

# Programas de Simulink y Matlab

**Visualización en MATLAB los  
resultados de una simulación  
en SIMULINK**

Ing. Eddie Ángel  
Sobrado Malpartida

## 1. Objetivo

- Determinar y Analizar la respuesta en el dominio del tiempo de los sistemas de primero y segundo orden a entradas de tipo impulso, escalón y rampa.

## 2. Introducción

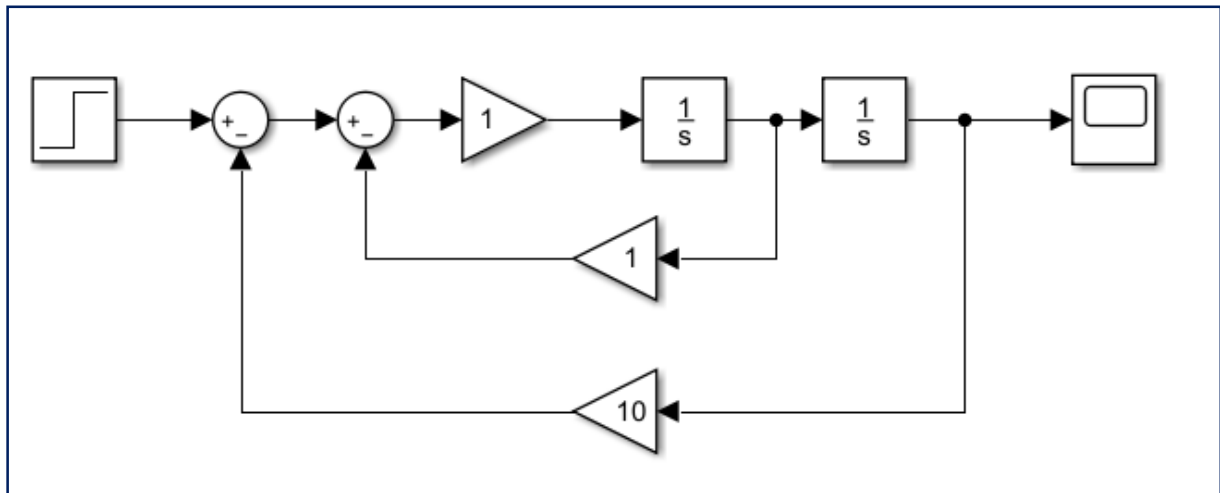
En la práctica previa se ha utilizado el elemento **Scope** para visualizar los valores que toma una señal cualquiera en un esquema Simulink. Sería interesante disponer, no sólo de la visualización de la señal sino también de sus **valores numéricos**, para poder trabajar posteriormente sobre ellos (por ejemplo, para poder calcular el valor máximo y mínimo de una señal y los instantes de tiempo en que se producen, etc).

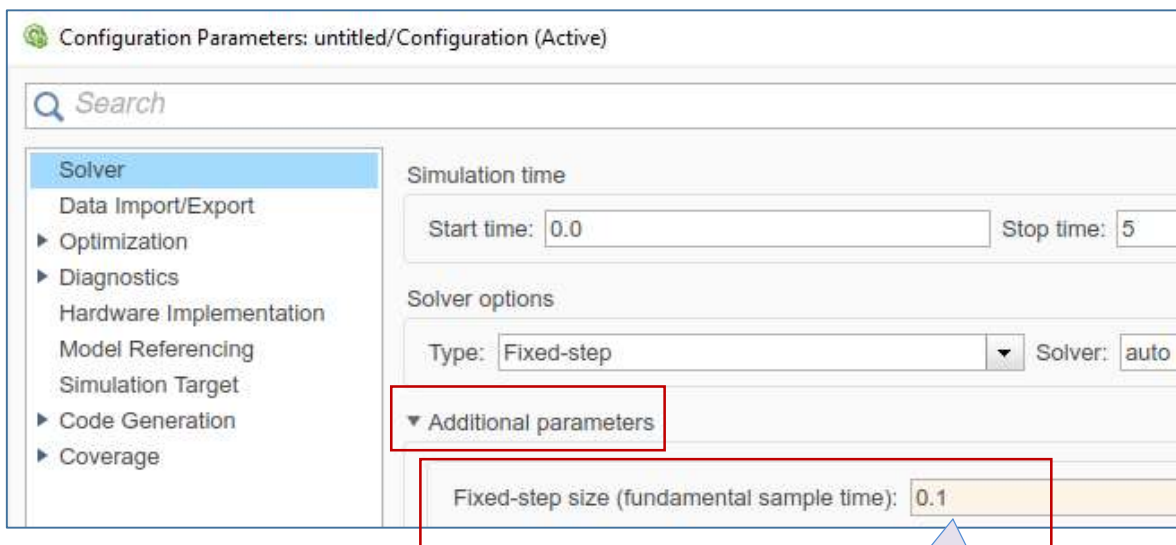
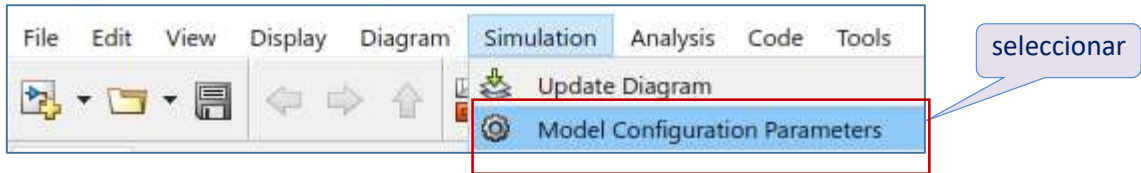
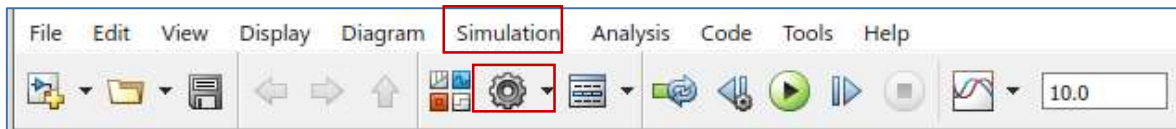
El elemento **Scope** permite, al mismo tiempo que representa los resultados de una simulación, guardar esos resultados en una variable de Matlab, de modo que se puedan consultar sus valores desde la ventana de comandos.

## III. Procedimiento

Implementemos el esquema que se muestra a continuación. En el siguiente sistema usaremos los bloques:

- *Gain*
- *Integrator*
- *Scope*
- *Step*
- *Sum*

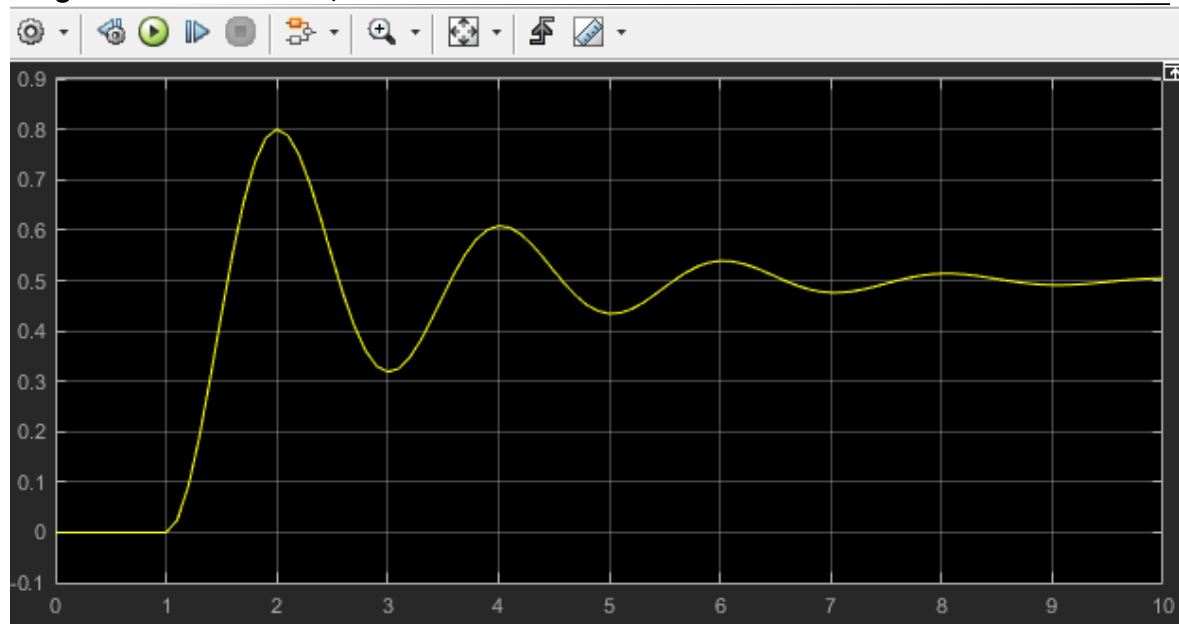




Se realizará la simulación de este esquema con los siguientes parámetros:

- Tiempo de simulación: 10 segundos.
- Amplitud del escalón de entrada: 5 unidades.

El resultado debería ser igual al que muestra la siguiente figura:



## Bloque STEP



Block Parameters: Step

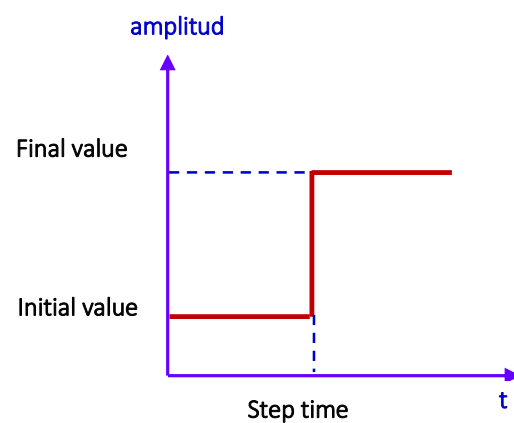
Step  
Output a step.

Parameters

Step time:  
1

Initial value:  
0

Final value:  
5

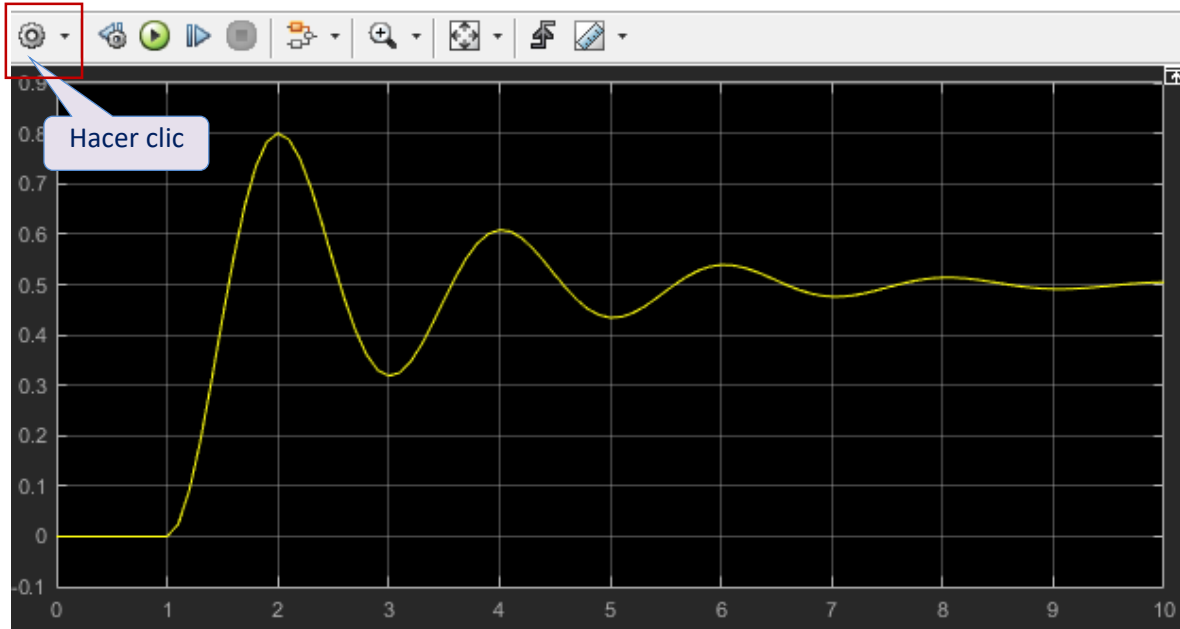


- **Step time:** Por defecto el escalón (cambio de magnitud) se producirá en el instante uno (estará retrasado). Si desea lo podrá colocar en cero el parámetro **step time**
- **Initial value:** El valor inicial del escalón también será cero
- **Final value:** el valor final indica la amplitud final, que en este caso particular es cinco. por defecto el valor es la unidad.

## Almacenando valores en variables desde el SCOPE

---

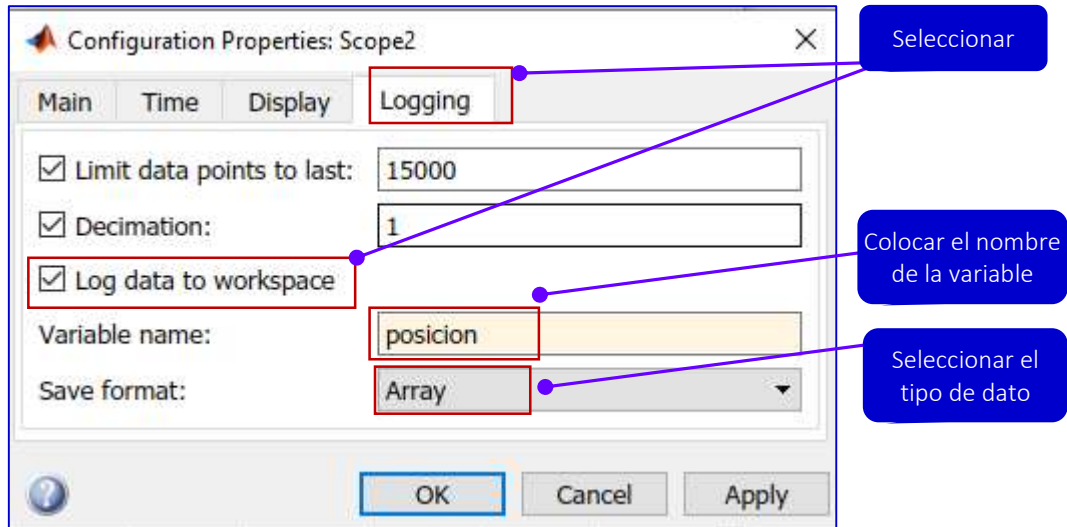
Ahora haremos click sobre el botón **Parameters** del elemento **Scope**, para abrir la ventana de configuración:



En la ventana de configuración deberemos seleccionar la pestaña **Logging** y en ella marcaremos la opción '**Log data to workspace**' para guardar los datos en el espacio de trabajo de Matlab en una variable.

Será necesario indicar dos parámetros:

- En primer lugar, tendremos que asignar un **nombre a la variable** en la que deseamos guardar los datos. Por defecto, esta variable es '**ScopeData**', pero nosotros cambiaremos ese nombre por '**posicion**' (sin acento para evitar problemas) que es el dato que almacena la información mostrada en el SCOPE
- En segundo lugar, será necesario especificar un formato para los datos. Simulink ofrece varias posibilidades: '**Structure with time**', '**Structure**', '**Array**', etc. Nosotros elegiremos el formato **ARRAY**, que quiere decir que los datos se guardarán en forma de vector.



Una vez hecho esto, lanzaremos de nuevo la simulación. El resultado sobre la ventana de **Simulink** será el mismo, pero en la ventana de comandos de **Matlab** podremos comprobar cómo se han creado dos nuevas variables. Para ello bastará escribir en la ventana de comando:

```
>> whos
```

Name	Size	Bytes	Class	Attributes
posicion	101x2	1616	double	
tout	101x1	808	double	

Se han creado dos variables: **posicion** y tout. La única variable que nos interesa es la primera de ellas.

Si comprobamos el tamaño de la variable con el comando **size**, veremos que se trata de un **vector** de 2 columnas y 'N' filas (este último dato puede variar según los ordenadores):

```
>> size(posicion)
ans =
101 2
```

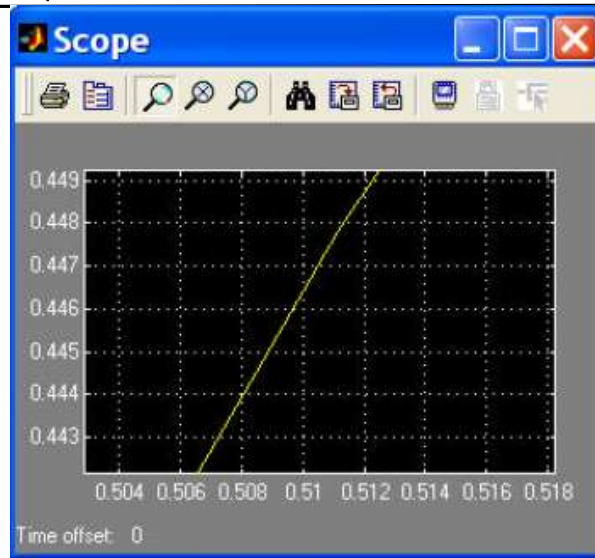
La primera de las columnas del **vector** contiene **instantes de tiempo** y la segunda contiene los valores de las magnitudes en cada instante.

Si representamos las primeras 10 filas de la variable podremos comprobar esto (nota: se usa la instrucción **'format long'** para mostrar más cifras decimales):



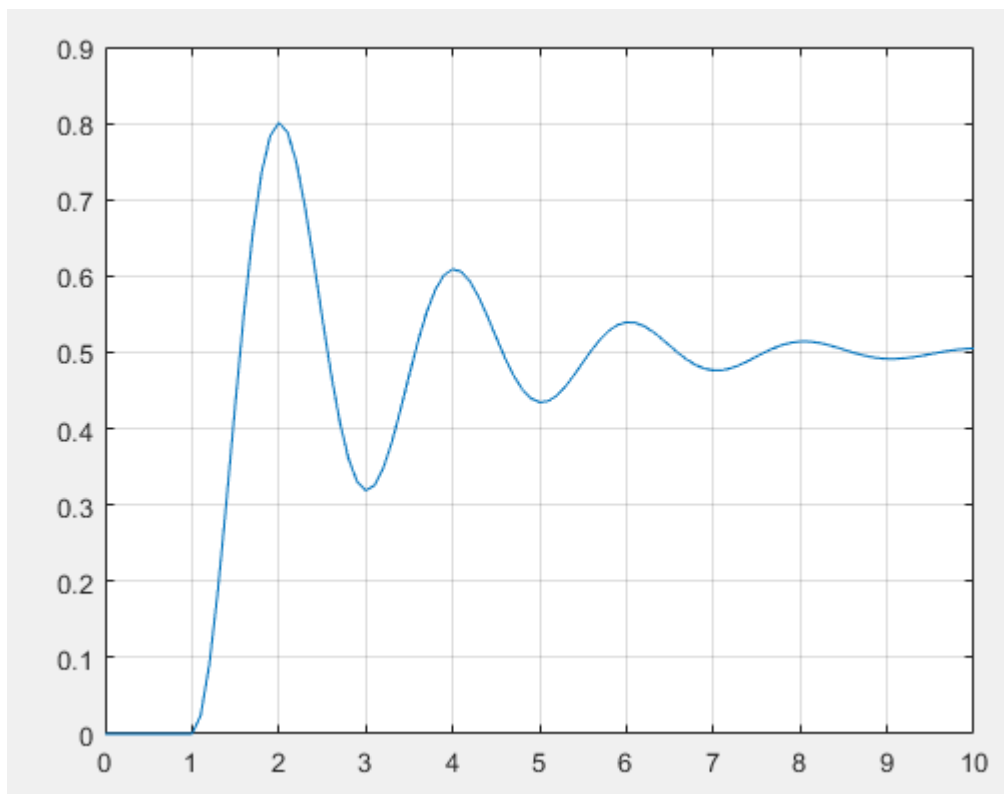
```
>> format long
>> posicion (1:10,:)
ans =
0 0
0.000000000000000 0.000000000000000
0.00004019018290 0.00000000403807
0.00024114109742 0.00000014536088
0.00124589567003 0.00000387902391
0.00626966853305 0.00009806359195
0.03138853284814 0.00243553350653
0.13244976834872 0.04137887820543
0.31114891413343 0.20203969791657
0.51114891413343 0.44779711805874
```

La primera de las columnas, como se ha dicho, corresponde a los instantes de tiempo; y podemos comprobar cómo no están separados uniformemente. La segunda columna corresponde a los valores que toma la señal, y podemos comprobar que los datos son correctos haciendo zoom sobre el gráfico que aparece en el elemento Scope (por ejemplo, en el instante 0.511 segundos el valor de la señal es 0.448 aproximadamente):



Desde la ventana de comandos de **Matlab**, también será posible representar mediante la instrucción **plot** la señal guardada en la variable **posicion**. Dado que la primera columna contiene valores de tiempo y la segunda columna contiene valores de posiciones, la forma correcta de la instrucción **plot** debería ser:

```
>> plot(posicion(:,1), posicion(:,2))
```





## Programación en Matlab/Simulink

Supongamos que nos interesa calcular el valor máximo de la señal. Dado que los valores de la señal están contenidos en la segunda columna de la variable **posicion**, bastará con utilizar el comando **max** de Matlab (se recomienda consultar la ayuda de Matlab acerca de este comando):

```
>> [maximo, indice] = max(posicion(:,2))

maximo =
    0.8010
indice =
    21
```

Los valores pueden ser ligeramente distintos en diferentes ordenadores debido a las precisiones de los cálculos. En cualquier caso, el comando **max** recorre la segunda columna de la variable **posicion** y nos devuelve su valor máximo y cuál es la fila en la que se produce ese valor (lo que hemos denominado índice). Si queremos comprobar a qué instante de tiempo corresponde ese máximo, bastará con comprobar el valor que toma la primera columna de la variable **posicion** para ese mismo índice (nota: se utiliza la instrucción **format short** para mostrar menos cifras decimales).

```
>> format short
>> posicion(indice,:)
ans =
0.9111 0.7883
```

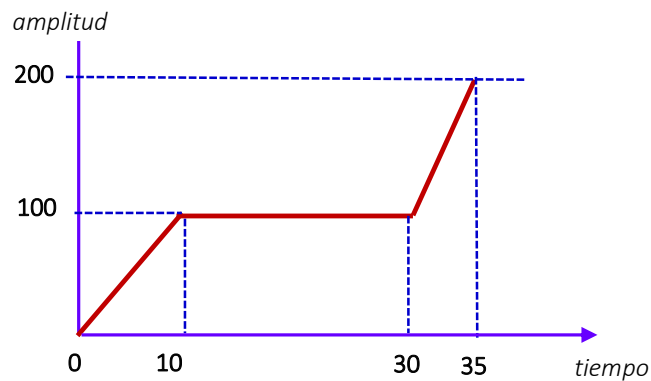
En resumen, hemos obtenido como resultado que la posición de la masa móvil (x) en el ejemplo del sistema muelle-amortiguador alcanza un valor máximo de 0.7883 metros en el instante 0.9111 segundos.

## introducción de **señales arbitrarias** en SIMULINK

En la práctica anterior se comprobó como Simulink dispone de muchas señales de entrada, válidas para la mayor parte de las aplicaciones: se dispone de las señales:

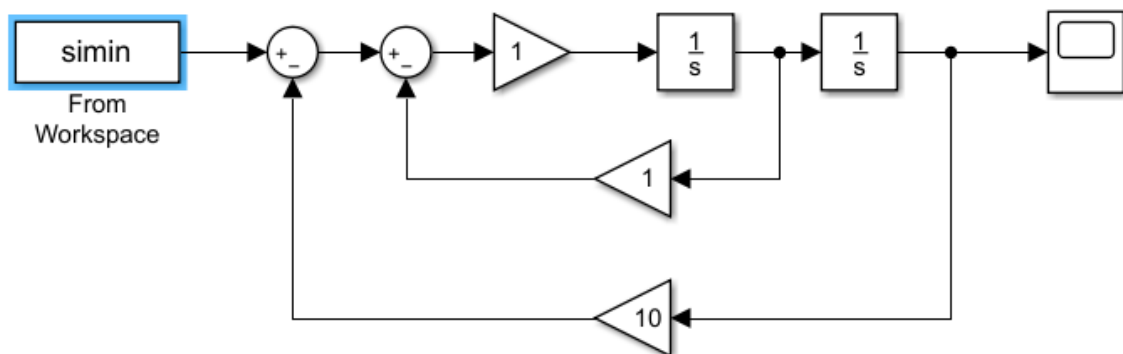
- ✓ escalón,
- ✓ rampa, señal senoidal,
- ✓ tren de pulsos,
- ✓ etc.

No obstante, en algunas ocasiones se desea utilizar señales de **forma arbitraria**, no disponibles directamente en Simulink. Por ejemplo, podríamos desear utilizar una señal de entrada (**fuerza aplicada al conjunto muelle-amortiguador en el sistema** sobre el que se está trabajando en esta práctica) como la que se muestra a continuación:

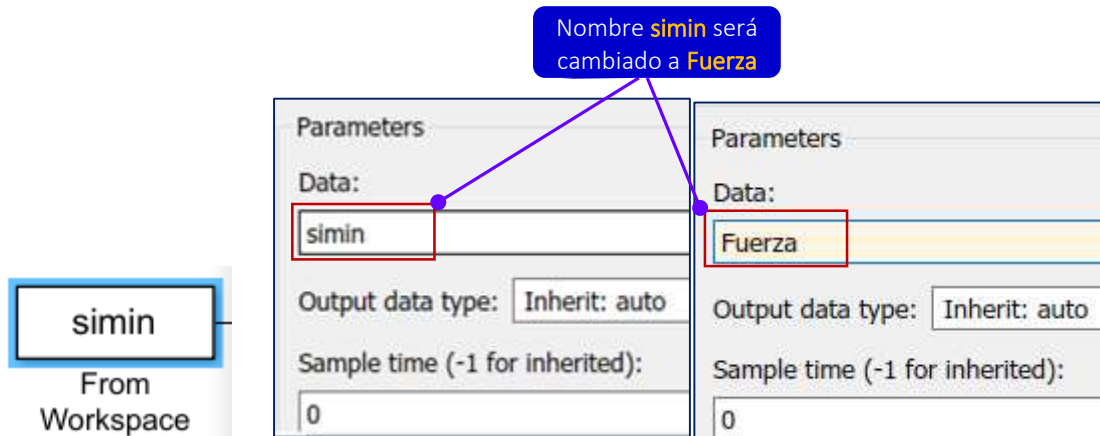


La forma más sencilla de introducir una señal como la propuesta es mediante el bloque **From Workspace** de la categoría **Sources**.

A continuación, se muestra el esquema **Simulink** una vez sustituido el bloque escalón por el bloque **From workspace**:

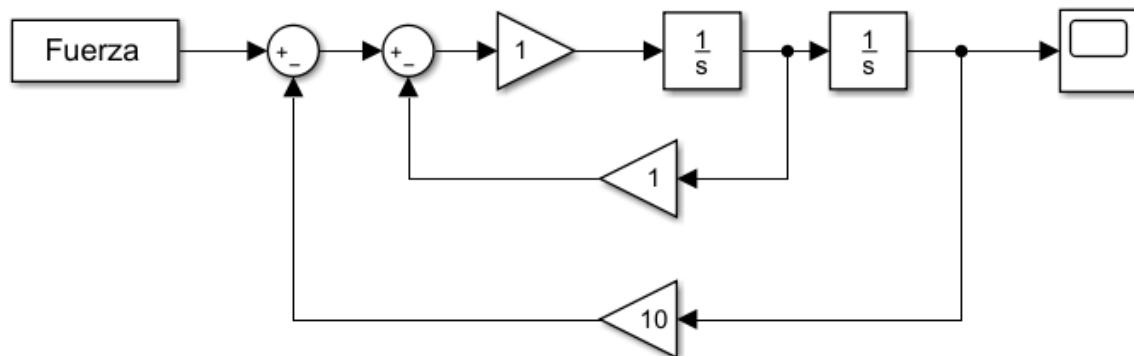


Una vez modificado el esquema **Simulink**, haremos clic sobre el elemento '**From Workspace**' para acceder a sus parámetros de configuración. Aparecerá una ventana como la que se muestra a continuación:



El parámetro nombre de la variable de **Matlab** de la que **Simulink** tomará los datos para utilizarlos como entrada, por defecto es **simin**.

Cambiaremos este nombre a **Fuerza**, que es la fuerza aplicada sobre el conjunto muelle-amortiguador.



El formato que debe tener esta variable es similar al de las variables guardadas en **Matlab** a través del bloque **Scope**: la primera columna debe contener los valores de los instantes de tiempo y la segunda los valores de la señal.

### Creación de la señal de entrada al sistema

A continuación, pasaremos a definir la señal de la figura inicial. Lo más cómodo será definirla por tramos, utilizando un tramo para cada trazo recto de la señal. Además, y para simplificar la creación de la señal, se usarán variables distintas para el tiempo y para la señal. Los tiempos se guardarán en la variable **tmp** y los valores de la señal (o datos) en la variable **dat**.

Utilizaremos un intervalo de **0.1** segundos.

Primer tramo:

## Programación en Matlab/Simulink

entre 0 y 10 segundos. La variable tiempo (vector **tmp1**) deberá tomar valores entre 0 y 10 a intervalos de 0.1; y la señal (vector **dat1**) deberá responder a la ecuación de la recta  **$x=10t$** .

Esto lo conseguimos con las siguientes sentencias de Matlab:

```
>> tmp1 = [0:0.1:10]
>> dat1 = 10*tmp1
```

### Segundo tramo:

Entre los instantes 10.1 segundos y 30 segundos. la variable tiempo (vector **tmp2**) deberá tomar valores entre 10.1 y 30 a intervalos de 0.1; y la señal (vector **dat2**) deberá responder a la ecuación de la recta  **$x=100$** . Esto lo conseguimos con las siguientes sentencias de Matlab:

```
>> tmp2 = [10.1:0.1:30]
>> dat2(1:200)=100
```

La segunda instrucción crea un vector de 200 elementos, tantos elementos como tiene el vector **tmp2**, todos ellos con valor 100.

### Tercer tramo:

Entre 30.1 y 35 segundos. La variable tiempo (vector **tmp3**) deberá tomar valores entre 30.1 y 35 a intervalos de 0.1; y la señal (vector **dat3**) deberá responder a la ecuación de la recta  **$x=20t-500$** . Esto lo conseguimos con las siguientes sentencias de Matlab:

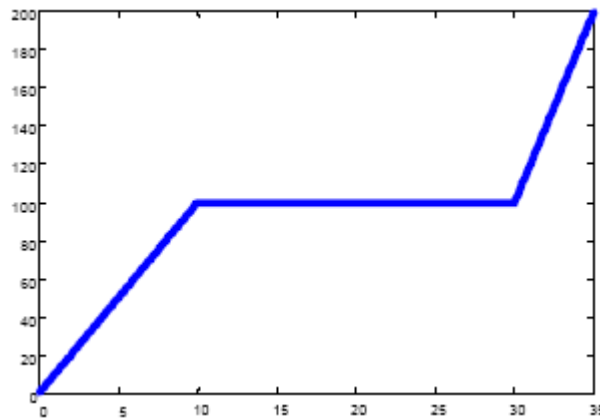
```
>> tmp3 = [30.1:0.1:35]
>> dat3 = 20*tmp3-500
```

El último paso es crear las variables **tmp** y **dat** como concatenación de las variables **tmp1**, **tmp2**, **tmp3** y **dat1**, **dat2**, **dat3** disponibles. Dado que tanto **tmp** como **dat** deben ser vectores columna, se deberán combinar los vectores y trasponer el resultado, tal y como indican las siguientes sentencias de Matlab (atención a la comilla final que indica la operación de trasposición):

```
>> tmp = [tmp1, tmp2, tmp3]';
>> dat = [dat1, dat2, dat3]';
```

## Programación en Matlab/Simulink

Para comprobar que la señal se ha generado correctamente, lo más sencillo es utilizar el comando **plot** de Matlab: la instrucción **plot(tmp,dat)** debería producir el siguiente resultado:



Una vez comprobado que las variables **tmp** y **dat** contienen valores correctos, generaremos a partir de ellas la variable fuerza necesaria para el bloque **From Workspace**. La primera columna deberá contener los tiempos y la segunda los datos, con lo que las instrucciones Matlab a utilizar serán:

```
>> fuerza(:,1)=tmp  
>> fuerza(:,2)=dat
```

Con la variable fuerza creada, es posible lanzar la simulación. Dado que la señal está definida durante 35 segundos, el tiempo de simulación se fijará exactamente en 35 segundos (desde la opción **Simulation parameters** del menú **Simulation**). El resultado sobre el osciloscopio se muestra en la figura siguiente, y se debe apreciar cómo la salida del sistema (posición) reproduce aproximadamente los valores de la entrada (fuerza aplicada) con pequeñas oscilaciones.

