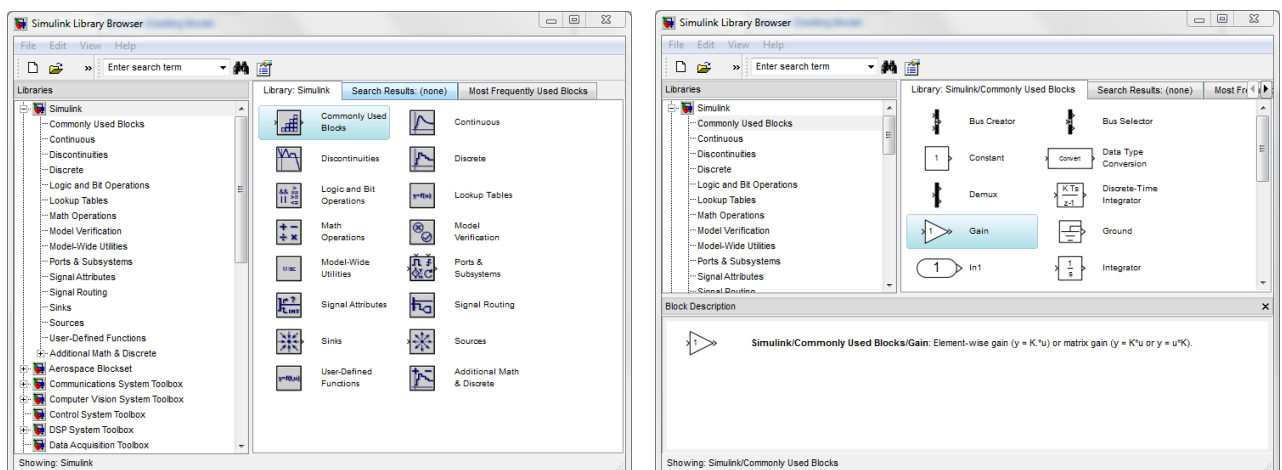
### 2.1 Inicio del software

El primer paso consiste en cargar el “Simulink Library Browser”, para lo que es necesario estar ejecutando MATLAB previamente. Una vez arrancado MATLAB, podemos:

1. Ejecutar el comando *simulink* desde la línea de comandos de MATLAB.
2. Pulsar el botón “Simulink Library” que está en la barra de herramientas de MATLAB



Una vez cargado, aparecerá una ventana del Browser. Si es la primera vez que lo hacemos dentro de la sesión actual de MATLAB, la ventana puede tardar unos segundos en aparecer.



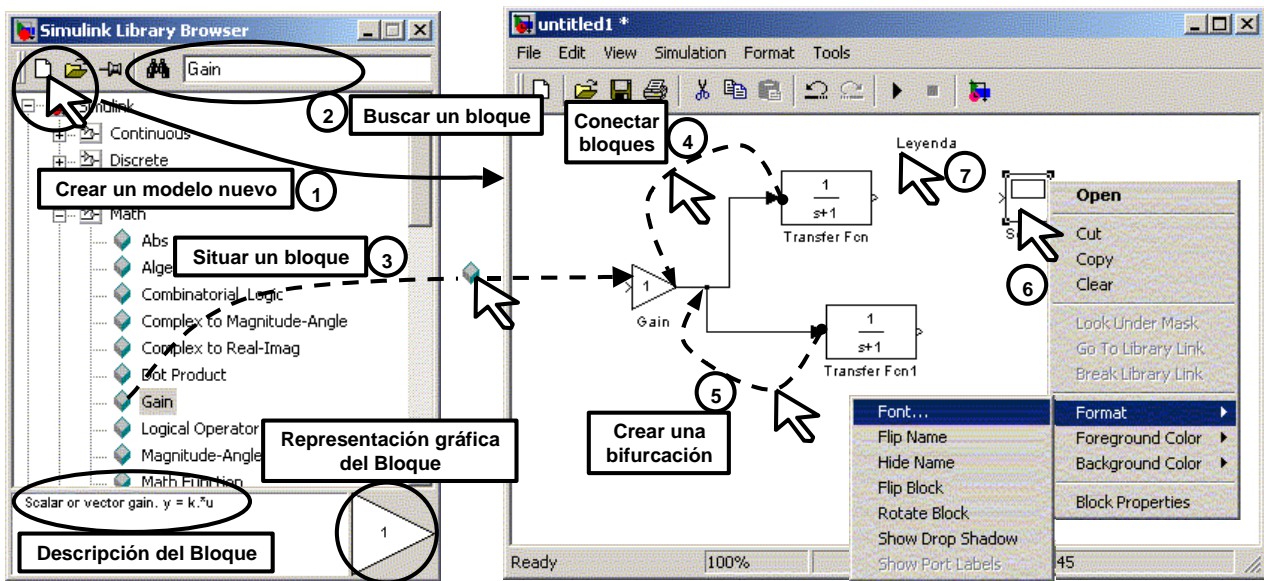
Si queremos que esta ventana esté siempre visible, podemos seleccionar la opción: **View > Stay on Top**. También podemos activar la opción **View > Show Block Descriptions** para abrir un panel en el que se visualiza una descripción del bloque seleccionado.

### 2.2 Creación de un modelo

Para crear un modelo propiamente dicho, debemos abrir una ventana del editor de SIMULINK:

1. En el menú del Simulink Library Browser o en la ventana principal de MATLAB, seleccionamos **File > New > Model**, lo que abrirá un modelo vacío. También es posible crear un nuevo modelo pulsando el icono “new file” en la barra de herramientas del Simulink Library Browser.
2. Para salvar el modelo, seleccionamos **File > Save** o **File > Save As** en el editor de Simulink. Si todavía no le hemos dado un nombre al modelo, siempre nos aparecerá el cuadro de diálogo Save As para introducir el nombre del fichero y seleccionar el directorio en el que queremos guardarlo.

Para simular un sistema, se deben insertar en las ventanas de simulación los distintos componentes con los que se va a construir el modelo. Se pueden seguir los siguientes pasos:

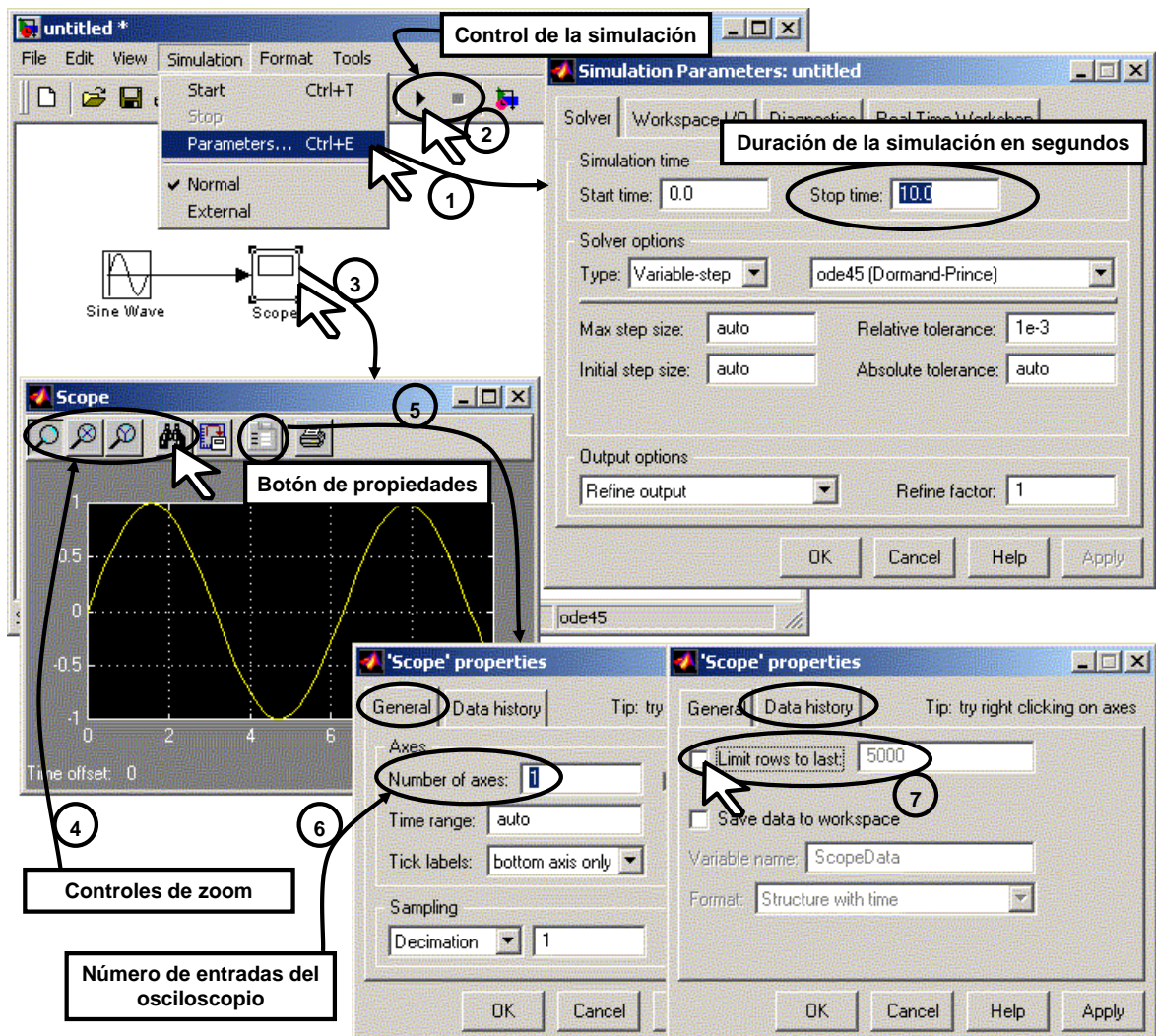


- 1) **Crear un nuevo modelo:** Para abrir una nueva ventana de simulación se debe pulsar el botón "nuevo modelo".
- 2) **Buscar un bloque:** Se puede buscar un bloque expandiendo el árbol de la biblioteca o buscándolo directamente por su nombre en la ventana de búsqueda. En este caso, si hay más de un bloque que pueda corresponder a ese nombre, irán apareciendo a medida que se pulse la tecla "enter" (retorno).
- 3) **Situación un bloque:** Para situar un bloque, se mantiene pulsado el botón izquierdo del ratón sobre el icono en forma de rombo que hay junto al nombre del bloque y se arrastra hacia la posición deseada en la ventana de simulación.
- 4) **Conectar bloques:** En cada bloque, las puntos de salida aparecen indicados mediante una flecha saliente del bloque " $|>$ ", mientras que los puertos de entrada a cada bloque se indican con una flecha entrante al mismo " $>|$ ". Se conecta la entrada de un bloque a la salida de otro, manteniendo pulsado el botón izquierdo del ratón mientras se arrastra desde el símbolo de entrada de uno de los bloques hasta el de salida de otro o viceversa.
- 5) **Crear una bifurcación:** Si se desea llevar la salida de un bloque a la entrada de más de uno se necesita crear una bifurcación en la conexión. Para hacerlo, se arrastra con el ratón desde la entrada del nuevo bloque a conectar hasta la línea de la conexión que se va a bifurcar.
- 6) **Modificar los bloques:** Se pueden rotar o aplicar simetrías a los bloques usados, según convenga la colocación de entradas/salidas para el esquema que se esté realizando, pulsando sobre él el botón derecho del ratón y utilizando los menús desplegables o mediante la opción "Format" del menú principal ("Format\Flip Block", "Format\Rotate Block", etc.). También mediante los menús o haciendo doble clic sobre el bloque, se pueden modificar sus parámetros.
- 7) **Inserción de textos:** Se puede incluir un texto aclaratorio o informativo en cualquier parte de la ventana del modelo, haciendo doble clic en una zona libre y escribiendo directamente el texto.

También se pueden cambiar los nombres y posiciones de los bloques que se empleen para la simulación antes o después de conectarlos. Asimismo los enlaces de las conexiones pueden moverse o modificarse. Para eliminar cualquier elemento basta con seleccionarlo con un clic y eliminarlo con la tecla "sup" o "delete", o utilizar alguno de los menús.

Conviene guardar ("File\Save as") periódicamente el modelo, incluso antes de terminarlo, para evitar perder el trabajo realizado.

Un ejemplo trivial incluiría la selección de dos componentes: "Simulink\Sources\Sine Wave" y "Simulink\Sinks\Scope" de la ventana "Simulink Library Browser", y el arrastre de los mismos hasta la ventana de dibujo. En el caso de nuestro ejemplo básico, para conectar el generador de señales y el osciloscopio, simplemente se debe situar el ratón sobre el punto de salida del generador, pulsar el botón izquierdo, arrastrar el ratón hasta el punto de entrada del osciloscopio y soltar el botón del ratón.

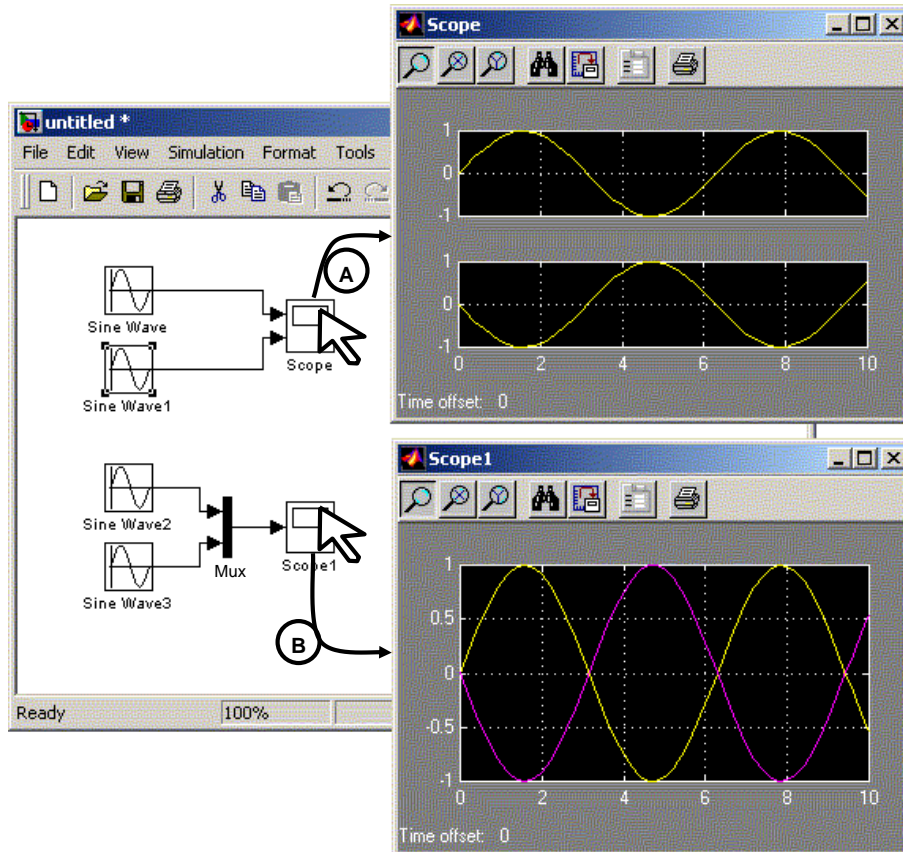


### 2.3 Control de la simulación y visualización de resultados.

Antes de poder ejecutar la simulación, es necesario seleccionar los parámetros apropiados para la misma (1). Desde el menú: "Simulation\Parameters", se puede desplegar un cuadro de dialogo, en el que se controlan parámetros de la simulación de entre los cuales el que se modifica más habitualmente es el tiempo final de la simulación. (Otros parámetros accesibles son el tiempo de inicio de la simulación, el método matemático que se empleará para llevarla a cabo, o las variables que se tomarán/guardarán de/en el espacio de trabajo). La simulación se puede poner en marcha o detener mediante el menú anterior o los botones de la ventana (2).

Para visualizar los resultados de la misma son muy útiles los bloques se encuentran en el grupo "Sinks" de la biblioteca de Simulink. De entre ellos, quizás el más útil es el bloque "Scope" que simula el comportamiento de un osciloscopio. Tras realizar una simulación se pueden ver los resultados que ha registrado haciendo un doble

clic sobre él (3). Para ver correctamente los resultados se utilizan los controles de zoom (4), siendo conveniente pulsar siempre tras una simulación el botón de autoescala (el de los prismáticos) para ver el total de los datos registrados. Los otros tres botones de zoom permiten respectivamente ampliar un área señalada con un arrastre del ratón, ampliar el eje "X" de la misma manera o ampliar el eje "Y".



El bloque "Scope" tiene una serie de propiedades a las que se accede a través del botón correspondiente de la ventana "Scope" (5). Dos de las más útiles son la que permite elegir el número de entradas que se desean para el osciloscopio (6) "Number of axes" (que será también el número de gráficos que representará) y la que determina si el osciloscopio almacena todos los datos de la simulación o sólo los últimos obtenidos (7) "Limit rows to last". Respecto a este último control, es conveniente eliminar la marca "√" del cuadradito blanco para que el osciloscopio mantenga todos los datos registrados durante la simulación completa.

Si se desea visualizar más de una señal en un osciloscopio, existen dos posibilidades:

- Aumentar el número de entradas del osciloscopio como se comentó anteriormente.
- Utilizar un bloque "Mux" para que ambas señales aparezcan en el mismo gráfico.

## 2.4 Biblioteca de componentes

Todos los componentes básicos de Simulink, se pueden encontrar en su biblioteca de componentes. A continuación se describen los componentes básicos de la biblioteca de Simulink:

**Continuous:** Bloques que pueden ser representados como una función continua en el tiempo.

**Derivative:** La salida del bloque se corresponde con la derivada de la entrada.

**Integrator:** La salida del bloque se corresponde con la integral de la entrada. Los parámetros del bloque

permiten controlar el valor inicial de la salida, así como la existencia de límites superiores e inferiores en la salida.

**Transfer Fcn:** Permite expresar una función de transferencia mediante su expresión en la variable compleja  $s$ . Sus parámetros son los polinomios del numerador y del denominador de la función, expresados como vectores fila.

**Transport Delay:** La salida del bloque se corresponde con la entrada al mismo retrasada una cantidad de tiempo, que se fija como parámetro en el bloque.

**Zero-Pole:** Función de transferencia expresada en función de la ganancia en régimen permanente, y la situación de los polos y ceros del sistema.

### **Math: Bloques que realizan operaciones matemáticas sobre sus entradas.**

**Abs:** Calcula el valor absoluto de su entrada.

**Gain:** Aplica una ganancia constante a la entrada.

**Math Function:** Este bloque incluye la mayor parte de las funciones matemáticas típicas, con la excepción de las funciones trigonométricas.

**Product:** Calcula el producto escalar de sus entradas. Un parámetro del bloque permite regular el número de entradas del mismo.

**Sign:** Calcula el signo de la entrada. +1 indica positivo, -1 negativo, y 0 un valor nulo.

**Sum:** Calcula la suma de todas sus entradas. Un parámetro permite indicar el número de entradas, y si estas deben ser invertidas antes de la suma. Ejemplo: un valor para el parámetro "+++" indicaría que el bloque tiene 4 entradas, y que la tercera de ellas debe ser invertida antes de sumarla.

**Trigonometric Function:** En este bloque se incluyen todas las funciones trigonométricas típicas.

### **Nonlinear: Bloques no lineales.**

**Dead Zone:** Incluye una *zona muerta* en el sistema, centrada en torno a cero. El sistema no responde ante estos valores. La magnitud de la zona muerta puede ser modificada, y echa asimétrica por medio de los parámetros del sistema.

**Relay:** La salida pasa al estado on=1 cuando la entrada supera un valor umbral, y a estado off=0 cuando cae por debajo de un umbral distinto. El estado inicial es off.

**Saturation:** La señal de salida no sobrepasa un valor umbral, configurable con los parámetros del bloque.

**Switch:** Una entrada del sistema permite escoger cual de las otras dos entradas se presenta en la salida.

### **Signals&Systems: Manejo de sistemas y señales.**

**Subsystem:** Permite la realización de sistemas jerárquicos. Al abrir el subsistema, nos permite incluir en su interior, nuevos bloques constructivos, e incluso anidar nuevos subsistemas.

**In1:** Por defecto un subsistema no contiene entradas. Por cada entrada que se desee añadir se le debe incluir uno de estos bloques.

**Out1:** Por defecto un subsistema no contiene salidas. Por cada entrada que se desee añadir se le debe incluir uno de estos bloques.

**Mux:** Permite la inclusión de un conjunto de señales en una única línea de transmisión (que transmite datos vectoriales), lo que facilita la representación en el dibujo. Parámetros: número de entradas. Admite tanto entradas escalares como vectoriales.

**Demux:** Permite la descomposición de los datos puesto en forma vectorial en una línea mediante un multiplexador. Parámetros: número de salidas.

**Data Store Memory:** Define una variable del entorno de trabajo que se va a usar como lugar de almacenamiento de datos útil para evitar tener que hacer conexiones complejas que compliquen el diagrama de bloque que se está usando.

**Data Store Read:** Lee el valor actual de una variable de almacenamiento, que debe estar previamente definida mediante un bloque Data Store Memory

**Data Store Write:** Cambia el valor actual de una variable de almacenamiento, que debe estar previamente definida mediante un bloque Data Store Memory

## Sinks: Sumideros de señales.

**Display:** Representa numéricamente el valor de una variable.

**Scope:** Representa gráficamente la evolución en el tiempo de una variable.

**To Workspace:** Guarda el valor de la señal indicada en una variable del entorno de trabajo del *Matlab*. Se puede escoger el nombre de la misma, y limitar su tamaño.

**To File:** Guarda en un fichero de tipo ".mat" los datos de la señal de entrada al bloque.

**Stop Simulation:** Detiene la simulación si el valor de la entrada es distinto de 0.

## Sources: Fuentes de señales.

**Chirp Signal.** Genera una señal senoidal, modulada en frecuencia, entre un valor inicial y un valor final.

**Clock:** Tiempo que se lleva de simulación.

**Constant:** Proporciona una señal de valor constante.

**From Workspace:** Proporciona una secuencia de datos tomadas del entorno de trabajo del *Matlab*. La variable elegida debe contener una matriz indicando los valores de la señal, y los instantes en los que la señal toma estos valores.

**From File:** Proporciona datos tomados de un fichero ".mat", en el que debe estar el valor de la variable, junto a los instantes de tiempo en que toma cada valor.

**Pulse Generator:** Genera una onda cuadrada, de la que se puede controlar la amplitud, el periodo y el tiempo de *duty* (relación entre el tiempo que la onda toma su valor máximo y el tiempo que toma el valor mínimo).

**Ramp:** Genera una señal de tipo *Rampa*.

**Random Number:** Genera números aleatorios distribuidos según una función normal.

**Signal Generator:** Simula un generador de señales electrónico, permitiendo generar ondas dientes de sierra, ondas cuadradas y senoidales.

**Sine Wave:** Generador de ondas senoidales.

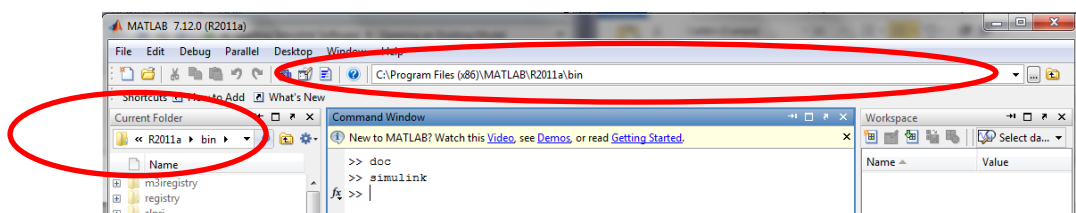
**Step:** Genera una señal de tipo escalón.

**Uniform Random Number:** Genera números aleatorios distribuidos según una función uniforme.

## 2.5 Abrir un modelo existente

Los ficheros que contienen un modelo de Simulink se guardan con la extensión *.mdl*. Para abrir un modelo existente podemos:

1. Seleccionar la opción **File > Open** en el Simulink Library Browser o en la ventana principal de MATLAB. Navegar hasta el directorio que contiene el archivo y seleccionarlo.
2. Desde la línea de comandos de MATLAB, escribir el nombre del archivo que contiene el modelo sin incluir la extensión. Por ejemplo para abrir el modelo ejemplo.mdl, deberíamos escribir ejemplo. Es necesario asegurarse previamente que el directorio que contiene el modelo es el directorio de trabajo o está incluido en el path de búsqueda de MATLAB.
  - a. Para cambiar el directorio de trabajo se puede pulsar la barra de navegación incluida en la barra de herramientas de MATLAB o la que aparece en el panel "Current Folder".

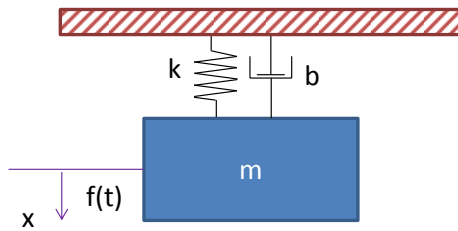


- b. Utilizar el comando **cd** desde la línea de comandos de MATLAB.



### 3 EJEMPLO 1. Amortiguación de un vehículo

Consideremos el sistema formado por la amortiguación de un vehículo. Dicho sistema puede modelarse mediante un conjunto masa-muelle-amortiguador como el representado en la figura:



$$\ddot{x} = \frac{1}{m} (f(t) - b\dot{x} - kx)$$

$$G(s) = \frac{X(s)}{F(s)} = \frac{1}{2s^2 + bs + k}$$

Usaremos como parámetros del sistema:  $m=2\text{Kg}$ ,  $b=0.7\text{Ns/m}$  y  $K=1\text{N/m}$ .

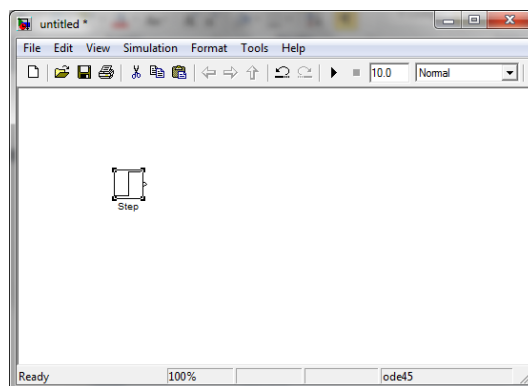
*Introducción del modelo utilizando las ecuaciones*

En primer lugar crearemos una nueva ventana para el modelo. Los bloques que necesitaremos son:

Bloques necesarios	Localización en el Simulink Library Browser
Step	Sources
Gain	Math Operation
Sum	Math Operation
Integrator	Continuous
Scope	Sinks
To Workspace	Sinks
Mux	Signal Routing
Clock	Sources

Para introducir las ecuaciones, debemos tener en cuenta que la posición ( $x$ ) es la integral de la velocidad ( $\dot{x}$ ) y que la velocidad es la integral de la aceleración ( $\ddot{x}$ ). También debemos tener en cuenta que la entrada a nuestro sistema es la fuerza  $f(t)$  y la salida la posición  $x(t)$ .

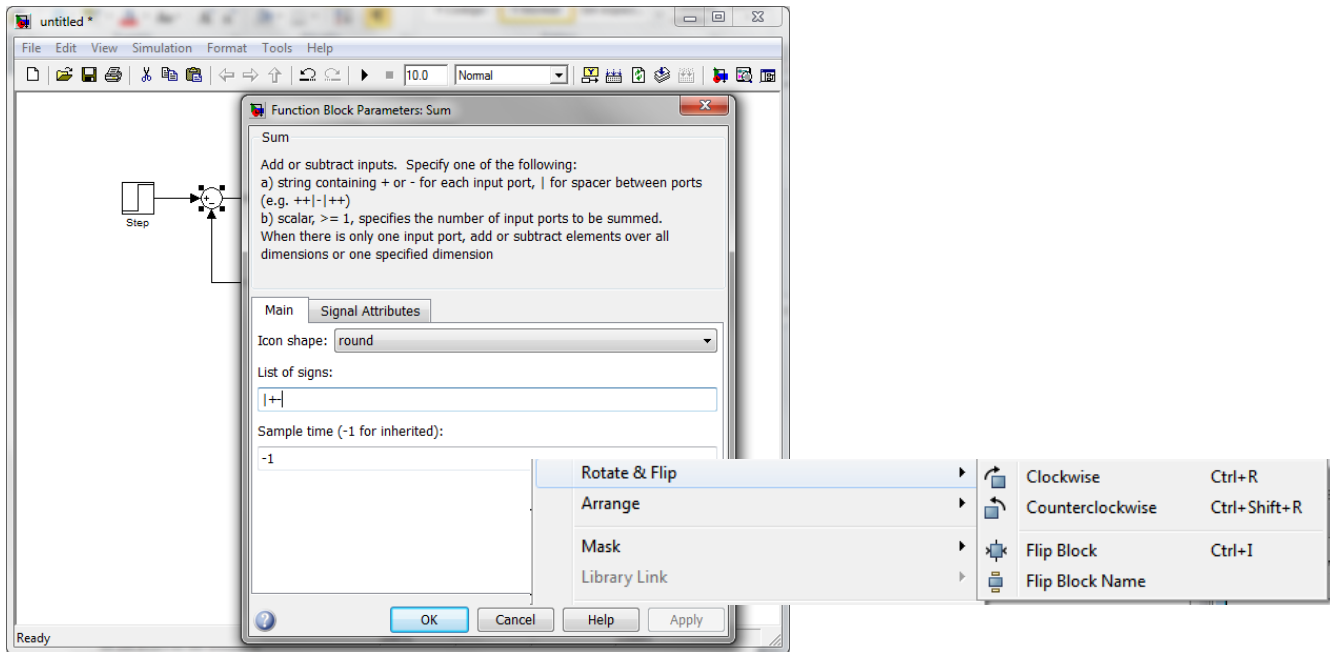
Para generar la fuerza en nuestro modelo, usaremos el bloque step que simula un cambio de tipo escalón en el valor de la fuerza aplicada.



La derivada de mayor orden que aparece es la aceleración, que se obtiene:

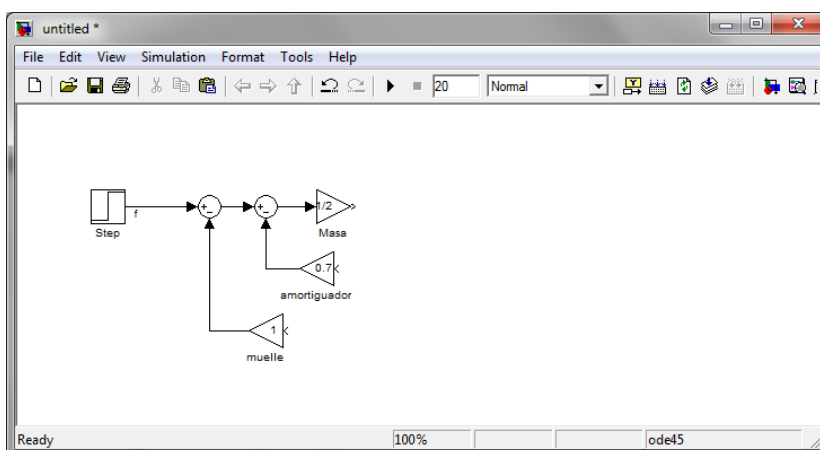
1. restando a la fuerza aplicada las fuerzas generadas por el muelle (proporcional a la posición) y por el amortiguador (proporcional a la velocidad).
2. Dividiendo el balance anterior por la masa del bloque.

Para incluir un bloque sumador con una entrada positiva y una negativa, arrastraremos el bloque sum al modelo y haremos una doble pulsación sobre el elemento para abrir la ventana de parámetros del bloque. En el cuadro List of signs, sustituiremos la cadena “|++” por “|+-” para indicar que la primera posición no se usa, la segunda es una entrada positiva y la tercera una entrada inversora. Como tenemos que hacer un balance de tres cantidades, usaremos dos bloques sum.

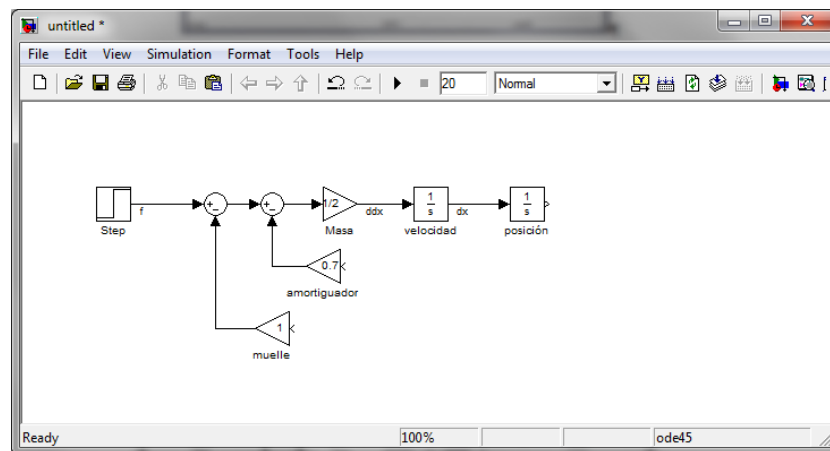


En el primer bloque sumador introduciremos la fuerza debida al muelle. Para ello, arrastramos un bloque de tipo gain al modelo. Para cambiar su orientación podemos seleccionar el bloque y pulsar **Ctrl+I** o **Ctrl+R** o usar la opción de menú **Rotate & Flip > Flip Block**.

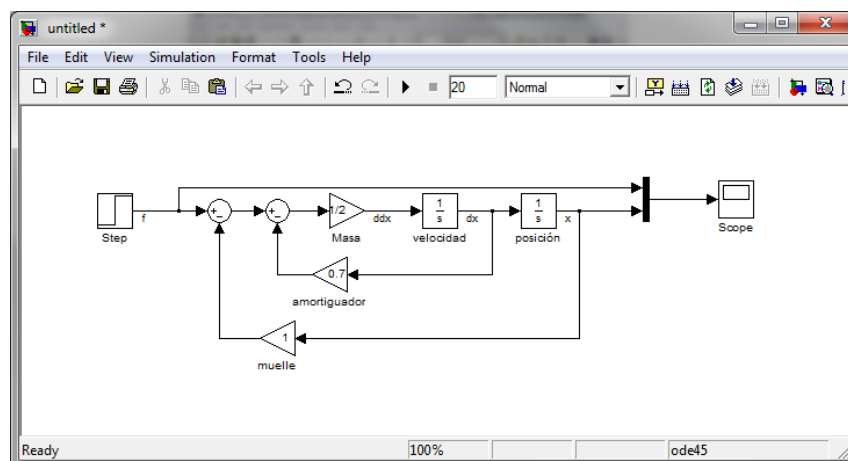
Por defecto el valor del bloque de ganancia es 1, que coincide con el valor de k en este ejemplo. Podemos cambiar el nombre del bloque pulsando dos veces sobre el mismo e introduciendo el nuevo texto, por ejemplo “muelle”. De la misma forma, introduciremos otro bloque ganancia para modelar el amortiguador. Para modificar la ganancia del bloque pulsaremos dos veces sobre el mismo y el cuadro de edición gain escribiremos 0.7. La salida del bloque que representa el amortiguador la conectaremos a la entrada inversora del segundo sumador. A la salida del segundo sumador tenemos ahora la fuerza resultante que actúa sobre la masa, si la dividimos por la masa del bloque (usando un bloque gain con ganancia 1/m) tendremos la aceleración a la que está sometido el bloque.



A partir de la aceleración, podemos obtener la velocidad por integración. Para ello, arrastramos un bloque de tipo Integrator al modelo. De la misma manera, podemos obtener la posición a partir de la velocidad.

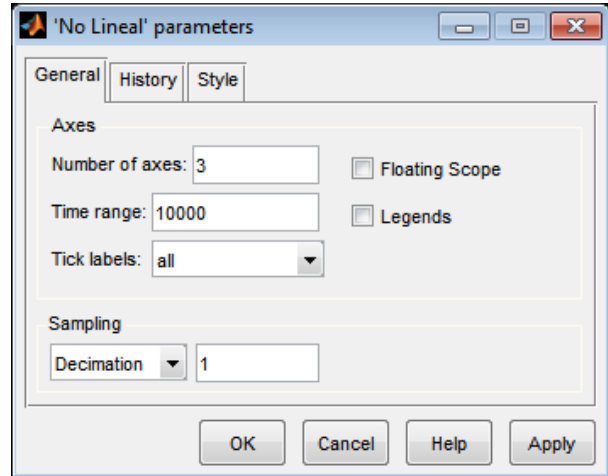
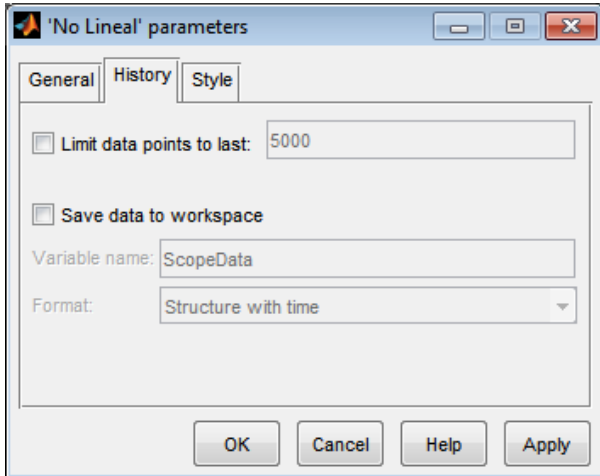


Ahora ya podemos llevar a las entradas de los bloques que representan el muelle y el amortiguador las señales correspondientes (posición y velocidad). Si queremos nombrar las señales basta con hacer una doble pulsación del botón izquierdo del ratón sobre la flecha que representa la señal y escribir el nombre en el recuadro que aparece. Para completar el modelo, incluiremos un osciloscopio en el que representaremos la fuerza aplicada y la posición de salida. Para ello incluiremos un bloque de tipo scope y otro de tipo mux en el modelo. El bloque **mux** es un bloque que multiplexa varias entradas en una única salida. A la primera entrada del bloque mux (la superior) conectaremos la fuerza aplicada y la segunda el desplazamiento obtenido. La salida del multiplexador se conecta a la entrada del osciloscopio.

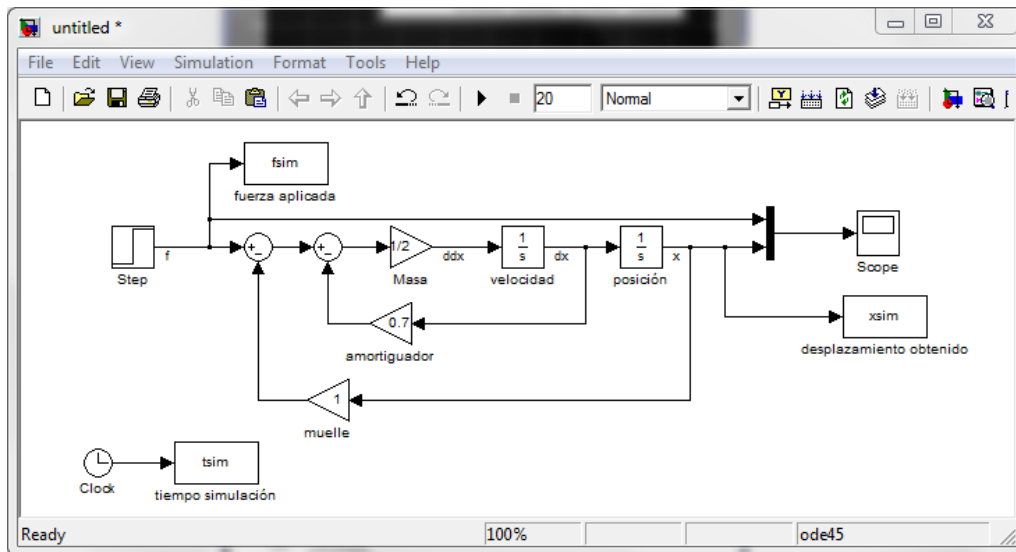


Podremos adaptar el dibujo creado al tamaño de la pantalla pulsando la barra espaciadora, así nos evitaremos andar con los zoom.

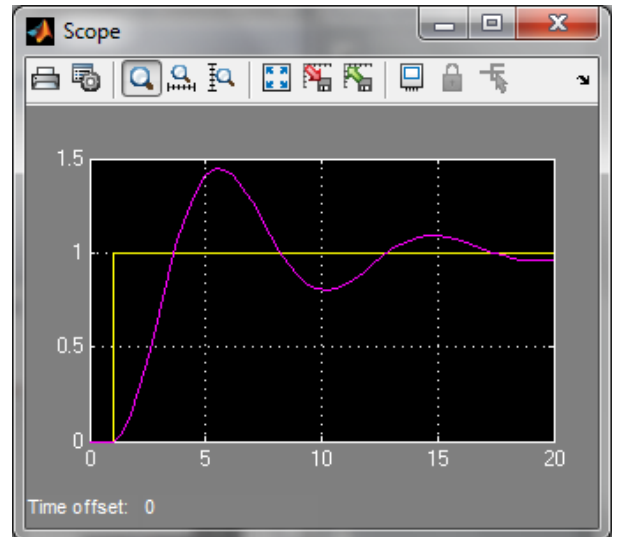
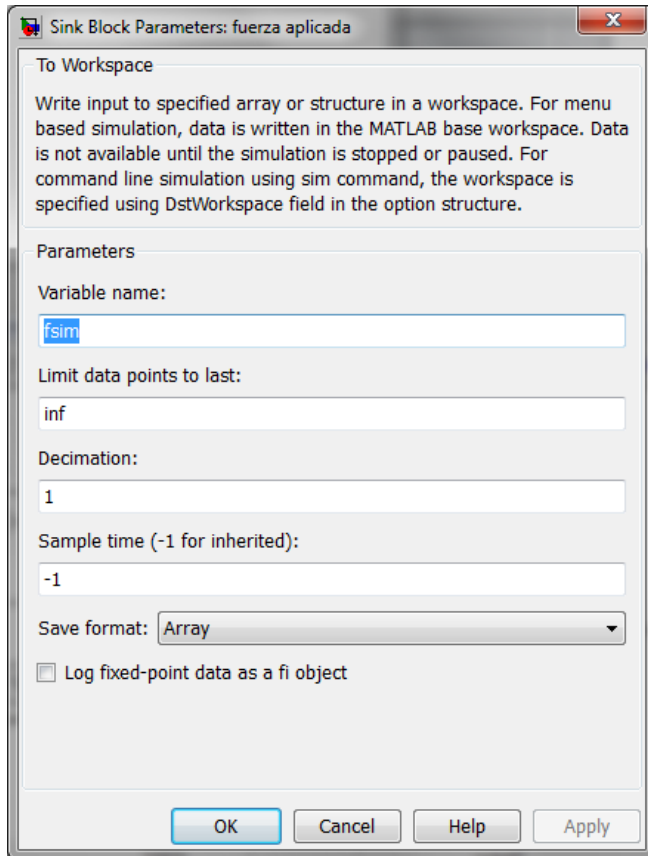
Para configurar el osciloscopio, pulsaremos dos veces sobre él con el botón izquierdo del ratón. En la ventana que aparece, pulsaremos el icono parámetros (a la derecha del icono de impresora) y en la ventana de configuración que aparece seleccionamos la pestaña "History". Desactivaremos la opción que limita los puntos a los últimos 5000. De este modo podremos ver toda la simulación. En la pestaña "General", especificaremos el número de ejes necesarios para representar nuestras señales.



Por último, incluiremos bloques del tipo ToWorkspace con el fin de salvar en el espacio de trabajo de Matlab los resultados de la simulación. Los encontraremos en la librería "Sinks". Para poder salvar en una variable los tiempos correspondientes a cada muestra de la simulación también es necesario incluir un bloque de tipo clock. El aspecto final del modelo es:



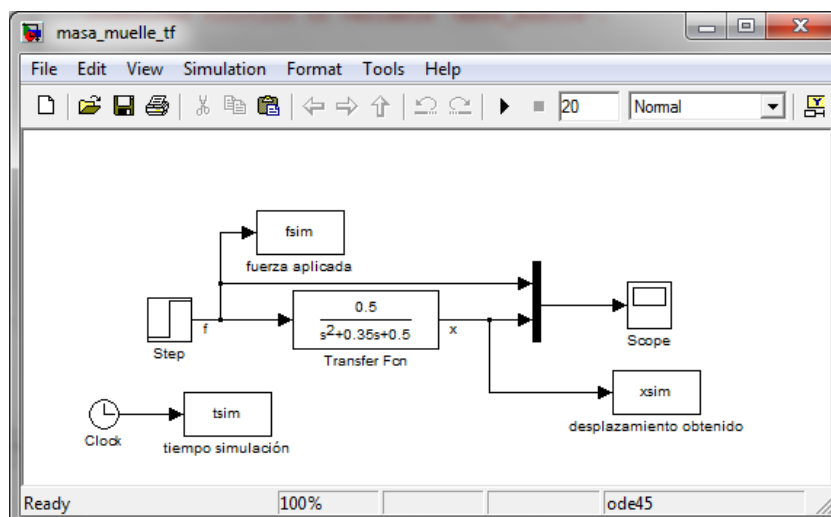
La configuración de los bloques ToWorkspace es como la que se muestra en la siguiente figura. En variable name pondremos el nombre de la variable en la que queremos guardar el resultado (por ejemplo fsim) y en el apartado Save Format elegiremos Array.



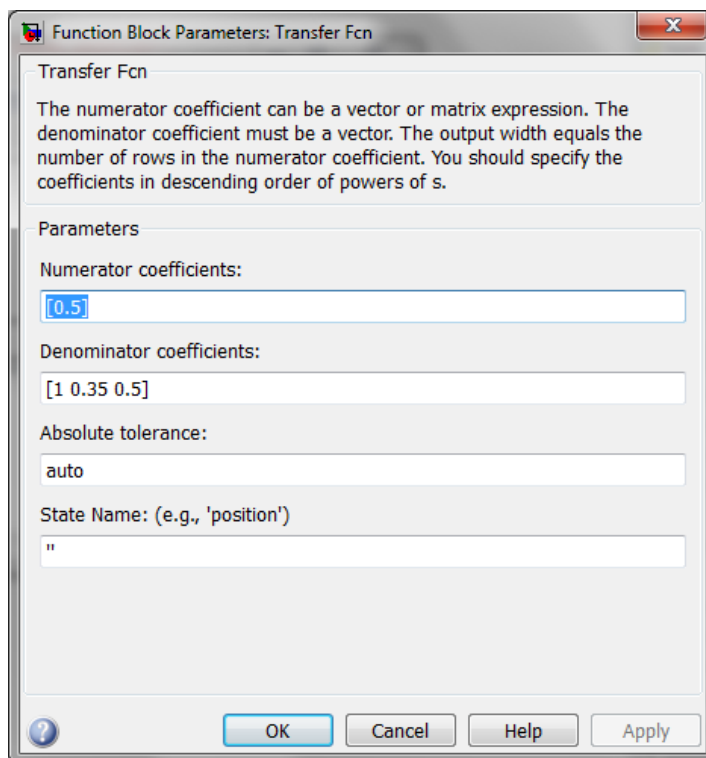
Para simular el modelo, ajustaremos el tiempo de la simulación a un valor adecuado a la dinámica de nuestro sistema (en este caso elegimos 20s) y pulsamos el botón play en la barra de herramientas del modelo. El resultado es el mostrado en la figura:

### *Simulación directa de la función de transferencia del modelo*

Si hubiésemos obtenido directamente la función de transferencia, podríamos simularla directamente usando un bloque de tipo Transfer Fcn que encontraremos en el grupo Continuous. El modelo que obtendríamos sería como el mostrado en la figura:

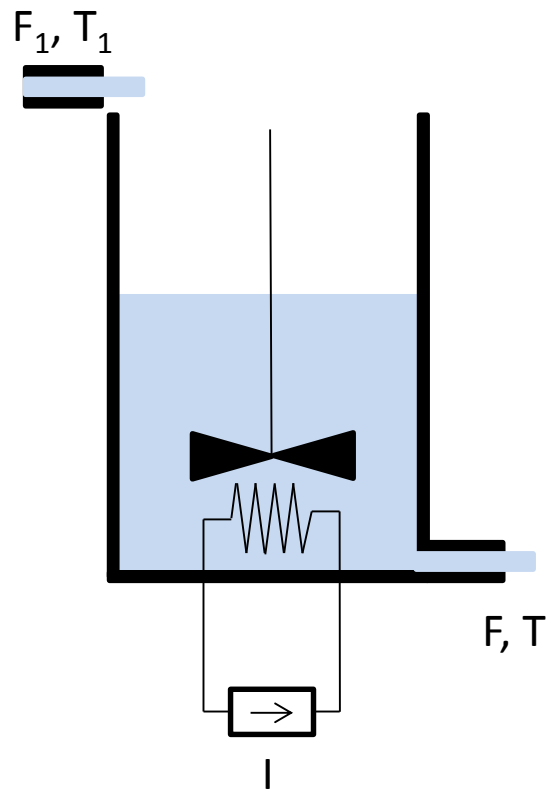


Para configurar el bloque con la función de transferencia, es suficiente con introducir dos vectores con los coeficientes del numerador y denominador de la función de transferencia.



## 4 EJEMPLO 2. Tanque calefactado

Consideraremos un tanque de agua calefactado de sección cuadrada de 0.2 m de lado y una altura máxima de 1.5m. Tenemos un caudal de entrada  $F_1$  con una temperatura  $T_1$ . El calentamiento se produce por una resistencia  $R$  de  $10 \Omega$ , alimentada por una fuente que proporciona una corriente de  $I$  Amperios. La salida se produce por un escape libre situado en la base del depósito. El depósito dispone de un sistema que asegura que la mezcla sea homogénea, por lo que podemos suponer que la temperatura del líquido del depósito es igual a la temperatura del líquido de salida.



Las ecuaciones del sistema son:

**Balance de masa:** 
$$\frac{dv(t)}{dt} = F_1(t) - F(t)$$

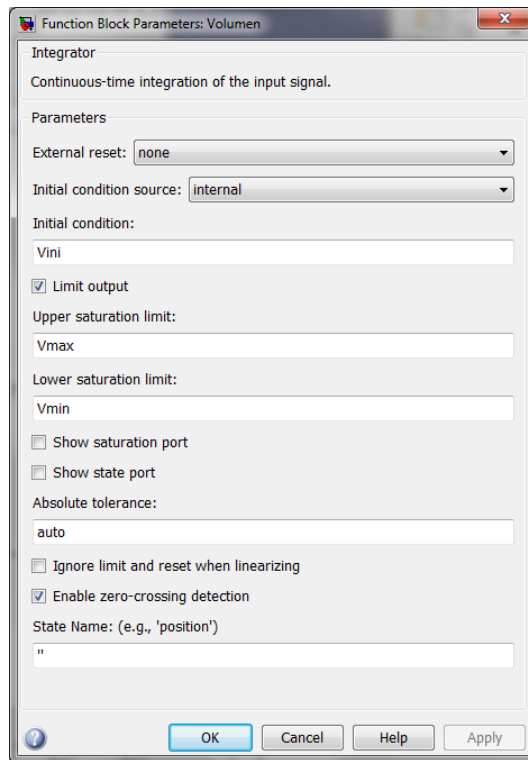
**Descarga por gravedad:** 
$$F(t) = K\sqrt{h(t)}$$

**Balance de energía:** 
$$V\rho c \frac{dT(t)}{dt} = F_1\rho c(T_1 - T) + RI^2$$

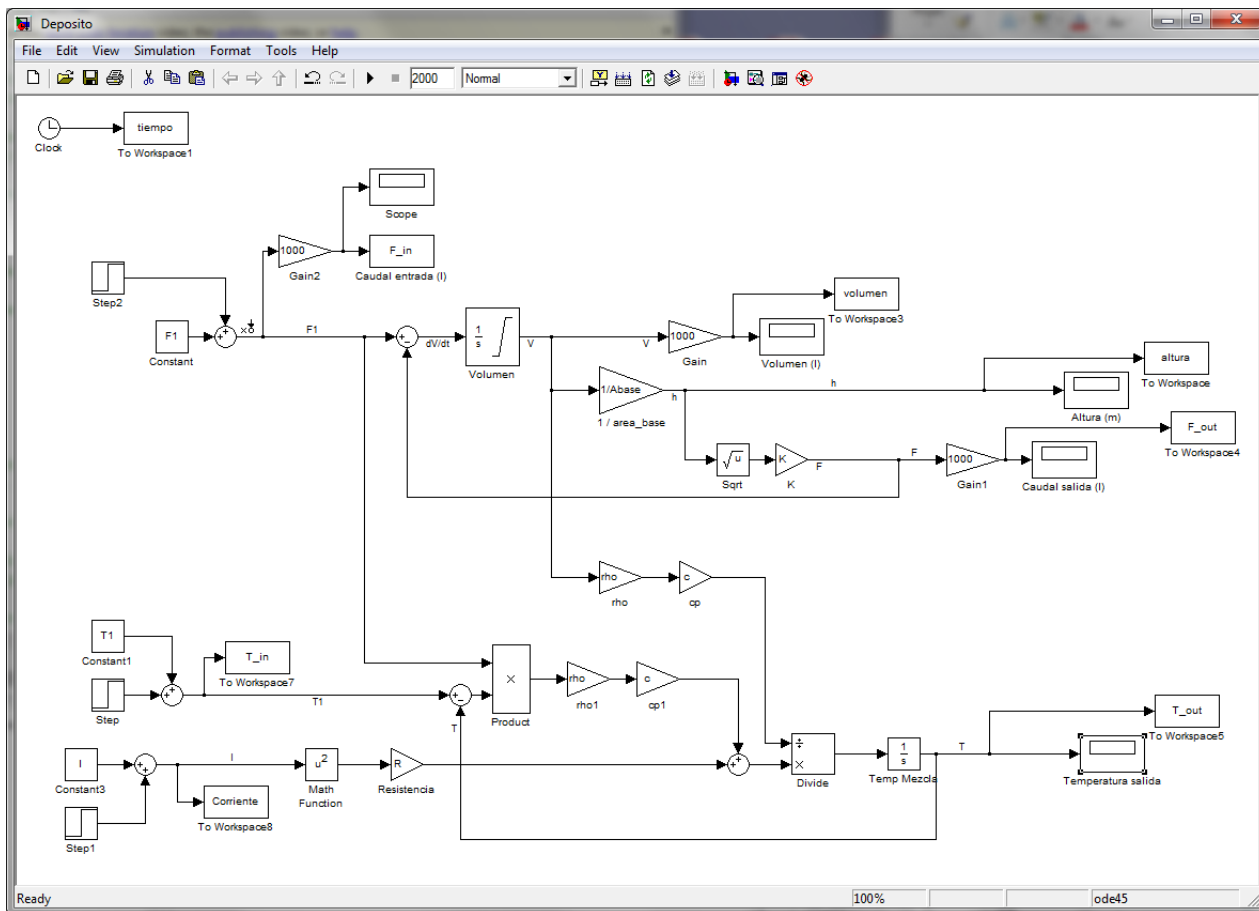
Consideraremos que el calentador está perfectamente aislado por lo que las pérdidas al exterior por calor son despreciables.

Para introducir este modelo, usaremos como bloques adicionales:

Bloques necesarios	Localización en el Simulink Library Browser
Constant	Sources
Product	Math Operations
Math Function	Math Operations
Sqrt	Math Operations



El modelo resultante es:





Este ejemplo es no lineal (saturación en el depósito, productos de variables, raíces cuadradas, ...). Por ello, al configurar los integradores debemos incluir el estado inicial:

Los bloques de tipo constante incluyen los valores en el punto de operación elegido, en este caso  $T_1=15^\circ\text{C}$ ,  $h=0.8\text{m}$  e  $I=40\text{A}$ .

Por comodidad, los parámetros del modelo los hemos definido como variables y su inicialización la hemos incluido en un "script de MATLAB" que se muestra a continuación:

```
% Iniciacion de variables para el modelo del tanque

% Parámetros del sistema

% propiedades del agua (se consideran constantes)
rho=1e3; % densidad del agua en Kg/m^3
c=4186.8; % calor especifico del agua en J/(Kg°C)

% Cte de proporcionalidad del caudal de desagüe
K=0.23e-3; % m^3/(sm^(1/2))

% Resistencia del calentador
R=10; % Ohmios

% Tanque
Abase=0.2*0.2; % Area de la base en m^2

Vmax=1.5*Abase; % volumen máximo en m^3
Vmin=0;

% Punto de equilibrio
T1=15; % °C
h=0.8; % m
I=40; % Amperios

% en la condición de equilibrio las derivadas son cero
% de conocer la altura, sabemos el caudal de salida
F=K*sqrt(h);
% Como el volumen no varía
F1=F;
% El volumen correspondiente es:
Vini=0.8*Abase; % este parámetro es muy importante para inicializar el integrador

% Con el balance térmico
Tini=(R*I^2+F1*rho*c*T1)/(F1*rho*c);
```

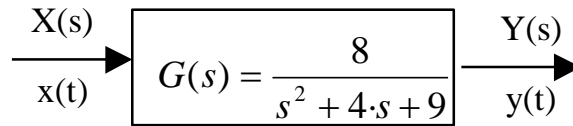
## Actividades obligatorias

Para el modelo del tanque, se debe calcular:

1. Indicar cómo evolucionan todas las variables de salida cuando:
  - a. El caudal de entrada se incrementa en 0.05 litros.
  - b. La corriente que se inyecta en la resistencia se incrementa en 5 amperios.
  - c. La temperatura del líquido de entrada aumenta  $3^\circ\text{C}$ .
2. Repetir los cálculos para el punto de funcionamiento definido por  $T_1=18^\circ\text{C}$ ,  $I=42\text{ A}$ ,  $F=0.23\text{ l/s}$

## 5 EJEMPLO 3. Introducir una FDT en simulink

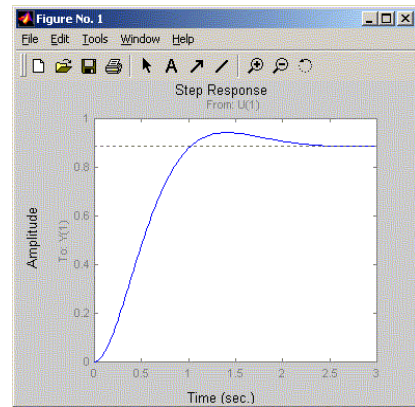
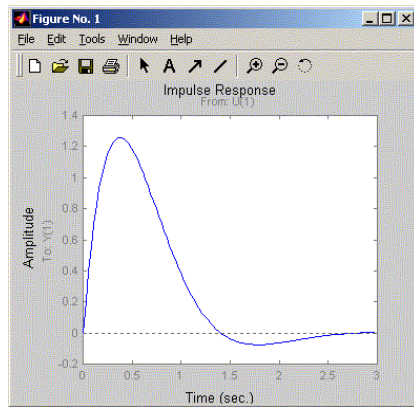
Simular la respuesta de un sistema descrito por su función de transferencia ante una determinada señal de entrada con Matlab o Simulink es muy sencillo



Con Matlab se puede obtener la respuesta ante un impulso de Dirac, un escalón, etc.:

```
>> g = tf([8], [1 4 9])
g=8/(s^2+4s+9)
```

```
>> step(g)
```



Con Simulink se construye el modelo del sistema, pudiendo optarse por introducir sus parámetros de forma explícita o dejarlos como parámetros a los que se les puede asignar un valor como variables desde la ventana de comandos de Matlab. Una vez ejecutada la simulación se obtendrá en el bloque “Scope” el resultado de la misma.

## 6 EJEMPLO 4. Calentador. Modelo no lineal y lineal

Si se desea simular un sistema más complejo basta con trasladar las ecuaciones de su modelo a un modelo de bloques de Simulink.

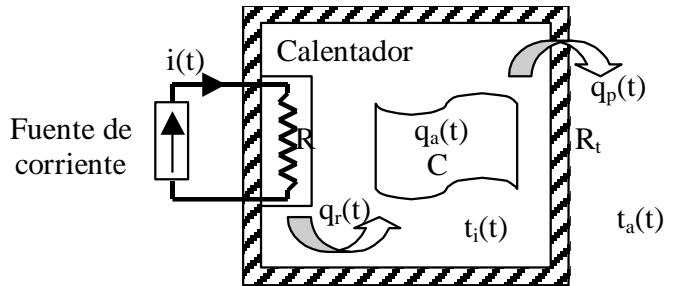
Modelo del sistema:

$$q_r(t) = q_a(t) + q_p(t)$$

$$q_r(t) = R \cdot i^2(t)$$

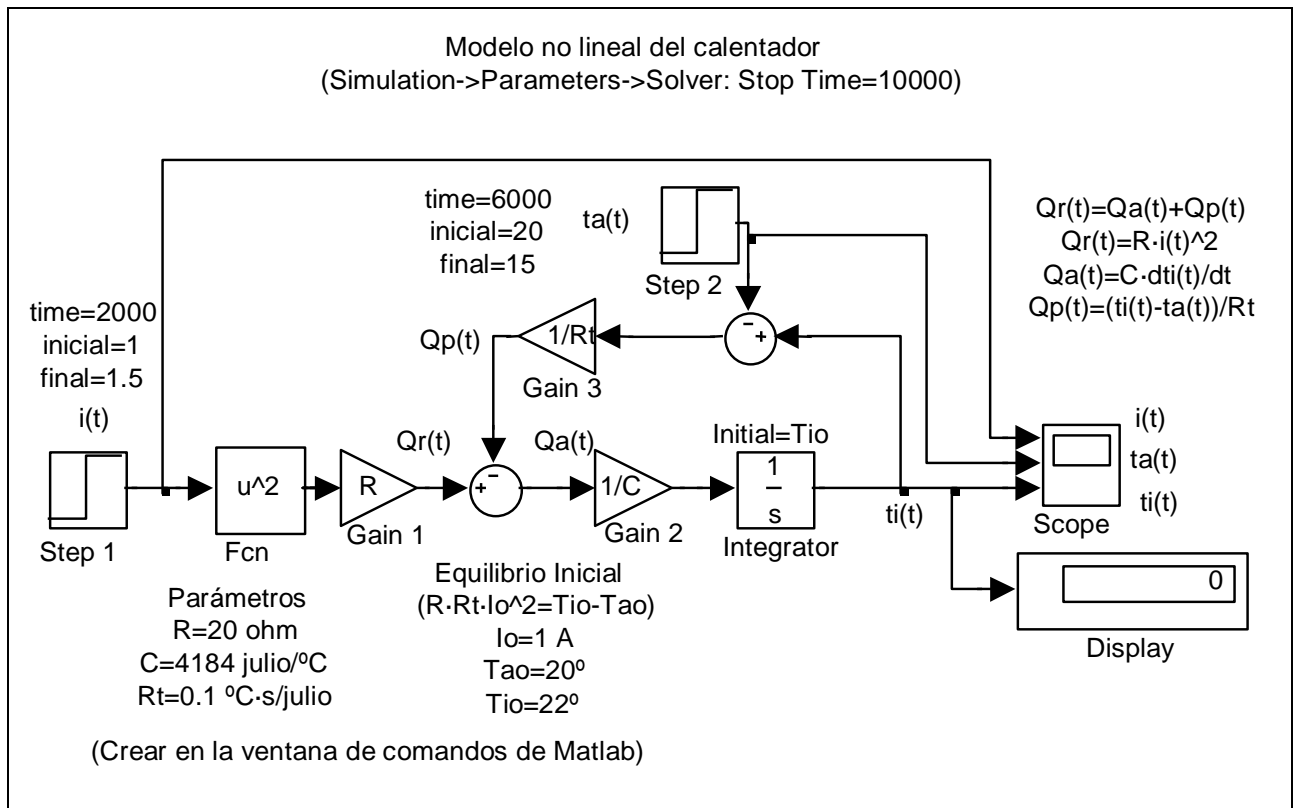
$$q_a(t) = C \cdot dt_i(t)/dt$$

$$q_p(t) = (t_i(t) - t_a(t)) / R_t$$



Simulink puede simular sistemas no lineales, como se puede ver en el siguiente modelo, siendo necesario en muchos casos definir los valores iniciales de algunas de las variables del sistema (en bloques como los "integrator"). Los valores del modelo representado que se han dejado como parámetros son asignados en la ventana de comandos de Matlab (debe tenerse cuidado con el uso de mayúsculas y minúsculas en la denominación de estos parámetros). Los archivos de ejemplo indicados contienen el modelo representado y el espacio de trabajo con los valores asignados a los parámetros para la ventana de comandos de Matlab.

Archivos: [calentador\\_no\\_lineal.mdl](#), [calentador.mat](#)



También se puede linealizar el modelo respecto a un punto de funcionamiento y construir el modelo en transformadas de Laplace. Luego se trasladan las ecuaciones del modelo linealizado a un modelo de

Simulink mediante, por ejemplo, bloques "Función de Transferencia" (TransferFcn).

Modelo linealizado del sistema en Transformadas de Laplace:

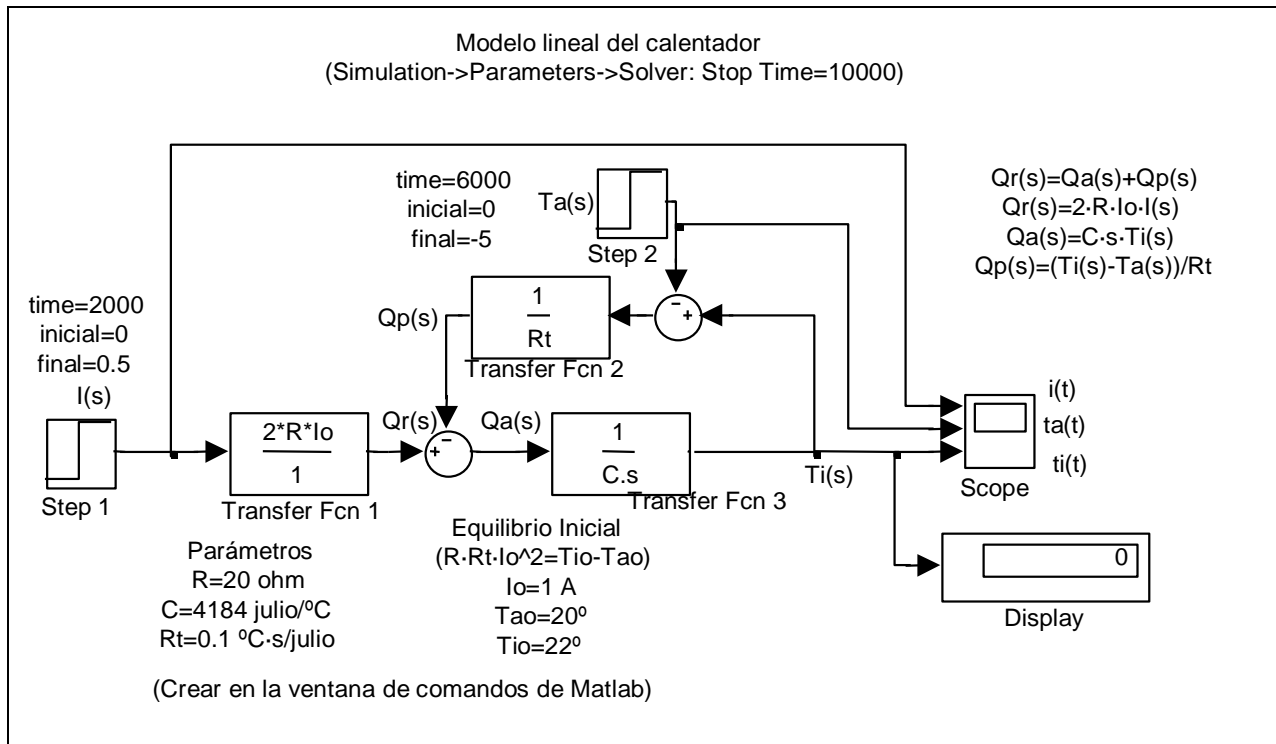
$$Q_r(s) = Q_a(s) + Q_p(s)$$

$$Q_r(s) = 2 \cdot R \cdot I_o \cdot I(s)$$

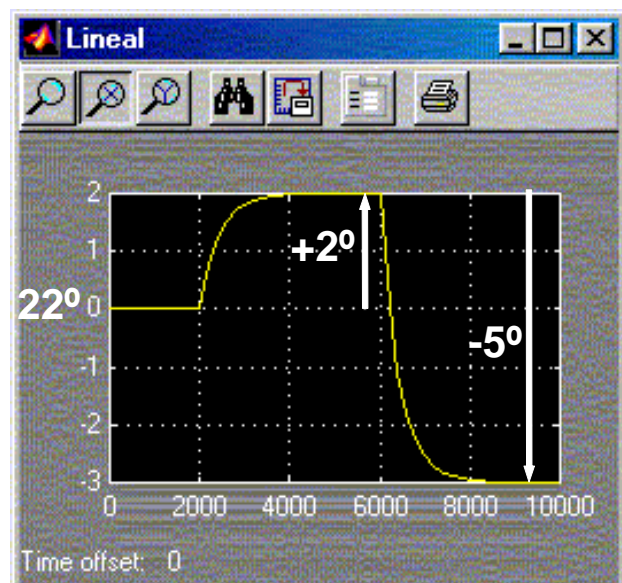
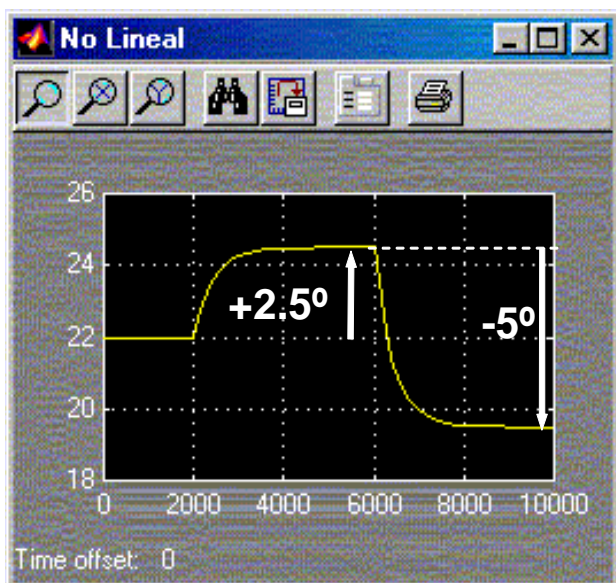
$$Q_a(s) = C \cdot s \cdot T_i(s)$$

$$Q_p(s) = (T_i(s) - T_a(s)) / R_t$$

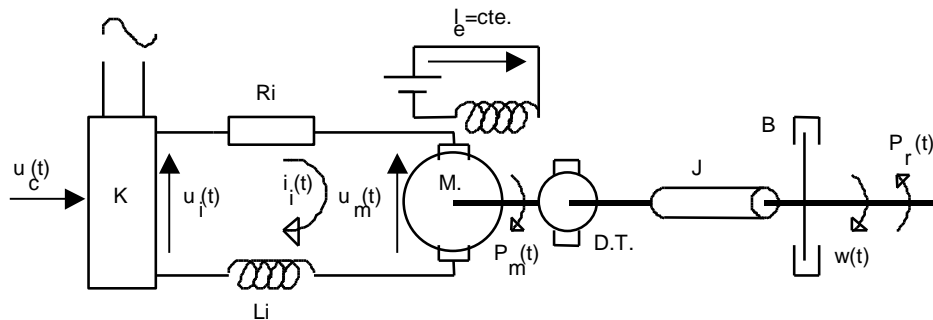
Archivos: calentador\_lineal.mdl, calentador.mat



Al comparar los resultados de la simulación de ambos modelos, se puede observar el error que introduce la linealización del modelo.



## 7 Ejercicio 5. Creación de un modelo de simulación



El sistema de la figura está compuesto por los siguientes elementos:

- Un rectificador controlado que alimenta a un motor CC con una tensión continua  $u_i(t)$  proporcional a la tensión  $u_c(t)$ , con constante de proporcionalidad  $K$ .
- Un motor CC con corriente de excitación constante de parámetros:  $R_i$ ,  $L_i$ ,  $cte.$ , contraelectromotriz  $K_b$  y  $cte.$  de par del motor  $K_p$ .
- El conjunto rotor-eje del motor y de la D.T. tiene una inercia  $J$  y un coeficiente de rozamiento viscoso  $B$  (que se representan en la figura). En el extremo del eje existe un par resistente variable,  $p_r(t)$ , debido a los elementos mecánicos que mueve el motor y que no aparecen en la figura.

### Modelo matemático del sistema:

$$u_i(t) = K \cdot u_c(t)$$

$$u_i(t) = R_i \cdot i_i(t) + L_i \cdot di_i(t)/dt + u_m(t)$$

$$u_m(t) = K_b \cdot w(t)$$

$$p_m(t) = K_p \cdot i_i(t)$$

$$p_m(t) = p_r(t) + J \cdot dw(t)/dt + B \cdot w(t)$$

### En transformadas de Laplace:

$$U_i(s) = K \cdot U_c(s)$$

$$U_i(s) = R_i \cdot I_i(s) + L_i \cdot s \cdot I_i(s) + U_m(s)$$

$$U_m(s) = K_b \cdot W(s)$$

$$P_m(s) = K_p \cdot I_i(s)$$

$$P_m(s) = P_r(s) + J \cdot s \cdot W(s) + B \cdot W(s)$$

### Con

$$K = 1$$

$$K_b = 0.11 \quad [\text{V} \cdot \text{s}/\text{rad}]$$

$$K_p = 1.6 \quad [\text{N} \cdot \text{m}/\text{A}]$$

$$R_i = 6 \quad [\Omega]$$

$$L_i = 0.23 \quad [\text{H}]$$

$$J = 17 \quad [\text{Kg} \cdot \text{m}^2]$$

$$B = 4.2 \quad [\text{Kg} \cdot \text{m}^2/\text{s}]$$

Construir el modelo del sistema en Simulink considerando a  $u_c(t)$  como entrada,  $p_r(t)$  como perturbación y  $w(t)$  como salida.

Dibujar **el diagrama de bloques del modelo en Simulink utilizado** para obtener los datos que se solicitan a continuación, **los resultados numéricos** correspondientes, y los **dibujos a mano alzada de los gráficos solicitados**

1. Dibujo del diagrama de bloques usado en las simulaciones.
2. Simular la respuesta del sistema a un escalón de cinco unidades en la entrada  $U_c(t)$ .
  - a. Obtener el valor al que tiende la respuesta en régimen permanente.
  - b. Determinar la constante de tiempo  $T$  del sistema.
3. Simular el efecto de un escalón de cinco unidades en la perturbación una vez que la respuesta anterior ha alcanzado el régimen permanente.
  - a. Dibujar aproximadamente la respuesta.
  - b. Obtener el valor al que tiende la respuesta en régimen permanente.
4. Simular la respuesta del sistema ante una entrada senoidal en  $U_c(t)$  de pulsación  $\omega=1$  rad/s y amplitud 5.
  - a. Obtener el valor de la amplitud de la senoide de la salida una vez alcanzado el régimen permanente.
  - b. Obtener la diferencia de fase entre la entrada y la salida una vez alcanzado el régimen permanente.