

## Práctica 8. Introducción a Xcos II

**Objetivo.** Realizar operaciones de intercambio de datos entre el área de trabajo de Scilab y Xcos, y creación de subsistemas.

### Introducción

En la práctica anterior se ha utilizado Xcos para la simulación de ecuaciones diferenciales mediante la construcción del diagrama de simulación basado en bloques integradores.

Para visualizar los resultados de la simulación se han usado los bloques CSCCOPE o CMSCOPE, los cuales permiten algunas funciones interactivas con el usuario para ayudar a analizar los resultados de la simulación, tales como: zoom, modificación de escalas, edición de colores, etiquetas y propiedades de los ejes y de las curvas.

### Exportando datos de Xcos a Scilab.

Cuando queremos hacer un procesamiento más detallado o una representación gráfica alternativa de los resultados de una simulación, Scilab cuenta con el bloque "**To Work Space**" (**TOWS\_c**) de la categoría "**Sinks**". Este bloque puede configurarse para que los resultados de una simulación se vallan guardando en una variable de Scilab, de modo que se puedan procesar sus valores desde el ambiente de trabajo de Scilab en un proceso *posterior* a la simulación.

### Ejemplo:

Retomaremos uno de los ejemplos de la práctica anterior, el cual era un sistema mecánico consistente en una masa **M** sometida a una fuerza externa **f(t)** y unida a un resorte de constante elástica **K**, y con un rozamiento viscoso **B**, como se describe en la figura 8.1

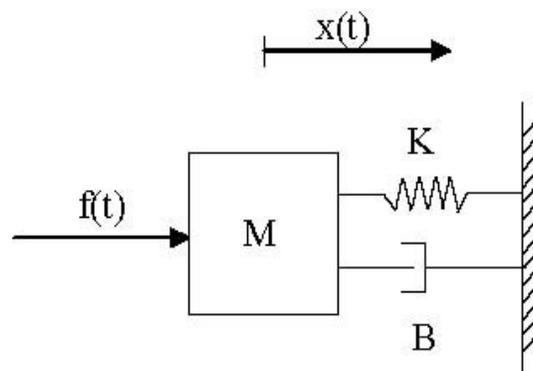


Figura 8.1.- Sistema mecánico masa-resorte-amortiguador.

La ecuación que describe la dinámica del sistema es

$$M \frac{d^2 x(t)}{dt^2} + B \frac{dx(t)}{dt} + Kx(t) = f(t) \quad (8.1)$$

Y su diagrama de simulación correspondiente en Xcos es el mostrado en la figura 8.2. Obsérvese que ahora se ha agregado un bloque TOWS\_c conectado a la salida del sistema.

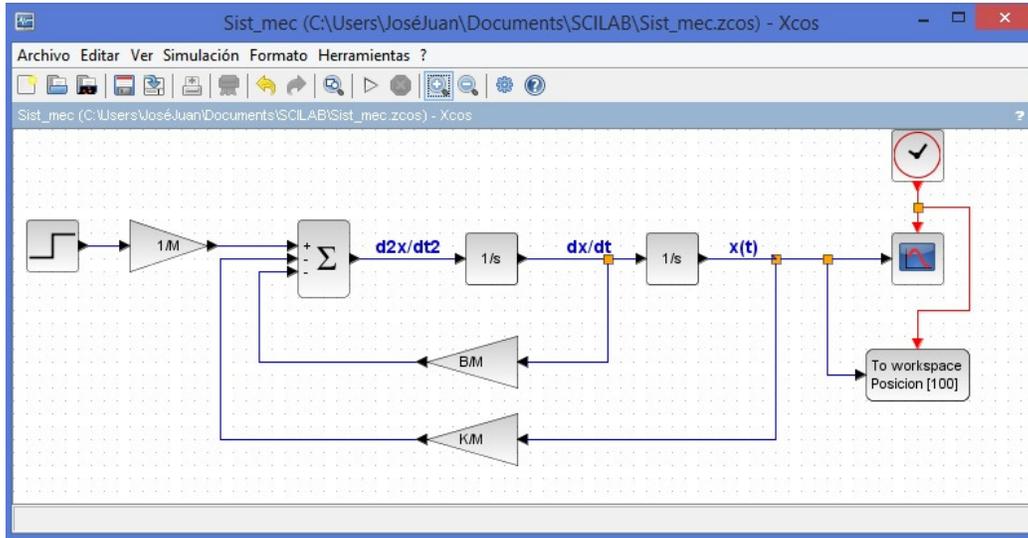


Figura 8.2.- Diagrama de simulación para el sistema masa-resorte-amortiguador.

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

- Tiempo de simulación: 10 seg.
- Amplitud del escalón de entrada: 5 unidades.
- tiempo inicial del escalón: 0 seg.
- Valores de las constantes del sistema  $M=1\text{Kg}$ .  $K=10\text{ Nt seg/m}$ ,  $B=1\text{ Nt}$ .
- Muestreo del bloque CLOCK\_c: Cada 0.1 seg. iniciando en el instante 0.

Para la configuración del bloque TOWS\_c, la cual se muestra en la figura 8.3 se debe considerar que en el intervalo de simulación de 0 a 10 seg. y con toma de muestras cada 0.1 seg se generarán 100 puntos de la solución, por lo tanto se deberá configurar el tamaño del buffer de 100 puntos. Para el nombre de la variable de Scilab en la que se guardarán los 100 puntos se elegirá 'Posicion', ya que se trata de la variable de posición de la masa del sistema simulado.

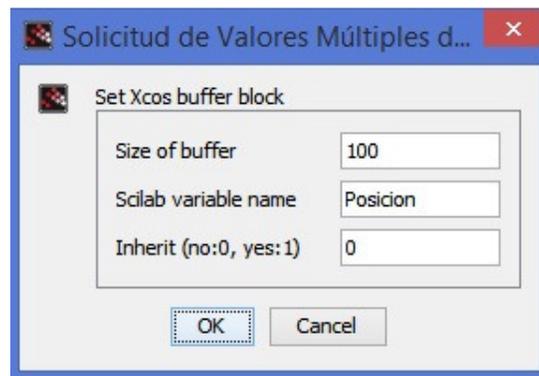


Figura 8.3.- Configuración del bloque TOWS\_c

Una vez que se ejecuta la simulación del sistema se obtiene el comportamiento de la salida  $x(t)$  del sistema, como se muestra en la figura 8.4.

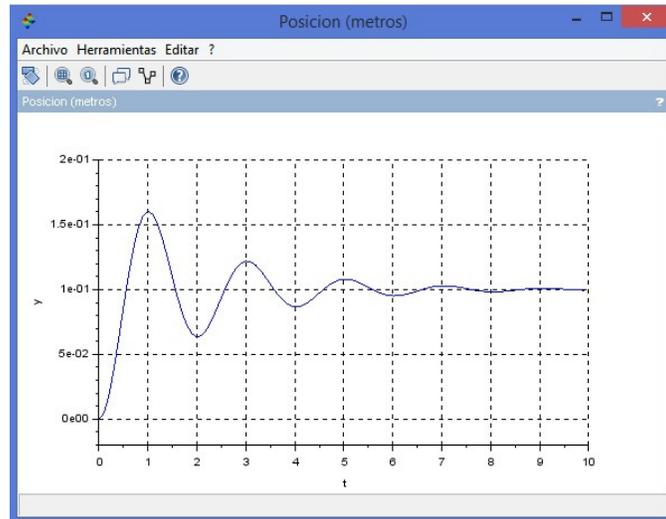


Figura 8.4.- Comportamiento de la salida  $x(t)$  del sistema.

Para observar el efecto del bloque TOWS\_c en Scilab, desde la consola de comandos se puede averiguar el contenido de la nueva variable `Posicion` tecleando:

```
Posicion //Despliega contenido de la variable Posicion
Posicion =
  values: [100x1 constant]
  time: [100x1 constant]
```

Lo que significa que la variable `Posicion` es una estructura que contiene dos variables:

- La variable `Posicion.values`, la cual es un vector de 100 x 1 que contiene los valores de la salida (variable  $x(t)$ ) del sistema simulado.
- y la variable `Posicion.time`, la cual es un vector de 100 x 1 que contiene los instantes de muestreo del tiempo de simulación.

Para comprobar esto, podemos visualizar los primeros 10 pares de valores  $t$ ,  $x(t)$  como sigue (se usa la instrucción `format long` para mostrar más cifras decimales):

```
t=Posicion.time; //Vector de instantes de muestreo
x=Posicion.values; //Vector de valores de posicion
[t(1:10) x(1:10)] //Despliega los primeros 10 valores de cada vector
ans =
  0.      0.
  0.1    0.0047970
  0.2    0.0181221
  0.3    0.0379059
  0.4    0.0616665
  0.5    0.0867820
  0.6    0.1107545
  0.7    0.1314307
```

```
0.8    0.1471750
0.9    0.1569684
```

Ya que tenemos los valores de la variable  $x(t)$  en una variable de Scilab, podemos procesarlos mediante los comandos disponibles. Por ejemplo, para visualizar el comportamiento de la variable  $x(t)$  desde la ventana de comandos de Scilab, podemos usar el comando `plot` como sigue para obtener la gráfica de la figura 8.5. Compárese con la obtenida en Xcos (figura 8.4).

```
t=Posicion.time; //Vector de instantes de muestreo
x=Posicion.values; //Vector de valores de posicion
plot(t,x,'-bo','LineWidth',2); xgrid;
a=get("current_axes");
a.font_size=4; //tamaño de las etiquetas de valores de los ejes
title('Respuesta de la posición del sistema ante un escalón
unitario','fontSize',4);
xlabel('tiempo (seg)','fontSize',3);
ylabel('Posición x(t)','fontSize',3);
```

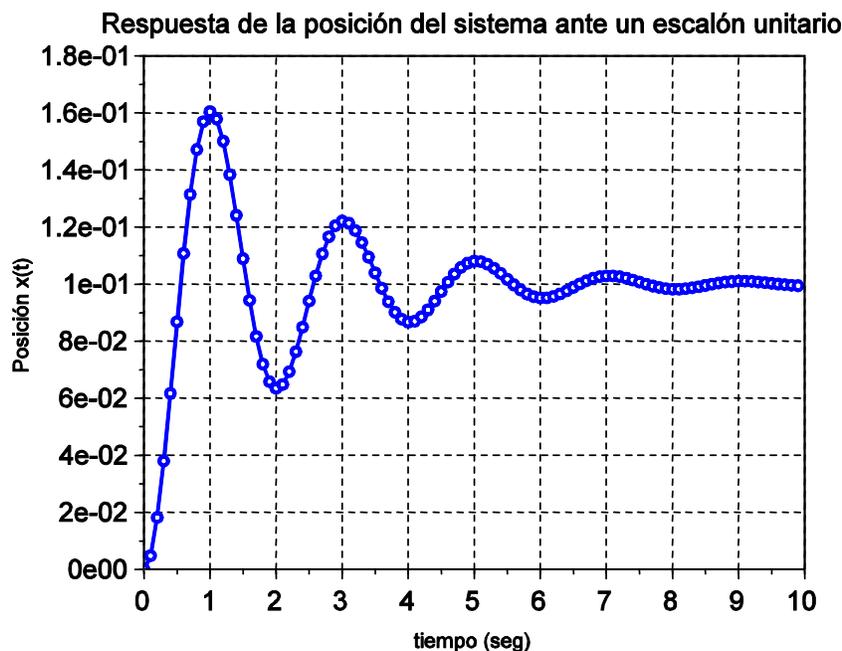


Figura 8.5.- Gráfica de los datos obtenidos en la variable `Posicion` mediante el comando `plot`.

Podemos hacer cualquier cálculo deseado sobre los valores de la variable  $x(t)$ . Por ejemplo, para calcular el valor máximo de la señal, bastará con utilizar el comando `max` de Scilab:

```
[maximo, indice] = max(x) //Busca el valor máximo y su índice en x(t)
indice =
    11.
maximo =
    0.1604538
```

Los valores pueden ser ligeramente distintos en diferentes ordenadores debido a las tolerancias y los *solvers* elegidos en los cálculos. Para verificar 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.

```
[t(indice), x(indice)]
ans =
    1.    0.1604538
```

Es decir, la posición  $x(t)$  de la masa del sistema mecánico masa-resorte-amortiguador alcanza un valor máximo de 0.1604538 metros en el instante  $1.0 \pm 0.05$  seg. (Recuérdese que la variable  $x(t)$  está muestreada cada 0.1 seg.)

### Modificación del número de puntos calculados.

Si observamos el resultado de la simulación del ejemplo anterior, podemos apreciar que es un poco burda, ya que aparece segmentada en líneas rectas (se aprecia mejor en la figura 8.5), esto se puede mejorar de dos maneras:

- 1) El paso de integración del método (solver) utilizado por Scilab tiene una precisión controlada por los parámetros de tolerancia absoluta, relativa y de tiempo y por el paso máximo de integración utilizado. Si deseamos un resultado más preciso de la simulación debemos modificar estos parámetros accediendo a la ventana de configuración (ver figura 8.6), sin embargo, *en la mayoría de los casos los parámetros por default son los más adecuados.*
- 2) El periodo de muestreo seleccionado en el bloque CLOCK\_c también se puede disminuir para producir una mayor densidad de puntos de la gráfica.

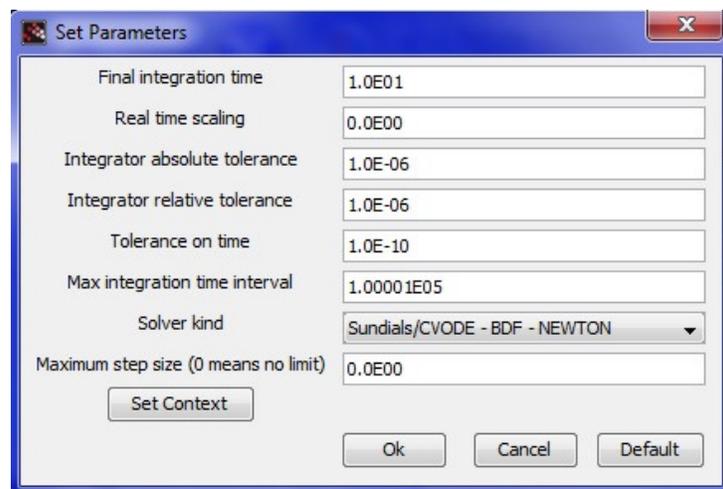


Figura 8.6.- Parámetros (default) usados para la simulación del sistema mecánico.

En la figura 8.7 (compárese con la figura 8.5) se muestra el resultado de simular el ejemplo del sistema mecánico masa-resorte-amortiguador modificando el periodo de muestreo del bloque CLOCK\_c con diferentes valores: 0.1, 0.01 y 0.001. En todos los casos se usa un buffer del bloque TOWS\_c de 10000 puntos.

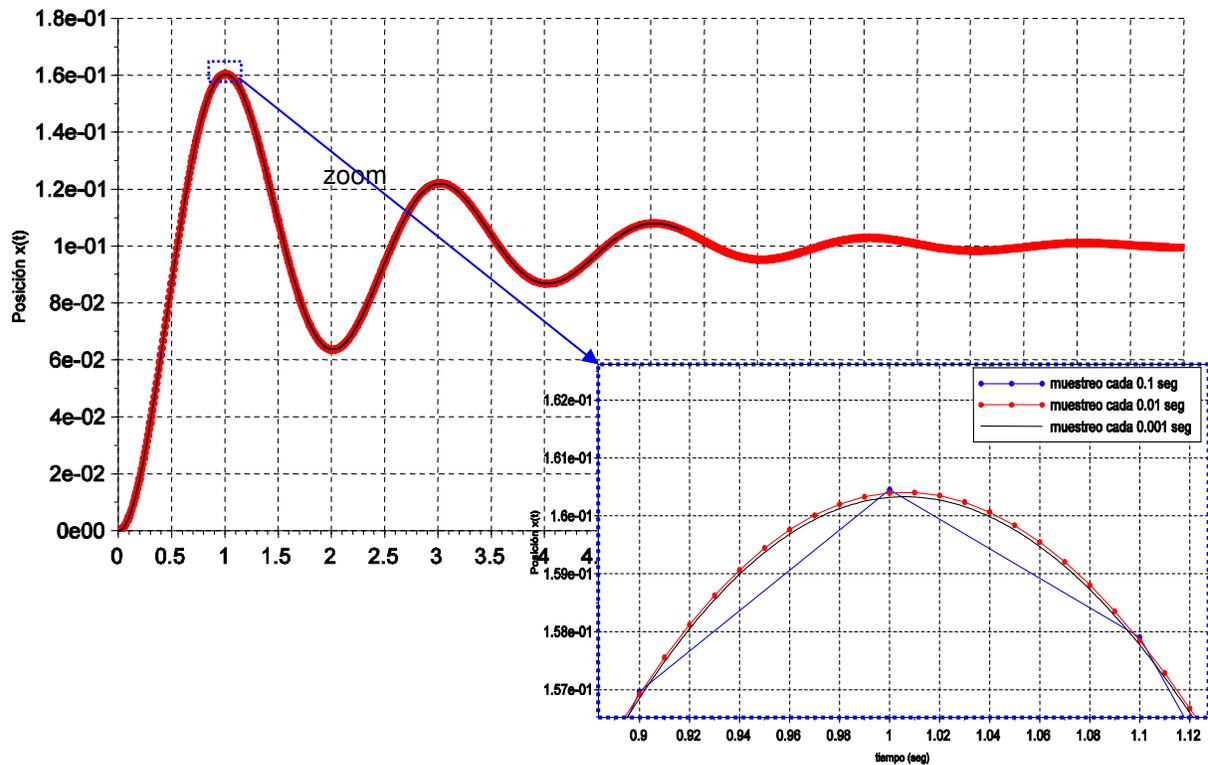


Figura 8.7.- Modificación del número de puntos calculados.

En la figura 8.7 se puede apreciar que al disminuir el periodo de muestreo se obtiene un mayor número de puntos de la respuesta del sistema e inclusive se modifica ligeramente la exactitud de los puntos calculados.

### Importación de variables de Scilab a Xcos.

En los ejemplos de simulación anteriores se han utilizado valores de constantes que aparecen en el diagrama de simulación de Xcos y cuyo valor es asignado mediante instrucciones de Scilab en la ventana "**Asignar Contexto**". Sin embargo, si en lugar de constantes queremos usar variables (señales que varían respecto al tiempo), el procedimiento es muy diferente.

Xcos dispone de muchas señales de entrada, válidas para la mayoría de las aplicaciones: señales escalón, rampa, sinusoidal, tren de pulsos, señales aleatorias, etc. Sin embargo, a veces se desea utilizar señales de forma arbitraria, que aunque pueden ser introducidas mediante el bloque "**Signal Builder**" o el bloque **CURV\_f** de Xcos, a veces resulta más cómodo utilizar una variable definida en Scilab e importarla desde Xcos.

Por ejemplo, si quisiéramos utilizar una señal de entrada (fuerza aplicada al conjunto muelle-amortiguador en el ejemplo que se está trabajando en esta práctica) como la que se muestra en la figura 8.8

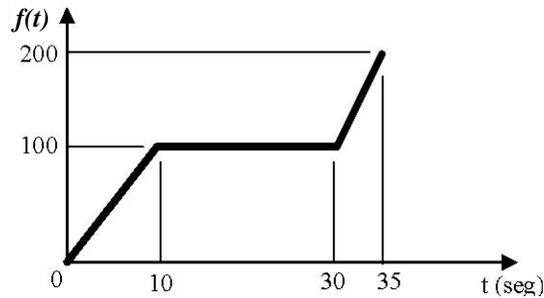


Figura 8.8.- Señal de entrada de forma especial.

Una manera de introducir una señal como la de la figura 8.9 en Xcos es generarla en una variable de Scilab y luego mediante el bloque FROMWSB (**From Workspace**) de la categoría "**Fuentes**" introducirla al diagrama de simulación de Xcos. En la figura 8.9 se muestra el aspecto que debe tener el diagrama de simulación de la práctica una vez sustituido el bloque escalón por el bloque 'From workspace'

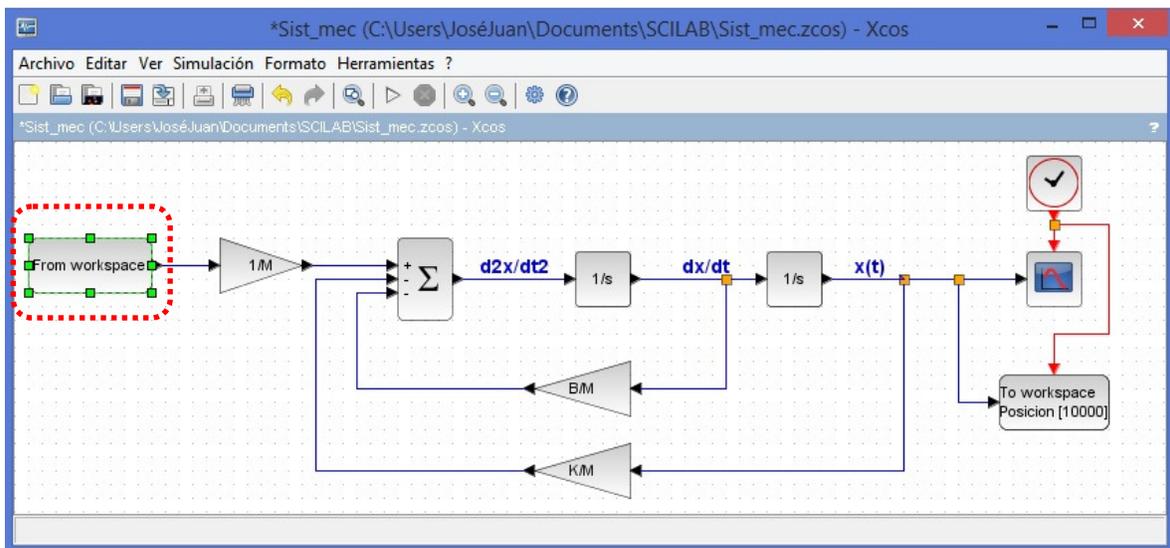


Figura 8.9.- Sistema Masa-resorte-amortiguador con entrada obtenida de un bloque FROMWSB

Una vez modificado el esquema Xcos, haremos click sobre el elemento 'FROMWSB' para acceder a sus parámetros de configuración. Aparecerá una ventana como la que se muestra en la figura 8.10.

El único parámetro que nos interesa por el momento es el nombre de la variable de Scilab de la que Xcos tomará los datos para utilizarlos como entrada. Por defecto, esta variable se denomina 'v'. Cambiaremos este nombre por otro más ilustrativo: Dado que la entrada al sistema considerado es la fuerza aplicada a la masa del sistema, llamaremos a esta variable '**fuerza**'. Esta variable debe ser una estructura con dos campos:

- El campo '**time**' debe ser un vector de dimensión  $n \times 1$ , conteniendo los valores de los instantes de tiempo.
- El campo '**values**' es otro vector de dimensión  $n \times 1$ , el cual debe contener los valores de la señal de fuerza en los instantes dados por **time**.

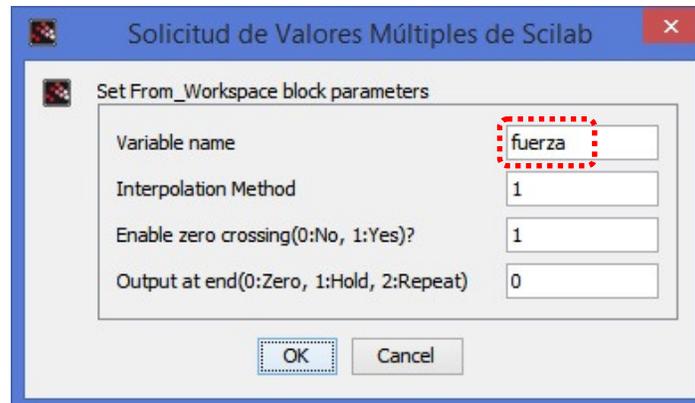


Figura 8.10.- Ventana de configuración del bloque 'From Workspace'

De acuerdo a la figura 8.8, la señal de fuerza se puede definir de acuerdo a las ecuaciones de los tres segmentos de recta que la conforman en cada uno de los tres sub intervalos como sigue

$$f(t) = \begin{cases} 10t & 0 \leq t \leq 10 \\ 100 & 10 < t \leq 30 \\ 20t - 500 & 30 < t \leq 35 \end{cases} \quad (8.2)$$

Por lo tanto, en Scilab podemos definir  $f(t)$  aprovechando que las desigualdades tienen valores lógicos equivalentes a 0 ó 1, como sigue

```
t=0:0.1:35;
f=10*t.*(0<=t & t<=10) + 100*(10<t & t<=30) + (20*t-500).*(30<t & t<=35);
plot(t,f); xgrid;
```

Que produce el resultado mostrado en la figura 8.11

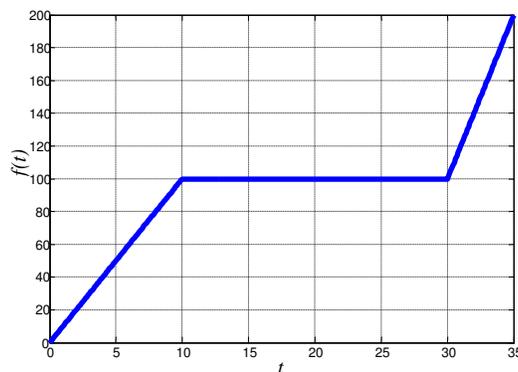


Figura 8.11.- Señal de fuerza generada mediante la expresión (8.2).

Una vez que se ha comprobado visualmente en la figura 8.12 que las variables **t** y **f** contienen valores correctos, generaremos a partir de ellas la variable fuerza necesaria para el bloque '**From Workspace**' como sigue

```
fuerza.time=t';
fuerza.values=f';
```

Ahora es posible lanzar la simulación. El tiempo de simulación se deberá fijar exactamente en 35 segundos (en la opción **Configuración** del menú **Simulación**) y el bloque SCOPE\_c se deberá reconfigurar para una escala vertical de 0 a 20 y un periodo de refrescamiento de 35 seg. El resultado de la simulación se muestra en la figura 8.12. Obsérvese cómo la salida del sistema (posición) da un seguimiento aproximado a los valores de la entrada (fuerza aplicada) con pequeñas oscilaciones.

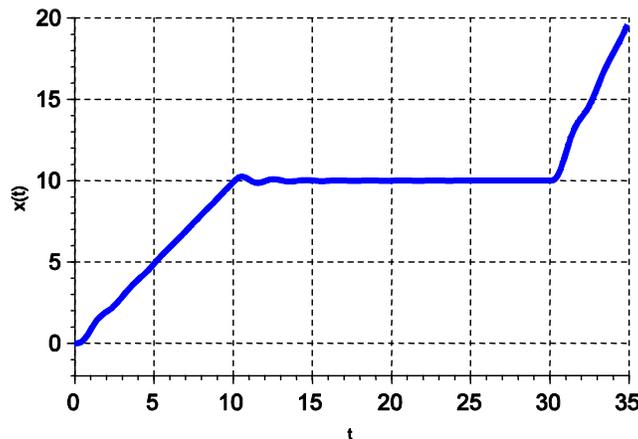


Figura 8.12.- Resultado de la simulación del diagrama de la figura 8.9 con la variable fuerza definida en Scilab.

### Generación de funciones del tiempo en Xcos.

Una alternativa para introducir una señal especial como la del ejemplo anterior sin recurrir a una variable importada desde Scilab, es generarla directamente en Xcos.

Para generar una señal (función del tiempo) desde Xcos, primeramente se debe generar la 'señal de tiempo', es decir, una señal cuyos valores en cada instante coincidan con el tiempo simulado, es decir,  $f(t) = t$  lo cual corresponde a una señal rampa unitaria (pendiente uno, inicio en cero) como se muestra en la figura 8.13



Figura 8.13.- Generación de la señal tiempo.

Una vez que se ha generado la señal de tiempo, se puede producir cualquier función del tiempo usando bloques de Xcos para hacer la operación deseada, o de manera más sencilla mediante un bloque de **función definida por el usuario 'Scifunc\_block\_m'**, por ejemplo, para generar la misma señal del ejemplo anterior se puede hacer con el diagrama de la figura 8.14. Obsérvese que dentro del bloque **'Scifunc\_block\_m'** se escribe la misma expresión que se usó en el código de Scilab para obtener el vector  $y1$ , salvo que ahora se reemplaza  $t$  por  $u(1)$ .

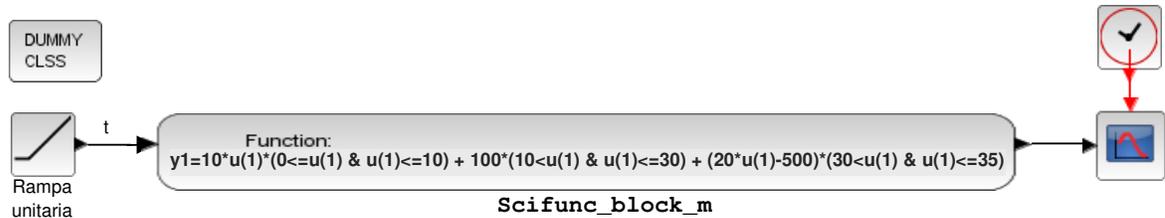


Figura 8.14.- Generación de la señal del ejemplo anterior mediante una funcion definida por el usuario en Xcos.

### Creación de Subsistemas (Superbloques).

Cuando un diagrama de simulación es muy grande se vuelve impráctico conservar todos los bloques a la vista del usuario. En este caso conviene definir conjuntos de bloques como parte de un subsistema y esconderlos a la vista del usuario, mostrando solamente la entrada y la salida de dicho subsistema.

Para Crear un subsistema en Xcos se seleccionan con el cursor todos los bloques que se encerrarán en un bloque común y se accede a la opción '**Selection to Superblock**' del menú '**Editar**'.

Por ejemplo, en el diagrama de la figura 8.2 seleccionamos con el cursor todos los elementos, dese la salida del bloque STEP hasta la entrada del bloque SCOPE y convertimos a un **superbloque**, entonces el diagrama se convierte en el mostrado en la figura 8.15. El cual aún contiene la información de los bloques originales. Si deseamos ver los bloques interiores del subsistema creado le damos doble click al bloque del subsistema y se muestra el diagrama de la figura 8.16.

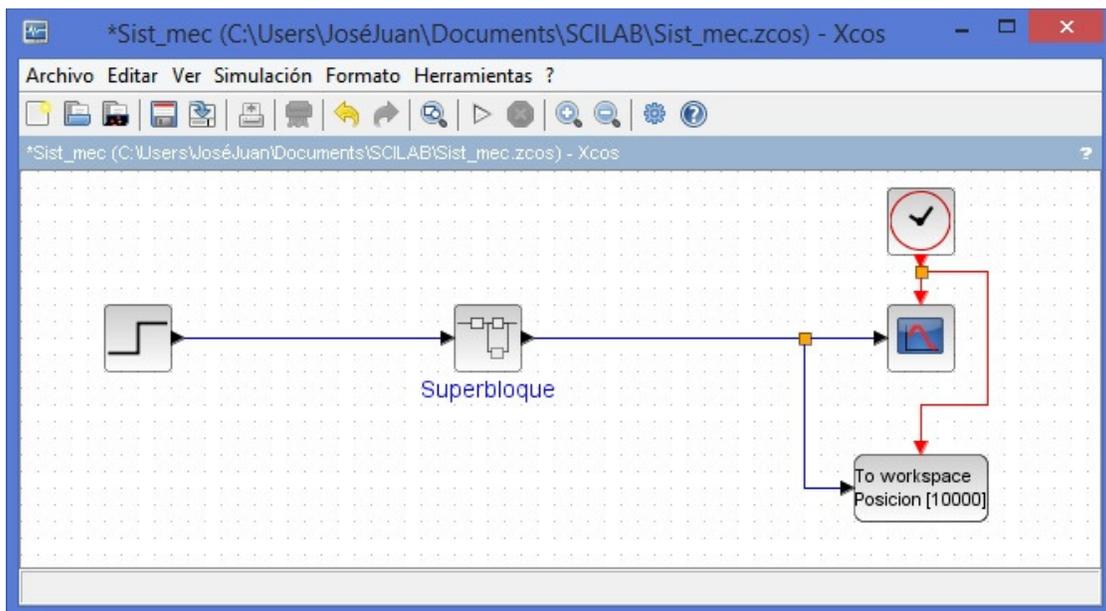


Figura 8.15 Creación de subsistema a partir de bloques de la figura 8.2.

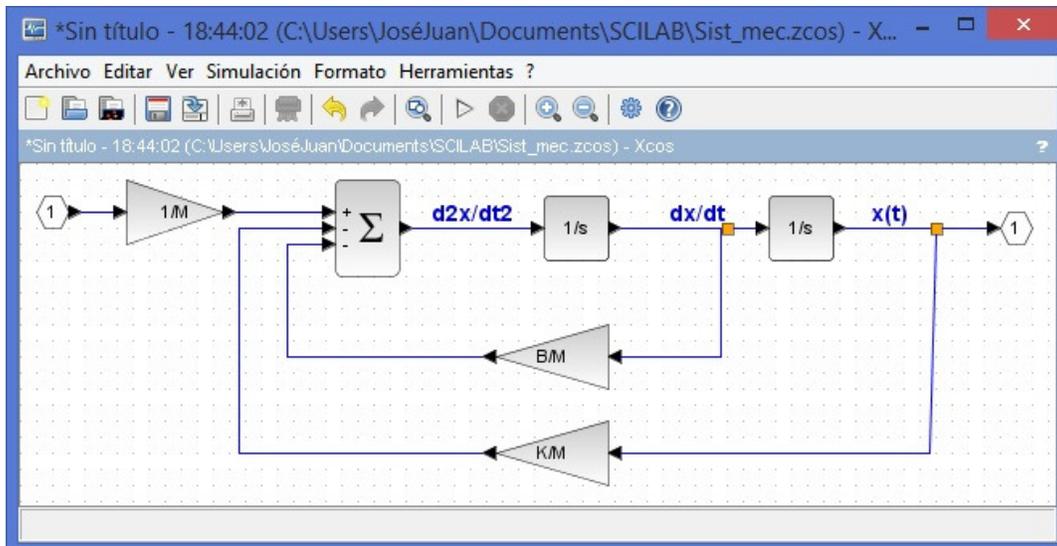


Figura 8.16.- Subsistema visualizado al dar doble click al *superbloque* de la figura 8.15.

### Sistemas Lineales en Xcos.

Aunque cualquier sistema, ya sea Lineal o No-lineal puede simularse obteniendo su diagrama de simulación con integradores a partir de sus ecuaciones de diferenciales, Xcos proporciona bloques para el caso de sistemas lineales que nos ahorran la obtención del diagrama de simulación mediante integradores, estos bloques se muestran en la figura 8.17.

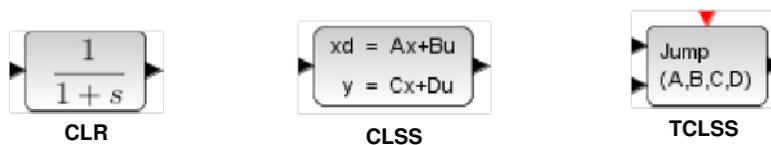


Figura 8.17.- Bloques para introducir Sistemas Lineales en Xcos.

- El bloque '**CLR**' (Continuous Linear Transfer Function) permite introducir un sistema lineal mediante su Función de Transferencia especificando los polinomios del numerador y denominador en potencias de 's'.
- El bloque '**CLSS**' (Continuous Linear Space State) permite introducir un sistema lineal especificando las matrices A, B, C y D de su modelo en variables de estado además del vector de condiciones iniciales para el vector de estado.
- El bloque '**TCLSS**' (Continuous Linear Space State) Es igual que el bloque CLSS, pero permite reiniciar cuando ocurre un evento los valores del vector de estados en base a una entrada adicional y a una entrada de eventos.

### Ejercicio.

1) Reemplazar el diagrama de simulación del sistema masa-resorte-amortiguador por un solo bloque de función de transferencia. Recordar que la función de transferencia de ese sistema es

$$\frac{X(s)}{F(s)} = \frac{1}{Ms^2 + Bs + K} \quad (8.3)$$

2) Considerando  $M=1$ ,  $B=1$ ,  $K=10$ , Obtener la respuesta del sistema ante un escalón de 5 Newtons de amplitud y comparar la respuesta con la figura 8.4

3) Obtener la respuesta del sistema ante la entrada mostrada en la figura 8.18

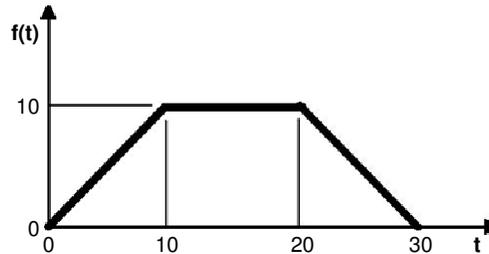


Figura 8.18.- Entrada de prueba del ejercicio.

4) Exportar la respuesta del inciso anterior a una variable de Scilab denominada 'xt' y Graficarla contra el tiempo mediante el comando `plot` junto con la gráfica de la entrada.

#### Desarrollo de la Práctica.

1. Probar todos los ejemplos propuestos por el profesor conforme los va explicando.
2. Realizar todos los ejercicios propuestos.

#### Reportar:

1) El diagrama de simulación elaborado para el ejercicio final de la práctica usando el bloque de función de transferencia y las gráficas obtenidas en cada inciso.