

# Programación en MATLAB

## Clase 6

Dr. Ing. Rodrigo Gonzalez  
rodraz@frm.utn.edu.ar

Universidad Tecnológica Nacional,  
Facultad Regional Mendoza.

# Resumen

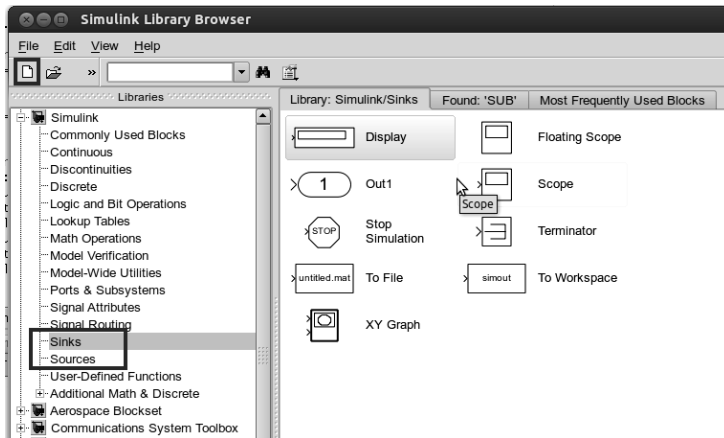
- 1 **Simulink**
  - Introducción
  - Bloques básicos
  - Ejemplo 1
  - Ejemplo 2
  - Ejemplo 3
  - Model properties
  - Subsistemas
- 2 **Solvers**
  - Tipos de solvers
  - Tipos de tiempos de muestreo
- 3 **Función sim()**

- Simulink es parte de la suite de programas de MATLAB.
- Interfaz gráfica para modelado de sistemas.
- Requiere de MATLAB para su ejecución.
- Contiene una gran biblioteca de *bloques*.
- Los bloques equivalen a las funciones de MATLAB.



En Simulink hay tres tipos de bloques:

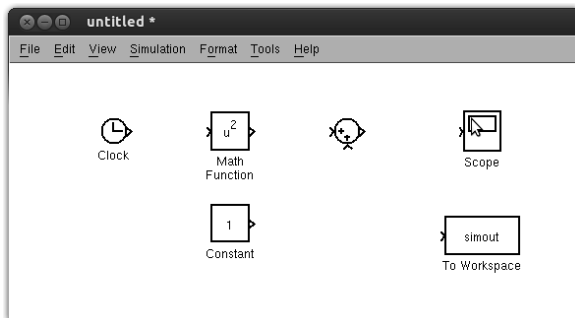
- Fuentes (*Sources*).
- Sumideros (*Sinks*).
- Y todos los demás.



Represente en Simulink la siguiente ecuación:

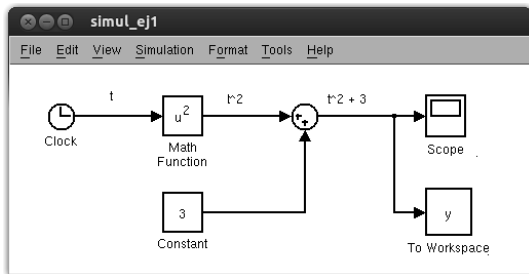
$$y = t^2 + 3$$

- Cree un nuevo modelo (CTRL+N).
- Busque en *Math Operations* los bloques necesarios.
- De *Source* agregue [Clock] y [Constant].
- De *Sink* agregue [Scope] y [To Workspace].



$$y = t^2 + 3$$

- Conecte los bloques de manera que representen la función deseada.
- En [Constant], *Constant value = 3*.
- En [Math Function], *Function: square*.
- En [To Workspace], *Save format: Array*.
- *Simulation > Configuration Parameters* (CTRL+E).
  - Verifique *Simulation Time*.
- *Simulation > Start* (CTRL+T).

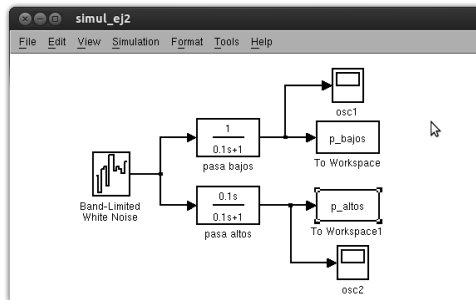


## Ejemplo 2

Divida una señal de ruido blanco en altas y bajas frecuencias, con las siguientes funciones de transferencia:

$$pasa\_bajos = \frac{1}{0.1s + 1} \quad , \quad pasa\_altos = \frac{0.1s}{0.1s + 1}$$

- Cree un nuevo modelo (CTRL+N).
- Busque en *Continuous* el bloque [Transfer Function].
- De *Source* agregue [Band-Limited White Noise].
- De *Sink* agregue [Scope] y [To Workspace].



## Ejemplo 3

La ecuación general de una masa con resorte responde a la ecuación,

$$m y'' + c y' + k y = F(t)$$

Donde,

$y(t)$ , posición del resorte.

$y_0 = 10$  m, posición inicial.

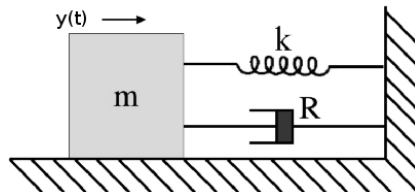
$v_0 = 0$  m, velocidad inicial.

$m = 1$  kg, masa.

$c = 0.1$  kg/s, coeficiente de fricción.

$k = 1$  N/m, constante del resorte.

$F(t) = 0$  N, fuerza externa aplicada al resorte.



Describe este sistema en Simulink.

**Importante:** se debe evitar el uso del bloque `Derivative`, ya que al derivar una señal se introduce ruido en la misma.



## Ejemplo 3

$$m y'' + c y' + k y = F(t) \rightarrow y'' = \frac{F(t)}{m} - \frac{c}{m} y' - \frac{k}{m} y$$

$y(t)$ , posición del resorte.

$y_0 = 10$  m, posición inicial.

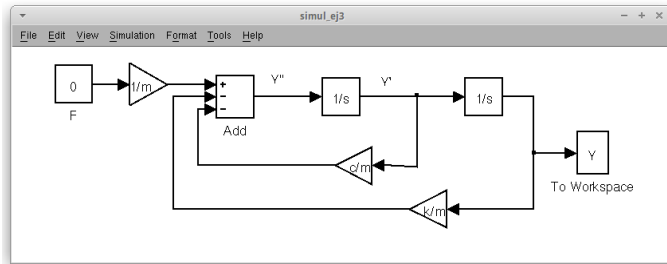
$v_0 = 0$  m, velocidad inicial.

$m = 1$  kg, masa.

$c = 0.1$  kg/s, coeficiente de fricción.

$k = 1$  N/m, constante del resorte.

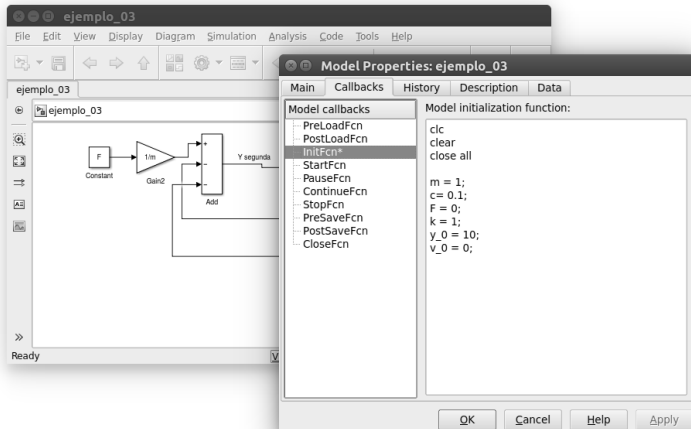
$F(t) = 0$  N, fuerza externa aplicada al resorte.



# Model properties

Es posible ejecutar funciones en MATLAB durante diferentes etapas de la simulación de un modelo en Simulink.

*File -> Model properties -> Model properties -> Callbacks*

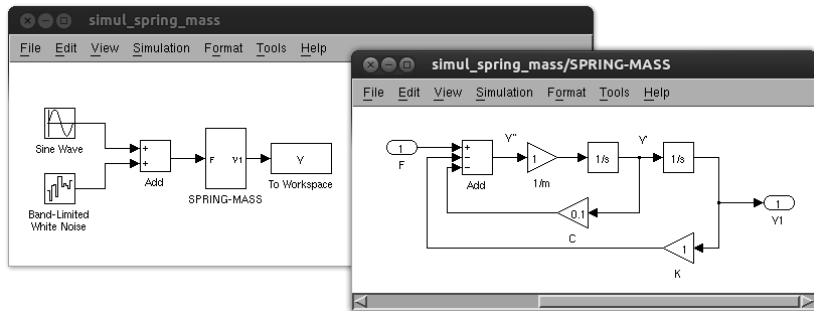


# Subsistemas

Cuando un modelo tiene varios componentes es conveniente dividirlo en subsistemas. Por ejemplo, ejecutar » `aeroblk_HL20`.

Para crear un subsistema:

- Seleccionar todos los elementos de un modelo (CTRL+A).
- *Edit/Create Subsystem* (CTRL+G).
- De *Source* agregue puerto de entrada [In1].
- De *Sink* agregue puerto de salida [Out1].

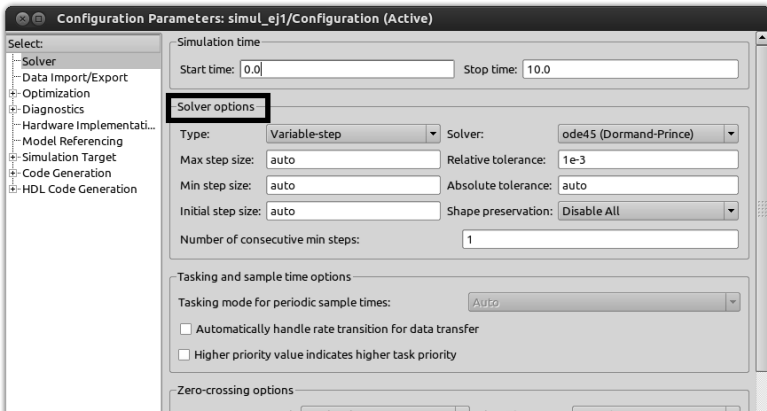


## Tipos de solvers

Simulink cuenta con varios *solvers* según las características de nuestro modelo. El *solver* se fija en *Simulation Parameters* (CTRL+E).

Tipos de *solvers*:

- 1 Variable-step, continuous solvers (por defecto).
- 2 Fixed-step, continuous solvers.
- 3 Variable-step, discrete solver.
- 4 Fixed-step, discrete solver.



# Solvers continuos

- Adecuados para sistemas que contienen estados continuos (Por ej. variable  $s$ ).
- Utilizan métodos de integración numérica (quad, quadl, quadgk).
- Si un modelo es híbrido (contiene estados continuos y discretos) se usan solvers continuos.

## Variable-step continuous solvers:

- Simulink varía dinámicamente el tiempo de muestreo. Se ahorra tiempo de simulación.
- El tiempo de muestreo se incrementa o reduce según tolerancia a un error prefijado.
- Para EDOs no rígidas: ode45, ode23 y ode113.
- Para EDOs rígidas: ode15s, ode23s, ode23t y ode23tb.
- *A generally accepted definition of a stiff system is a system that has extremely different time scales.*

## Fixed-step continuous solvers:

- Tiempo entre pasos de simulación constante.
- ode1, Método de Euler.
- ode2, Método de Heun, o método de Euler mejorado.
- ode3, Método de Bogacki-Shampine.
- ode4, RK4, Método Runge-Kutta de orden 4.
- ...

# Solvers discretos

- Útiles para sistema sin estados continuos (Por ej. variable  $z$ ).
- No utilizan métodos de integración numérica.
- Utilizan métodos de detección de cruce por cero (*Zero-Crossing Detection*) para detectar discontinuidades y así aumentar el tiempo de muestreo.

## Variable-step discrete solver:

- Útil para modelos discretos cuyos bloques tienen diferentes tiempos de muestreo.
- Usado por defecto si el modelo no tiene estados continuos.
- Se actualizan las salidas del sistema solo si han cambiado de estado.

## Fixed-step discrete solver:

- Tiempo entre pasos de simulación constante.
- Se actualizan las salidas del sistema aunque no hayan cambiado de estado.

## Ejercicio 1, sistema continuo

Tome el modelo de simulink del sistema masa-resorte y prueba su funcionamiento para diferentes tiempos de muestreo continuos, a saber:

- 1 **Type:** Variable-step, Rel. tol. =  $1e-3$ , Abs. tol. = auto (por defecto).  
**Solver:** ode45.
- 2 **Type:** Variable-step, Rel. tol. =  $1e-3$ , Abs. tol. =  $1e-6$ .  
**Solver:** ode45.
- 3 **Type:** Variable-step, Rel. tol. =  $1e-3$ , Abs. tol. = auto, Max step size = 0.5.  
**Solver:** ode45.
- 4 **Type:** Fixed-step. **Solver:** ode1.
- 5 **Type:** Fixed-step. **Solver:** ode4.
- 6 **Type:** Fixed-step. **Solver:** ode4, fixed-step = 0.001.

Analice la cantidad de elementos de los vectores de tiempo para todos los casos.  
¿Qué observa?

## Ejercicio 2, sistema discreto

Transforme el modelo de simulink del sistema masa-resorte en un sistema discreto usando un integrador discreto (*Discrete / Unit Delay*).

Pruebe su funcionamiento para diferentes tiempos de muestreo discretos, a saber,

- 1 **Type:** Variable-step. **Solver:** discrete solver.
- 2 **Type:** Fixed-step. **Solver:** discrete solver. **Fixed-step size:** auto.
- 3 **Type:** Fixed-step. **Solver:** discrete solver. **Fixed-step size:** 0.5.
- 4 **Type:** Fixed-step. **Solver:** discrete solver. **Fixed-step size:** 0.001.

Analice la cantidad de elementos de los vectores de tiempo para todos los casos.  
¿Qué observa?



El tiempo de muestreo de un bloque es un parámetro que indica cuando, durante la simulación, el bloque debe producir una salida y actualizar sus estados internos.

- **Tiempo de muestreo discreto.** Se fija un tiempo en segundos.

$$t_n = n T_S + |T_0|, \quad 0 \leq n \leq \frac{T_{SIM}}{T_S}$$

Donde:

- $t_n$ , tiempo de muestreo discreto.
- $T_S$ , tiempo entre muestras.
- $|T_0|$ , tiempo inicial.
- $T_{SIM}$ , tiempo total de simulación.

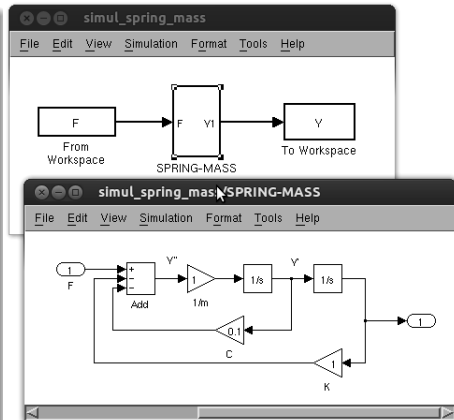
Ej: *Sample Time*: 0.01.

- **Tiempo de muestreo continuo.** El tiempo se divide en pasos grandes y pasos chicos. El solver determina el tamaño de estos pasos. La salida de un bloque solo entrega valores en pasos grandes. *Sample Time*: [0, 0].
- **Tiempo de muestreo heredado** (*Inherited*). Simulink determina el mejor tiempo de muestreo según el contexto en el que se encuentra el bloque. *Sample Time*: -1.
- **Tiempo de muestreo constante.** El bloque se ejecuta solo durante la inicialización del modelo. *Sample Time*: inf.
- Y más...

La función `sim()` permite ejecutar la simulación de un bloque de Simulink desde un programa o función de MATLAB.

#### Ejemplo 4

```
1 TStart = 0;
2 TFinal = 300;
3 t = (TStart:TFinal)';
4
5 % Estructura para estimular al sistema
6 F.time = t;
7 F.signals.values = zeros(size(t));
8 % F.signals.values = sin(t);
9 F.signals.dimensions = 1;
10
11 sim('simul_spring_mass' ...
12 [TStart TFinal] );
13
14 plot(tout, Y, 'k');
```



## Ejercicio 3

La ecuación que representa la velocidad de un cuerpo cayendo hacia la Tierra es

$$\frac{dv}{dt} = g - \frac{c}{m} v^2$$

Donde,

$g = 9.81 \text{ m/s}^2$ , aceleración debida a la gravedad.

$v(t)$ , velocidad.

$v(0) = 0 \text{ m/s}$ , velocidad.

$m = 70 \text{ kg}$ , masa.

$c = 0.3 \text{ kg/m}$ , coeficiente.

Cree un subsistema con la ecuación en Simulink cuya entrada sea  $m$  desde el *Workspace* y la salida sea  $v(t)$ . Simule la velocidad en función del tiempo para los primeros 15 segundos. Ejecute el modelo en Simulink desde MATLAB. Grafique.

## Ejercicio 3, solución

$$\frac{dv}{dt} = g - \frac{c}{m} v^2$$

