

Práctica 7. Introducción a Scicos

Objetivo. Realizar operaciones y simulaciones sencillas mediante la herramienta gráfica de simulación de Scilab llamada Scicos o Xcos.

Introducción.

Como ya se dijo antes, Scilab es un software que se ha inspirado en Matlab desde su origen. Matlab posee un entorno gráfico basado en diagramas de bloques para realizar simulación de sistemas dinámicos en el dominio del tiempo llamado Simulink. Por su lado Scilab ha desarrollado un entorno gráfico muy similar a Simulink, denominado Scicos o Xcos.

El uso de Xcos está orientado a que el usuario defina sistemas a simular mediante diagramas de bloques de manera que el usuario pueda interactuar con los bloques, los cuales representan subsistemas. Los bloques se pueden situar donde se requiera, modificar su tamaño, etiquetarlos, especificar sus parámetros, e interconectarlos para completar sistemas a partir de los cuales se pueden ejecutar simulaciones.

Las características básicas de Xcos son: Simulación de sistemas dinámicos, entorno gráfico, el sistema se representa como una interconexión de bloques elementales, cada bloque lleva asociado un modelo matemático que representa su relación entrada/salida.

Iniciando Xcos

Para iniciar **Xcos**, se puede proceder de diferentes formas, la más sencilla es teclear desde la ventana de comandos de Scilab:

```
Xcos
```

Otra manera de posible acceder al programa es mediante el botón de **Xcos** de la barra de botones de Scilab:



Figura 7.1.- Botón Xcos de la barra de herramientas de Scilab

Una vez que inicia Xcos, aparecen dos ventanas nuevas:

- El "**Explorador de Paletas**" o "**Palette Browser**", el cual es una lista de bloques básicos estándar de Xcos agrupados en categorías tales como sistemas de tiempo continuo, discontinuidades, sistemas de tiempo discreto, operaciones matemáticas, etc.)
- El "**editor**" de diagramas de simulación, el cual permite la introducción bloque por bloque tomados del explorador de paletas para formar un diagrama de simulación.

- El **compilador/simulador**, (integrado en la ventana del editor) permite convertir un diagrama de bloques en el código ejecutable para realizar la simulación y visualizar resultados numéricos o gráficos.

En la figura 7.2 se muestra el aspecto del **Palette browser** de Xcos. En la parte izquierda de esta ventana se muestran todas las categorías de bloques disponibles; cada una de ellas contiene bloques elementales con los que se pueden construir sistemas. En la parte derecha de la ventana se muestra una descripción gráfica de la lista de bloques de la categoría seleccionada..

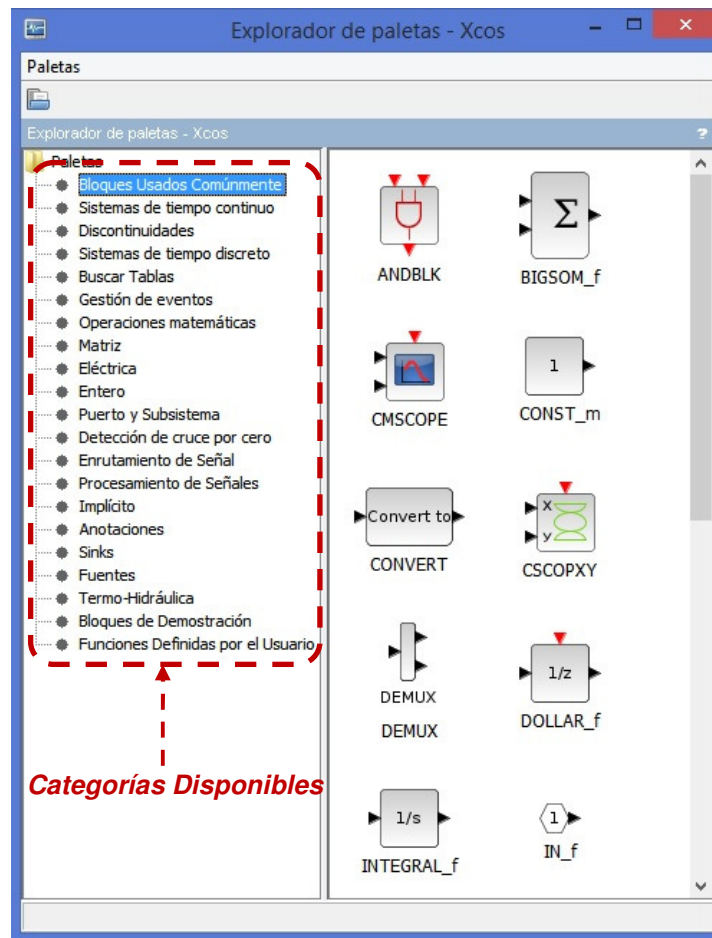


Figura 7.2 Explorador de paletas de Xcos.

Tipos de bloques en Xcos

Como puede observarse, existe multitud de bloques distintos; todos los bloques de Xcos se estructuran en diferentes categorías. La ventana que aparece al iniciar Xcos permite comprobar cuales son las categorías de los bloques existentes. Basta con hacer clic sobre la categoría deseada, para visualizar los bloques disponibles en cada una de las categorías disponibles.

Si se desea información sobre alguno de los bloques, se puede dar click-derecho sobre el bloque y aparece un menú de contexto, con la opción de incluirlo en el diagrama que se está construyendo, o con la opción "**Block help**", la cual abrirá el navegador de ayuda mostrando la descripción detallada del bloque seleccionado Ver figura 7.3.

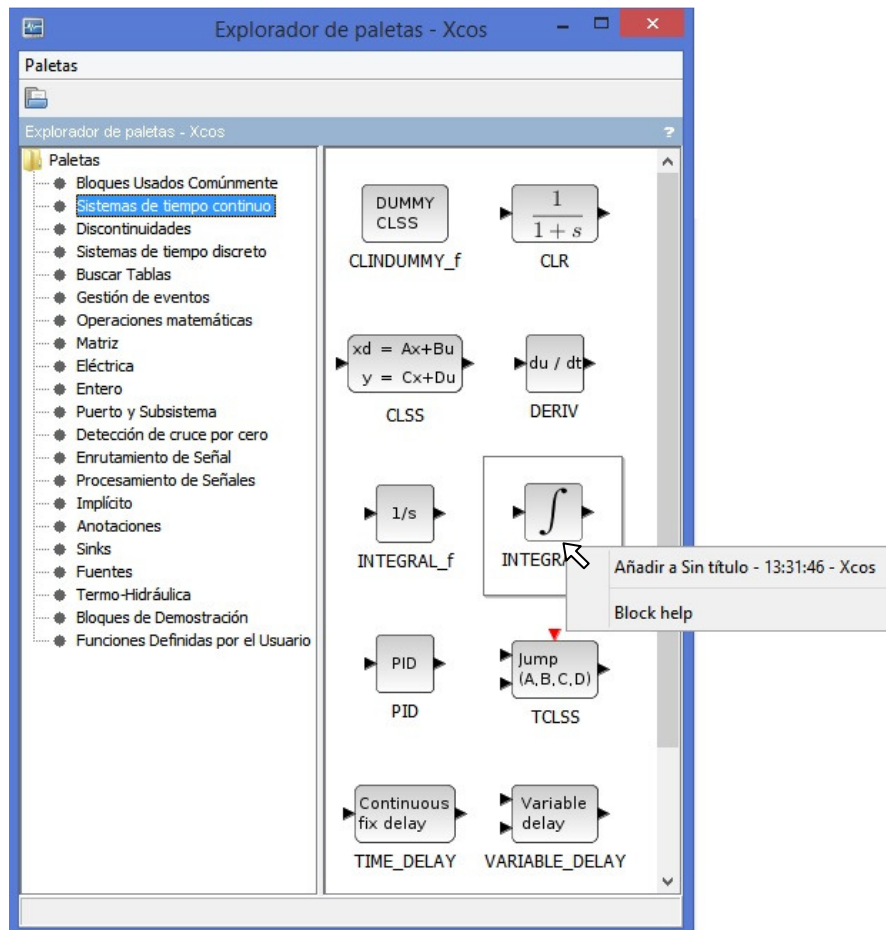


Figura 7.3.- Clic sobre la categoría de bloques "Sistemas de tiempo continuo".

A continuación se indican *algunos* de los bloques básicos que se usarán en esta práctica y las siguientes y la categoría a que pertenecen.

Sistemas de tiempo continuo: Bloques para introducir sistemas continuos lineales y retardos de transporte

- **DERIV** (bloque derivador: la salida es la derivada de la entrada)
- **INTEGRAL_f** (bloque integrador: la salida es la integral de la entrada)
- **CLR** (Sistema lineal continuo especificado por su Función de transferencia expresada como cociente de polinomios en la variable de Laplace s)
- **CLSS** (Sistema lineal continuo, especificado por las matrices A,B,C y D de su modelo en espacio de estado.)

Fuentes: Entradas o fuentes de señales

- **STEP_FUNCTION** (señal escalón)
- **RAMP** (señal rampa)
- **GENSIN_f** (señal sinusoidal)
- **PULSE_SC** (Señal tren de pulsos periodica)
- **RAND_m** (Señal aleatoria con distribución Uniforme o Gaussiana)

- **CONST** (Señal de Valor constante)
- **FROMWSB** (Señal importada desde el espacio de trabajo (consola de comandos) de Scilab)
- **CLOCK_c** (Generador periódico de eventos de reloj)

Sinks: Salidas o dispositivos de visualización/almacenamiento de variables del sistema

- **CSCOPE** (osciloscopio de un solo canal)
- **CMSCOPE** (osciloscopio de múltiples canales)
- **CSCOPXY** (Graficador de figuras de Lissajous Y vs. x)
- **AFFICH_m** (Display numérico)
- **TOWS_c** (Envía señal a una variable del espacio de trabajo de Scilab)

Operaciones Matemáticas: Operaciones matemáticas sobre señales

- **SUMMATION** (sumador/restador de señales)
- **SUM_f** (sumador de tres señales)
- **GAIN_f, GAINBLK, GAINBLK_f** (ganancia o multiplicación de una señal por una constante)

Enrutamiento de Señales: Permite realizar conexiones especiales entre señales

- **MUX** (multiplexor: Agrupa distintas señales en un vector o bus)
- **DEMUX** (demultiplexor: Desagrupa señales provenientes de un bus)

Funciones Definidas por el Usuario: Permiten al usuario introducir expresiones algebraicas que procesan la entrada del bloque para calcular la salida.

- **EXPRESSION** (Expresión algebraica genérica: Calcula la salida mediante la expresión definida por el usuario en función de la entrada o entradas al bloque. La expresión puede ser cualquier expresión válida en Scilab)
- **scifunc_block_m** (Función de Scilab o de Scicos: Calcula la salida mediante una expresión algebraica que contiene funciones estándar de Scitlab definida por el usuario y aplicada a la señal de entrada.)

Creación de un modelo sencillo

El procedimiento para la creación de un modelo consiste en seleccionar los bloques adecuados de entre los presentes en cada categoría, colocarlos sobre la ventana de edición y establecer las conexiones correspondientes entre ellos mediante el cursor del mouse. A continuación se ilustra el procedimiento mediante un sistema sencillo.

Ejemplo.

Crearemos un modelo que nos permita generar y visualizar 5 ciclos de una señal sinusoidal con frecuencia angular $\omega=3$ rad/seg, y amplitud 4, es decir, la señal a generar será $f(t) = 4\sin(3t)$.

Primer paso: creación del modelo.

Al ejecutar Xcos como se explicó arriba, automáticamente se abrirá (además de la ventana del explorador de paletas), una ventana "**Sin título**" en blanco como la mostrada en la figura 7.4, la cual es la ventana de diseño en donde podemos crear nuestro diagrama de bloques.

En esta ventana seleccionamos el ícono de **guardar modelo** para establecer la ruta y el nombre del archivo en donde se guardará el modelo como 'Sinusoide.zcos'

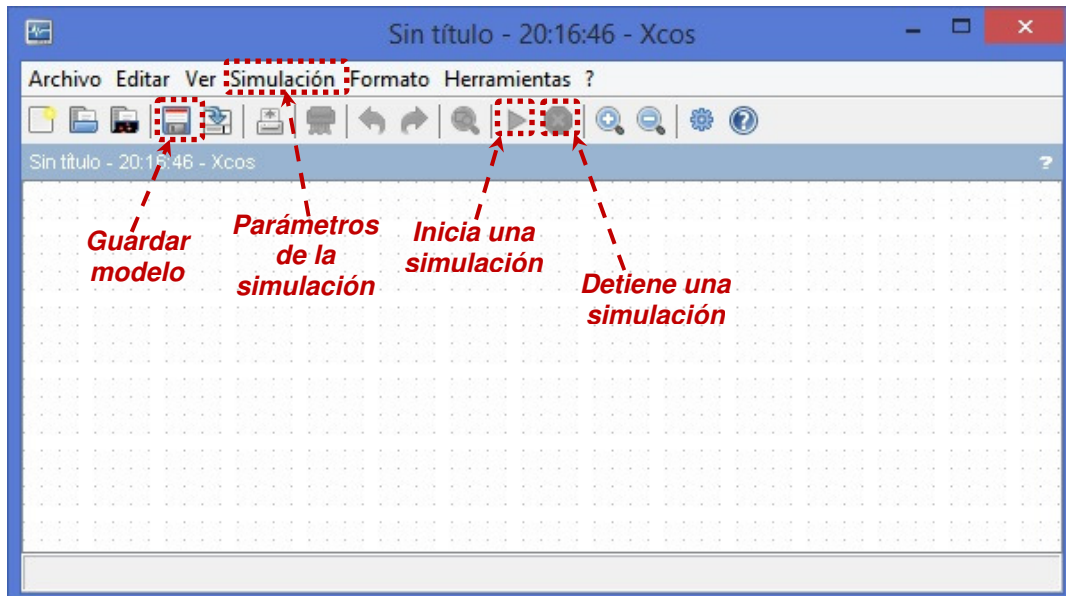


Figura 7.4.- Ventana de diseño para la creación, edición y ejecución de modelos en Xcos.

Segundo paso: introducción de bloques en el modelo.

Los elementos se introducen arrastrándolos con el ratón desde la ventana '*Explorador de Paletas Xcos*' que contiene el listado de los bloques hasta la ventana de diseño. En nuestro caso requeriremos los siguientes bloques:

- Dentro de la categoría '**Fuentes**', el bloque '**GENSIN_f**'. Este elemento corresponde a una señal sinusoidal, que es la que deseamos generar.
- Dentro de la categoría '**Sinks**', el bloque '**CSCOPE**'. Será el que utilizemos para visualizar la señal.
- Dentro de la misma categoría '**Fuentes**', el bloque '**CLOCK_c**'. Este bloque es necesario para especificar al osciloscopio el periodo de muestreo de la señal a desplegar.

Una vez introducidos estos bloques, la ventana de diseño presenta un aspecto como el que se muestra en la Figura 7.5.

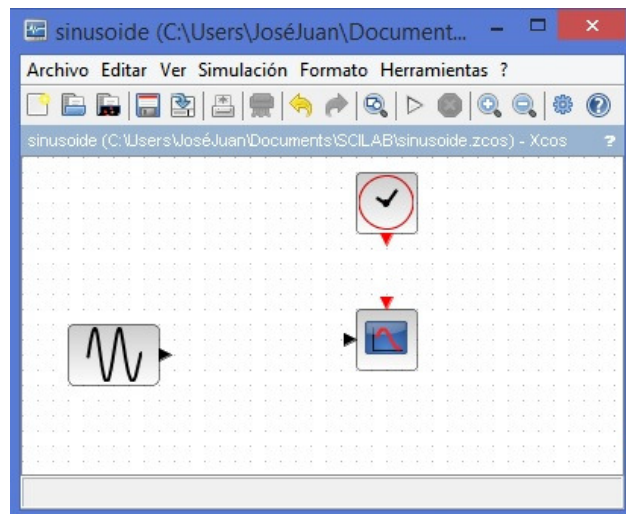


Figura 7.5.- Primeros tres bloques introducidos para el ejemplo 'sinusoide.zcos'.

Tercer paso: interconexión de elementos.

Una vez situados los elementos en la ventana de diseño, es necesario establecer conexiones entre ellos. Para conectar 2 elementos debe hacerse un arrastre con el ratón desde la salida de uno de ellos hasta la entrada del elemento al que se quiere conectar.

Obsérvese que en Xcos hay dos tipos de señales identificadas por el color de los puertos y sólo pueden ser interconectados los puertos del mismo color:

Color negro: Entradas y salidas de señales (Puertos de datos)

Color rojo: Entradas y salidas de Eventos (Puertos de control por eventos o señales de reloj)

Una vez establecidas las conexiones necesarias para nuestro sistema, debemos obtener un resultado como el que se muestra en la Figura 7.6.

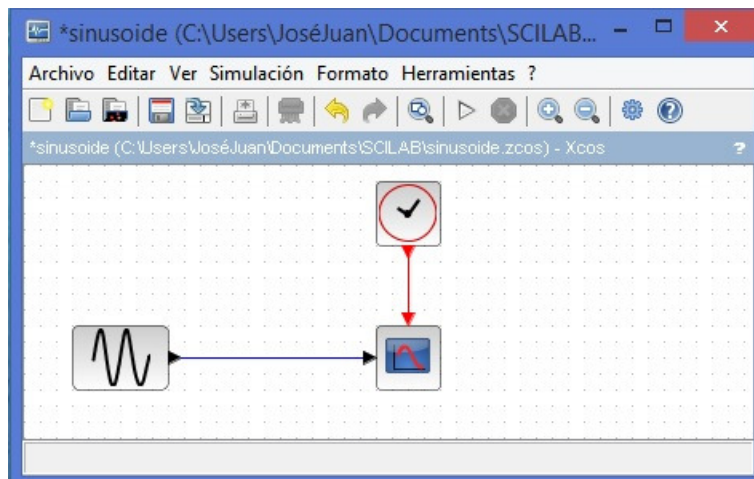


Figura 7.6.- Bloques interconectados del ejemplo sinusoide.zcos.

Cuarto paso: modificación de parámetros de los bloques introducidos.

Parámetros de la simulación. Todo simulador requiere que el usuario tenga una idea exacta o al menos aproximada de los siguientes parámetros de simulación:

- **¿Qué intervalo de tiempo se va a simular?** Para el ejemplo se desean 5 ciclos de la señal $f(t) = 4\sin(3t)$, por lo tanto se requiere un intervalo de 0 a $5 * (2\pi) / 3 \approx 10.47$ seg.
- **¿Qué paso de integración máximo se puede permitir?** para el ejemplo sería deseable tener cuando menos 10 puntos por ciclo, es decir, cada punto estará separado cuando mucho por $0.1 * (2\pi) / 3 \approx 0.21$
- **¿Qué tan exacta se requiere la simulación?** esta es una pregunta difícil en general, normalmente se recomienda iniciar con los valores por default de Scilab y si los resultados no nos satisfacen probar con *solvers* diferentes e ir disminuyendo las tolerancias absolutas y/o relativas.

Para modificar los parámetros de la simulación seleccionamos el submenú "**Configuración**" del menú "**Simulación**" (Ver figura 7.4) y entonces aparece el cuadro de diálogo mostrado en la figura 7.7.

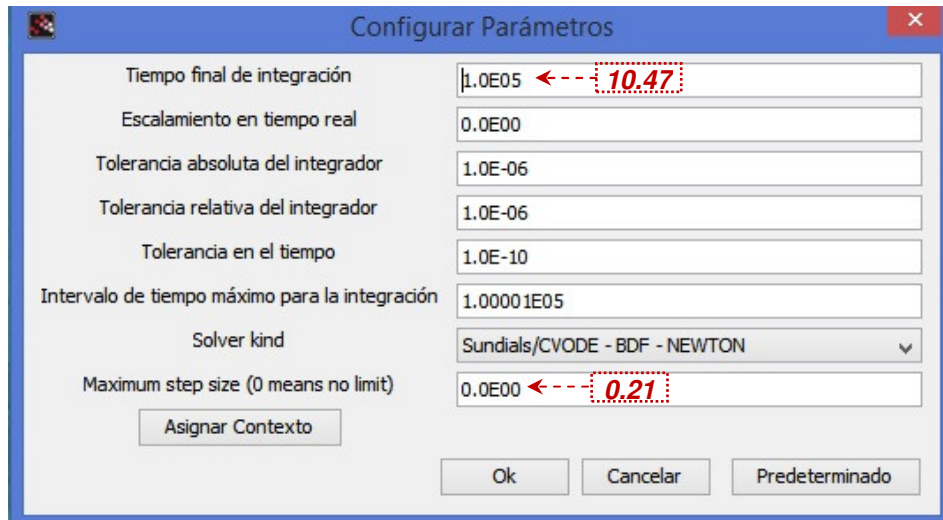


Fig. 7.7.- Parámetros de configuración de la simulación.

Parámetros de los bloques. Todos los bloques de Xcos permiten una cierta configuración que nos permitirá adaptarlos para que funcionen de acuerdo a lo requerido en diferentes aplicaciones:

- **Configuración del bloque GENSIN_f.-** En el bloque generador de la función sinusoidal se puede configurar su amplitud, frecuencia, fase, etc. Para ello se debe hacer doble clic sobre el bloque, con lo que aparecerá una ventana de introducción de parámetros. Para el ejemplo, introduciremos los siguientes valores:
 - Magnitud: 4 (la señal tomará valores de -4 a +4)
 - Frecuencia (rad/s): 3 (frecuencia angular de la señal sinusoidal)
 - Fase: 0 (Angulo de fase respecto del instante inicial $t=0$)
- **Configuración del bloque CSCOPE.-** En el bloque visualizador de señales se pueden configurar varios parámetros. Al hacer doble clic sobre el bloque, aparecerá la ventana de modificación de parámetros mostrada en la figura 7.8.

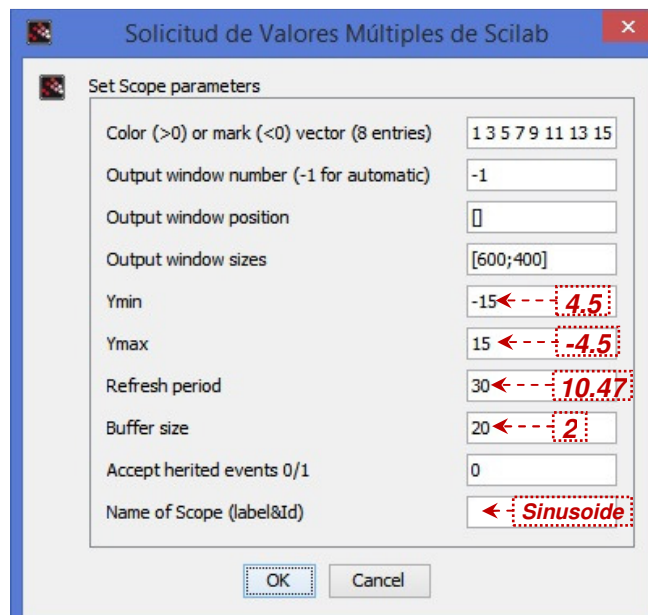


Fig. 7.8.- Ventana de configuración de parámetros del bloque CSCOPE.

A continuación se describe el significado de los parámetros más importantes mostrados en la figura 7.8 y se comentan los valores usados para el ejemplo:

- **Color:** Vector de 8 componentes (para el caso de 8 señales multiplexadas) para una señal solo se usa la primera componente: valor positivo significa color sólido (de acuerdo al mapa de colores **colormap**), valor negativo significa marcador (de acuerdo a la propiedad **mark_style** de **polyline**). Para el ejemplo se dejará el 1 por default pero se sugiere experimentar con otros valores positivos y negativos.
- **Ymin:** Máximo valor del eje vertical. Para el ejemplo se usará 4.5 (ya que la senoide varía de -4 a 4)
- **Ymax:** Mínimo valor del eje vertical. Para el ejemplo se usará -4.5 (ya que la senoide varía de -4 a 4)
- **Refresh period:** Máximo valor del eje horizontal. Si la señal de entrada al bloque supera este valor, la gráfica se redibujará cada múltiplo de este valor. Para el ejemplo solo deseamos visualizar 5 ciclos, por lo tanto este valor es el tiempo final de la simulación, es decir, 10.47seg.
- **Buffer size:** Se usa para aumentar la eficiencia. La gráfica solo se comensará a dibujar hasta que el buffer se llene. Para el ejemplo no se requiere eficiencia, se dejará en su valor mínimo que es 2.
- **Name of Scope:** Aquí podemos colocar el nombre del bloque. Para el ejemplo lo denominaremos "Senoide".

Configuración del bloque CLOCK_c.- El bloque de reloj controla el **periodo de muestreo** del bloque CSCOPE (es decir, el intervalo de tiempo entre los puntos desplegados. Al dar doble click sobre este bloque podemos configurar los siguientes dos parámetros:

- **Periodo:** Periodo de muestreo para el osciloscopio. Para el ejemplo se desean 10 puntos por ciclo de la senoide, por lo tanto este valor se puede poner en 0.21.
- **Tiempo de inicialización:** Instante de arranque del reloj. Para el ejemplo se pondrá en 0.

Quinto paso: Inicio de la simulación y comprobación de resultados.

Lanzaremos la simulación con la opción **'Iniciar'** del menú **'Simulación'** o bien, haciendo clic en el botón triangular de la ventana de diseño (ver figura 7.4). Aparecerá una ventana gráfica como la que se muestra en la figura 7.9.

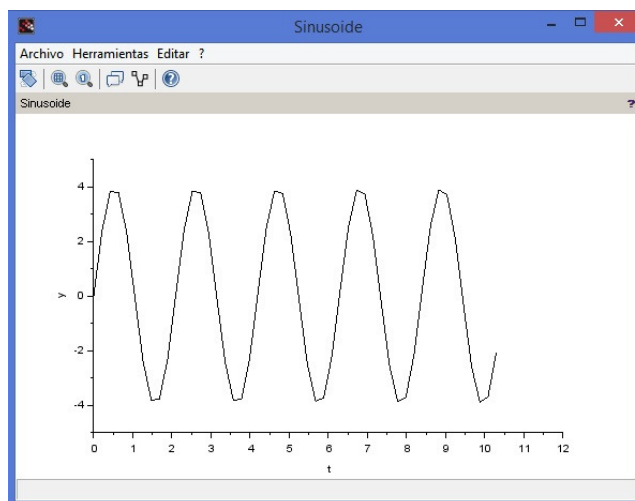


Figura 7.9.- Gráfica visualizada al ejecutar la simulación de senoide.zcos.

Ejercicio. Una vez creado el esquema, se probará la generación de diferentes señales que nos ofrece Xcos dentro de la categoría '**Sources**'. Para ello borraremos del esquema el bloque '**Sine Wave**' y en su lugar insertaremos otro bloque de esa categoría. probar al menos con los bloques siguientes para probar el efecto de modificar sus parámetros:

- **STEP_FUNCTION** (señal escalón)
- **RAMP** (señal rampa)
- **PULSE_SC** (Señal tren de pulsos periodica)
- **RAND_m** (Señal aleatoria con distribución Uniforme o Gaussiana)

Creación de un segundo modelo.

A continuación crearemos un modelo algo más complicado, en el que se realice alguna operación con las señales. Se va a crear un sistema para calcular la integral y la derivada de la señal sinusoidal $f(t) = \sin(3t)$.

Se utilizará el bloque que calcula la integral de una señal, el cual se denomina **INTEGRAL_f** y el que calcula la derivada de una señal, el cual se denomina **DERIV** y ambos se encuentran dentro de la categoría '**Sistemas de tiempo continuo**', además se utilizará un multiplexor, el cual es el bloque denominado **MUX** de la categoría '**Ruteo de señales**'. Con estos bloques se forma el diagrama de la Figura 7.10.

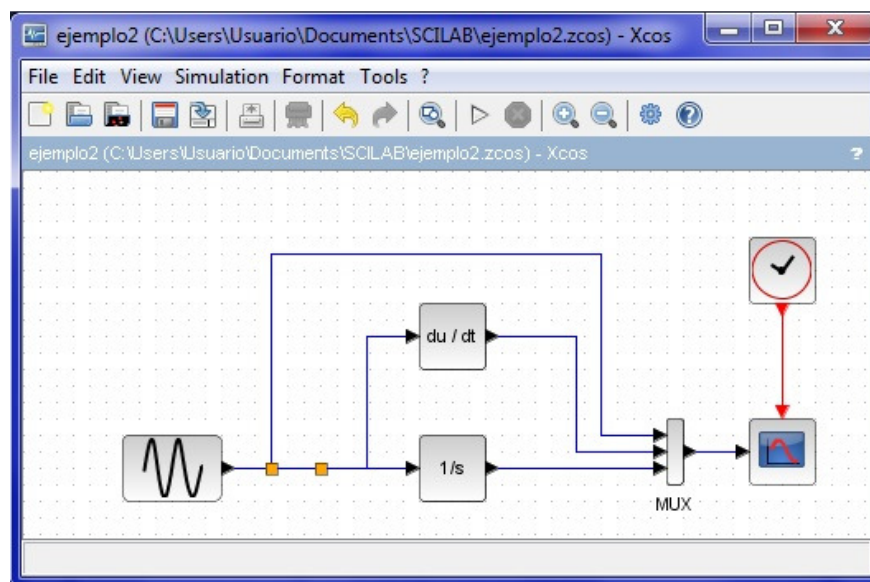


Figura 7.10.- Diagrama de simulación para calcular la integral de una señal pulsante.

Iniciando la simulación, debemos comprobar que en la ventana del osciloscopio se muestran las tres señales: la original $f(t) = \sin(3t)$ (en color negro), su derivada $\frac{d}{dt} f(t) = 3 \cos(3t)$ (color azul) y su integral $\int_0^t f(t) dt = -\frac{1}{3} \cos(3t) + c$ (en color verde), tal y como muestra la figura 7.11

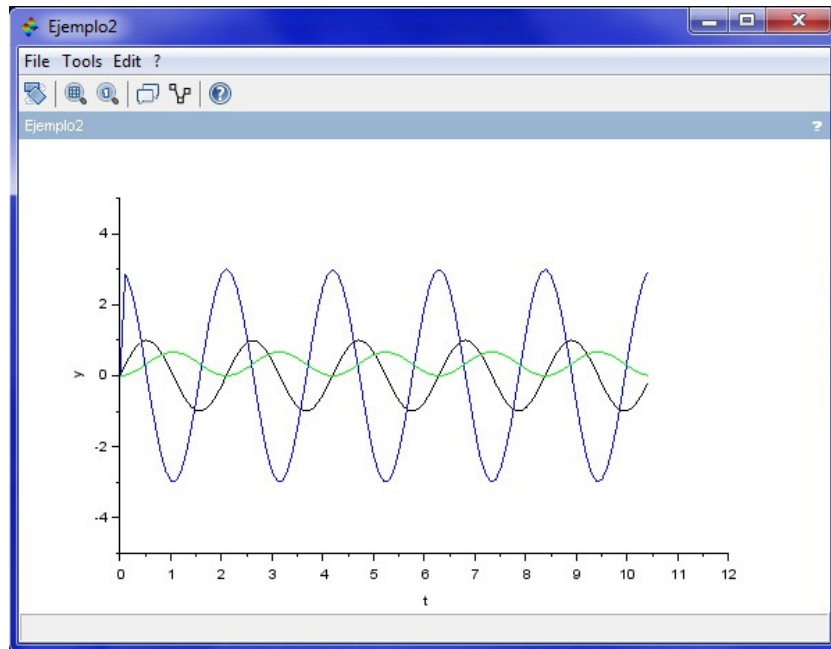


Figura 7.11.- Señal $f(t) = \sin(3t)$, su derivada y su integral.

Obsérvese que de la figura 7.11 se puede ver que el valor de la constante de integración en este caso es $c = 1$, el cual satisface las condiciones iniciales cero.

Simulación de un sistema descrito por una ecuación diferencial

Se desea simular 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 coeficiente de fricción viscosa B , como se describe en la figura 7.12

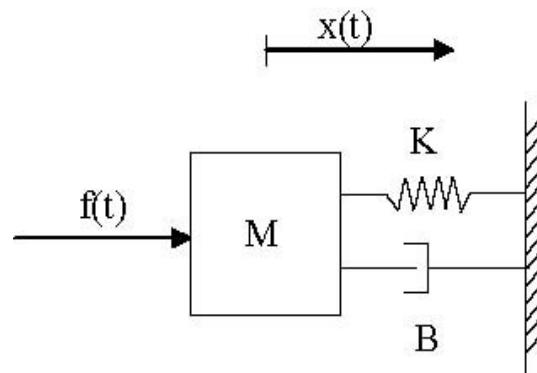


Figura 7.12.- Sistema mecánico masa-resorte-amortiguador.

El objetivo será ver cómo afecta la fuerza aplicada $f(t)$ al desplazamiento de la masa, descrito por $x(t)$.

La ecuación diferencial que rige el comportamiento de este sistema se puede obtener mediante la segunda ley de Newton y es el siguiente:

$$M \frac{d^2 x(t)}{dt^2} + B \frac{dx(t)}{dt} + Kx(t) = f(t) \quad (7.1)$$

Para representar este sistema en Xcos, seguiremos el siguiente procedimiento:

1) **Definir la salida del sistema y despejar su máxima derivada.**- En este caso la salida es $x(t)$ y su máxima derivada se despeja de (7.1), obteniéndose

$$\frac{d^2 x(t)}{dt^2} = -\frac{B}{M} \frac{dx(t)}{dt} - \frac{K}{M} x(t) + \frac{1}{M} f(t) \quad (7.2)$$

2) **Crear un diagrama de simulación con tantos integradores en cascada como el orden del sistema** en donde la salida del sistema es la salida del último integrador.- En este caso el orden del sistema es 2 y la salida del sistema $x(t)$ se toma del segundo integrador. Hasta este punto el diagrama de simulación lucirá como en la figura 7.13. Obsérvese que si la salida del segundo integrador es $x(t)$, forzosamente su entrada deberá ser $\frac{dx(t)}{dt}$, la cual es la salida del primer integrador y por lo tanto la entrada de éste deberá ser $\frac{d^2 x(t)}{dt^2}$

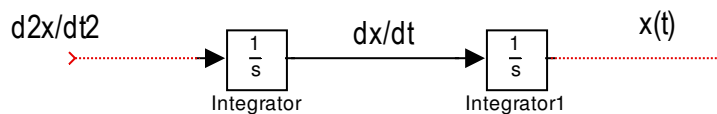


Figura 7.13.- Obtención de $x(t)$ mediante la doble integración de $\frac{d^2 x(t)}{dt^2}$

3) **Asignación de valores a los bloques de ganancias constantes.**- Consideraremos los siguientes valores para los parámetros del sistema: $M=1\text{Kg}$. $K=10\text{ Nt seg/m}$, $B=1\text{ Nt}$. Es posible usar los parámetros simbólicos K , B y M , dentro de los bloques de Xcos, para ello es necesario asignar previamente los valores numéricos que tomarán mediante la opción "**Asignar contexto**" del menú "**Simulación**". Al seleccionar esta opción se despliega el cuadro mostrado en la figura 7.14, en el cual podemos asignar valores a los parámetros, mediante comandos de Scilab



Figura 7.14.- Asignación de valores a parámetros simbólicos a ser usados en los bloques de Xcos.

4) Obtener la máxima derivada requerida en el diagrama anterior mediante bloques de Xcos mediante la ecuación (7.2).- En este caso solo se requieren sumas, restas y multiplicaciones por constante (ganancias constantes). Esto se debe a que el sistema es un SLIT. Con esto completamos el diagrama de simulación para obtener el diagrama de la figura 7.15.

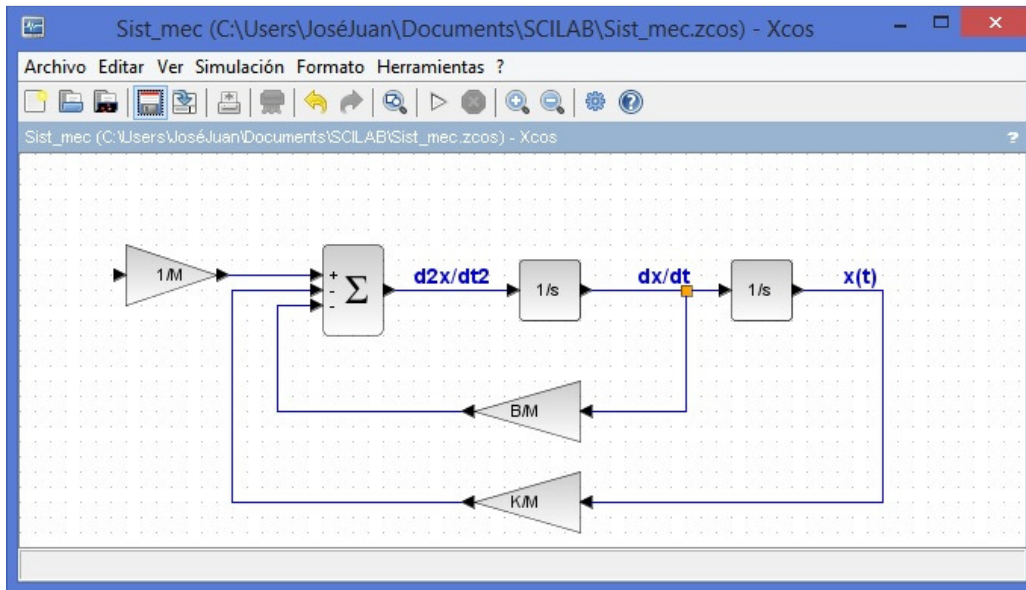


Figura 7.15.- Diagrama de simulación de la ecuación diferencial (7.1)

5) Conexión de una señal de entrada y un bloque de visualización para la salida. Para generar una señal de entrada $f(t)$ en este ejemplo se usará un bloque 'STEP_FUNCTION' para generar una señal escalón unitario y se visualizará la salida mediante un bloque 'CSCOPE'. El diagrama completo queda finalmente como se muestra en la figura 7.16.

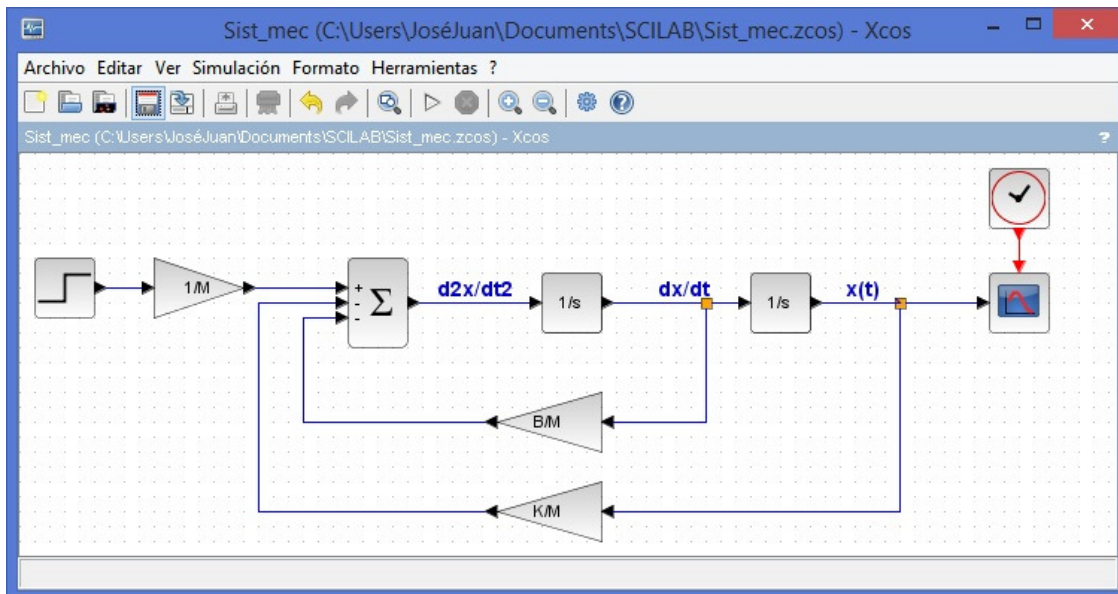


Figura 7.16.- Diagrama de simulación completo para el sistema masa-resorte-amortiguador.

6) Configuración de los bloques y los parámetros de la simulación. Antes de iniciar la simulación es necesario configurar los bloques y los parámetros de la simulación. En este caso se configura lo siguiente:

- 'STEP_FUNCTION' Se configura para generar un escalón unitario que inicia en el instante cero.
- 'CSCOPE' se configura para graficar en color azul un rango de valores de un intervalo vertical de -0.01 a 0.2 metros y un intervalo de tiempo de 10 segundos. También se configura con el título "Posición (metros)".
- 'CLOCK_c' se configura para tomar muestras a graficar cada 0.1 seg desde el instante cero.
- Los parámetros de simulación se dejan los valores por default, excepto el tiempo final de simulación, el cual se pone en 10 segundos.

Una vez arrancada la simulación aparece la gráfica mostrada en la figura 7.17.

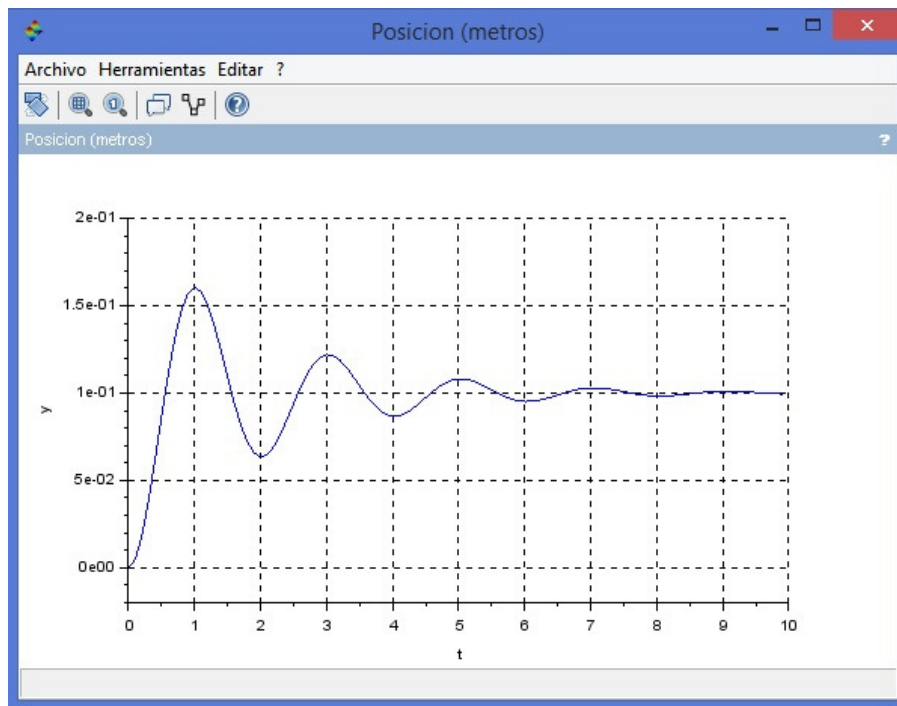


Figura 7.17.- Salida del sistema mecánico $x(t)$ después de la simulación.

Simulación de un Sistema No Lineal.

El procedimiento explicado en este último ejemplo es el más adecuado para simular **Sistemas No Lineales** a partir de su ecuación diferencial no lineal, por ejemplo, para la varilla actuada habíamos obtenido la siguiente ecuación diferencial en la práctica No. 5.

$$\ddot{\theta} = -a_2 \dot{\theta} - a_1 \sin \theta + b_0 \tau \quad (7.3)$$

donde: $a_2 = \frac{3\mu}{ml^2}$, $a_1 = \frac{3g}{2l}$ y $b_0 = \frac{3}{ml^2}$

Usando el procedimiento descrito en el ejemplo anterior obtenemos el diagrama de simulación de la figura 7.18. Previamente se tendrán que definir los parámetros simbólicos mediante la opción "**asignar contexto**", en la cual se pueden definir con comandos Scilab como sigue:

```
m=0.1;
mu=0.01;
L=0.5;
g=9.81;
a2=3*mu/m/L^2;
a1=3*g/2/L;
b0=3/m/L^2;
```

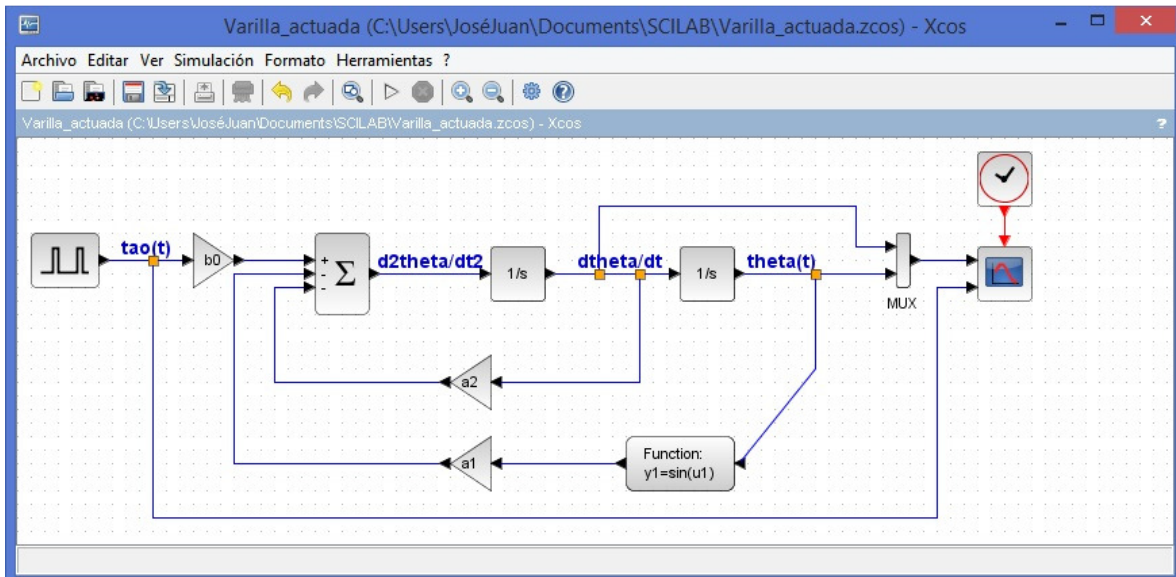


Figura 7.18.- Diagrama de simulación del sistema de la varilla actuada.

En la figura 7.18 se observa que el bloque 'Scope' utilizado tiene dos entradas en lugar de una, esto se debe a que se ha usado el bloque CMSCOPE en lugar de CSCOPE.

Además, en el diagrama de simulación de la figura 7.18 se observa que se está utilizando como entrada un generador de pulsos PULSE_SC. Este bloque se configura para generar un pulso de amplitud 1 y de ancho 0.5 seg que inicia en el instante 0.5 y luego permanece en cero durante el resto del tiempo de simulación (10 seg). Para lograr esto el bloque 'Pulse Generator' se configura al darle doble clic con los valores de los parámetros mostrados en la figura 7.19.

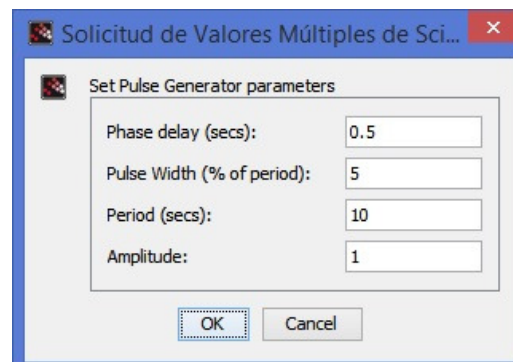


Fig. 7.19.- Configuración del bloque PULSE_SC.

Una vez que se ejecuta la simulación se obtiene la gráfica mostrada en la figura 7.20, en donde se muestra la respuesta de los dos estados del sistema (El ángulo de la varilla en color rojo y la velocidad angular de la varilla en color azul, así como el pulso de entrada). Obsérvese que el resultado coincide con el obtenido en la práctica No. 6. con la instrucción `ode` de Scilab.

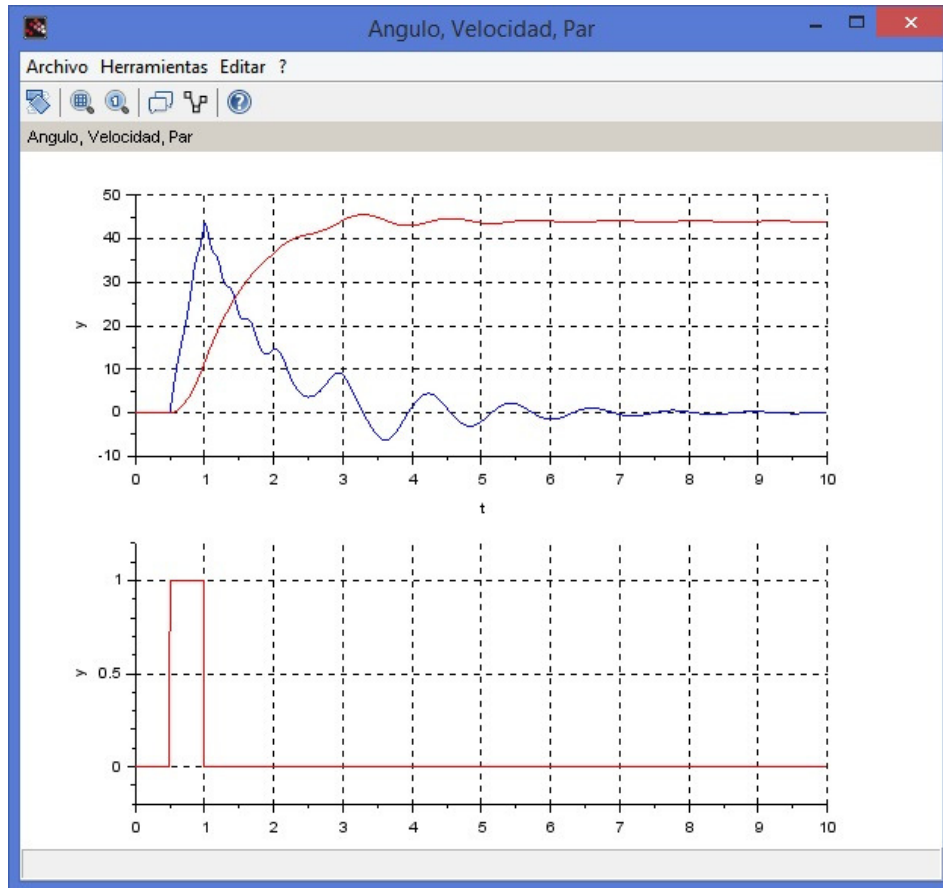


Figura 7.20.- Respuesta de la varilla actuada mostrada por el bloque 'Scope'

Ejercicio. Simular el comportamiento del circuito eléctrico de la figura 7.21.

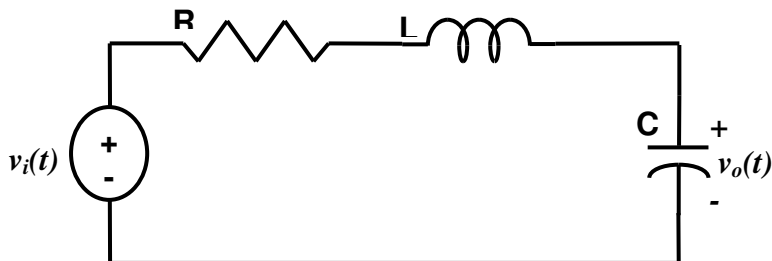


Figura 7.21.- Circuito RLC a simular

Cuya ecuación diferencial es

$$LC \frac{d^2 v_o(t)}{dt^2} + RC \frac{dv_o(t)}{dt} + v_o(t) = v_i(t) \quad (7.4)$$

Se tomará como entrada la señal de voltaje $v_i(t)$ y como salida la señal de voltaje $v_o(t)$ y se probarán los siguientes dos conjuntos de valores para los parámetros del circuito:

Conjunto No. 1	L=500 mH, R=20 ohms, C=500 μ F
Conjunto No. 2	L=500 mH, R=100 ohms, C=500 μ F

Se comprobarán los resultados obtenidos para dos tipos de entradas simulando de cero a 0.5 seg:

- $v_i(t) = \text{escalón unitario}$. Este caso corresponde a la respuesta transitoria del circuito a un voltaje de corriente directa.
- $v_i(t) = \text{senoidal}$. (amplitud 1V y frecuencia 20π rad/seg). Este caso corresponde a la respuesta transitoria del circuito a un voltaje de corriente alterna.

Desarrollo de la Práctica.

1. Probar todos los ejemplos propuestos por el profesor conforme los va explicando.
2. Realizar todos los ejercicios propuestos.
3. Contestar el cuestionario de evaluación de la práctica.

Reportar:

1) El diagrama de simulación elaborado para el ejercicio del circuito RLC y las gráficas obtenidas en las cuatro simulaciones.

2) Obtener la función de transferencia del circuito del ejercicio $G(s) = \frac{V_o(s)}{V_i(s)}$ y comparar los resultados obtenidos en Xcos ante una entrada escalón unitario utilizando el comando `csim('step', ...)` de Scilab.

3) En la respuesta del circuito del ejercicio ante una entrada de corriente alterna se pueden apreciar dos partes de la respuesta La primera parte es la respuesta transitoria, la cual dura un tiempo t_s y luego queda una respuesta puramente sinusoidal que dura hasta el infinito, llamada respuesta en estado estacionario o estable.

3a) ¿cuánto vale aproximadamente t_s ?

3b) Extiende la simulación hasta un tiempo grande, por ejemplo 5 seg y observa ¿cual es la amplitud de la respuesta en estado estacionario?