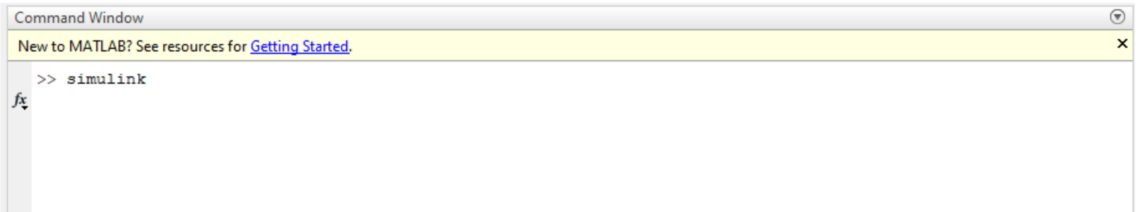
Laboratorio de Sistemas y señales I

En el presente Laboratorio se expone los conceptos relacionados con la herramienta computacional denominada **SIMULINK**, así como también se proporciona una descripción como ayuda rápida para facilitar la utilización del paquete computacional, con la finalidad de lograr la implementación de modelos propios y realizar simulaciones de los sistemas que resulten de interés.

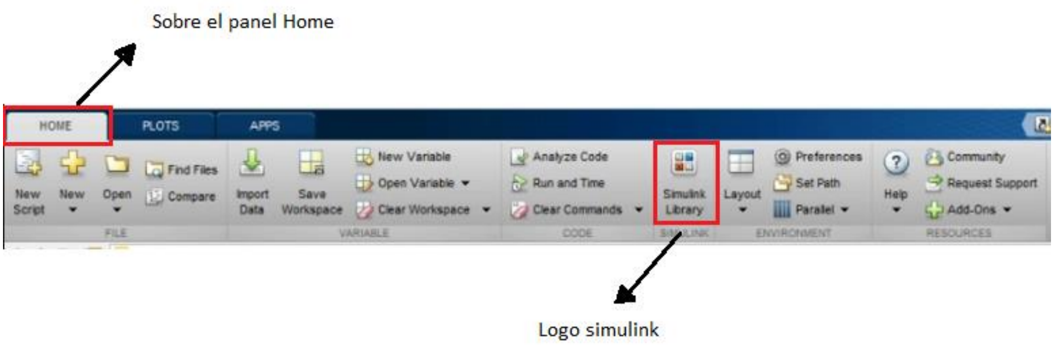
¿Qué es Simulink? Es una extensión de MATLAB que puede ser definida como un software que sirve para modelar, diseñar y analizar sistemas. Es un entorno gráfico en el cual el modelo a simular se construye seleccionando y arrastrando los diferentes bloques que lo constituyen. Soporta tanto sistemas lineales como no-lineales, modelos en tiempo continuo y en tiempo discreto.

¿Cómo se inicializa? Existen dos formas de iniciar el software desde MATLAB.

1- Desde la línea de comandos escribiendo >>simulink



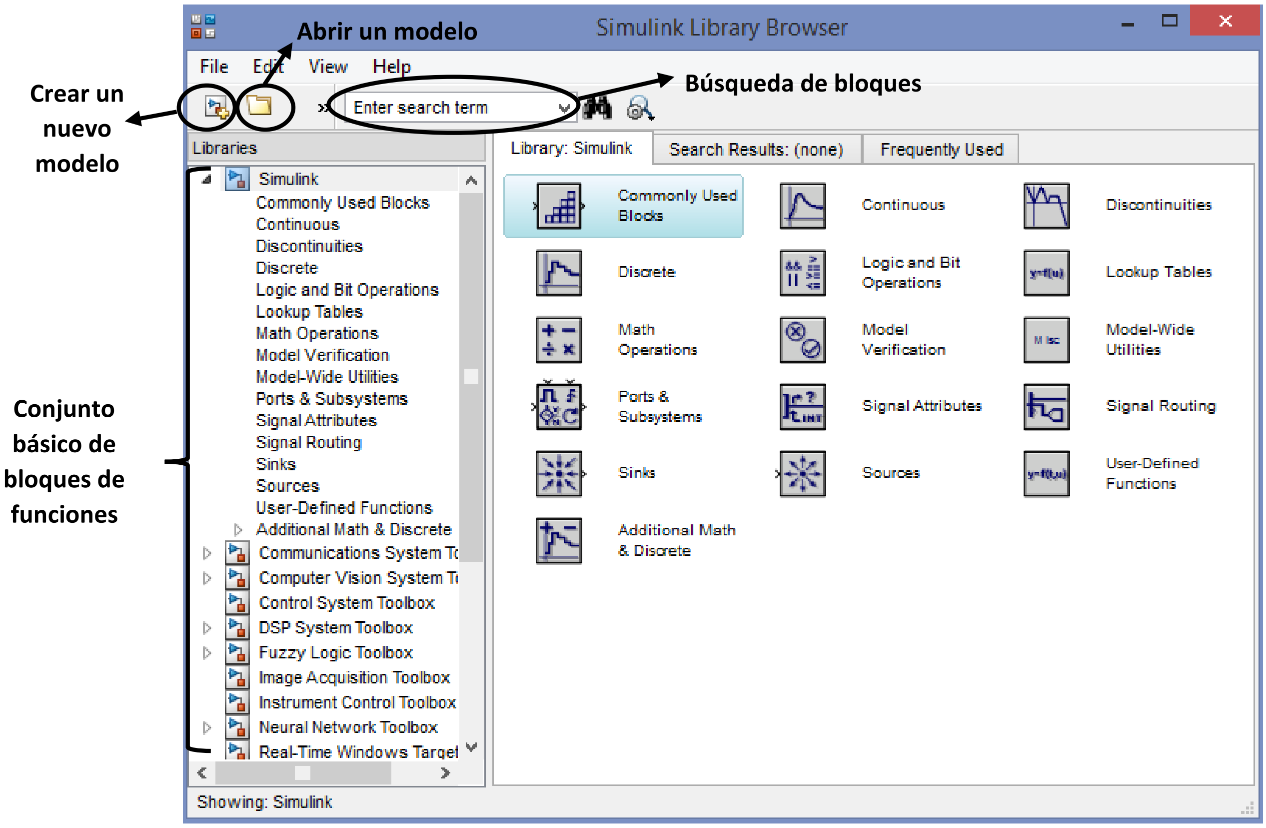
2- Desde el panel de control seleccionando sobre el logo.



Una vez iniciado el programa, el entorno de trabajo queda dividido en tres partes:

- La ventana de comandos de Matlab: desde la que se puede ejecutar cualquier comando del mismo, dar valores a variables y controlar la ejecución de las simulaciones.

- La ventana de la biblioteca de Simulink: desde la que se seleccionan los componentes que se van a insertar en el sistema a simular.

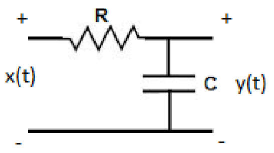


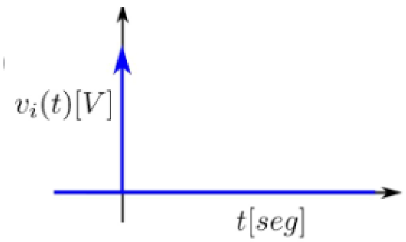
- La o las ventanas de los modelos: en las que se dibujan los modelos y se realizan y controlan las simulaciones.

Una vez expuesto esto, sin entrar tanto en detalle, se verifica si resulta sencillo llevar a cabo una simulación de un sistema LTI de primer orden.

Sistemas LTI Continuo de primer orden

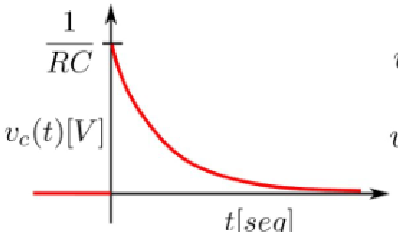
Recordando algunas cuestiones básicas de los sistemas de primer orden:





$$v_i(t) = \delta(t)$$

$$\frac{Y(s)}{X(s)} = \frac{\frac{1}{RC}}{s + \frac{1}{RC}} = H(s)$$

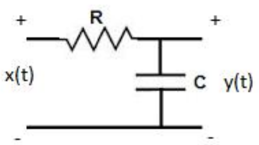


$$v_c(t) = ae^{-at}\ \mathrm{U}\ (t)$$

$$v_c(t) = \frac{1}{RC} e^{-t/(RC)}\ \mathrm{U}\ (t)$$

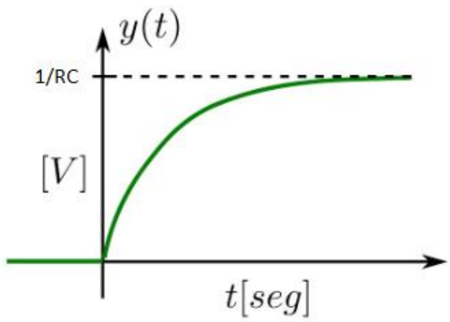
x(t) - > $v_i(t)$

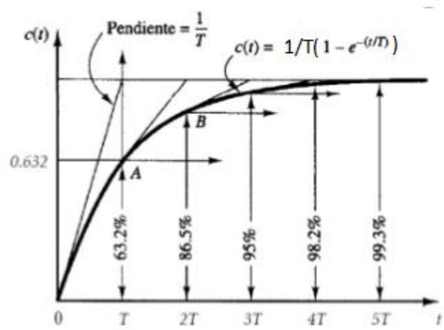
y(t) - > $v_c(t)$



$$x(t) = U(t)$$

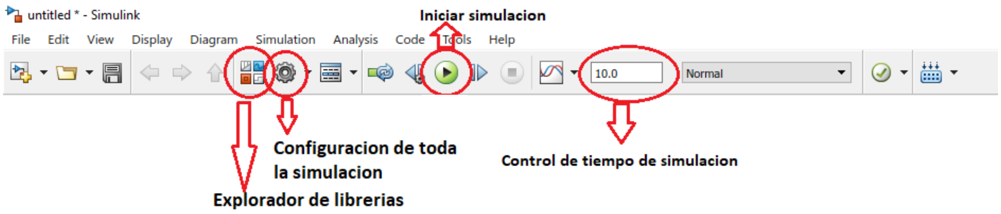
$$y(t) = \frac{1}{RC}\left(1 - e^{-t/(RC)}\right) U(t)$$



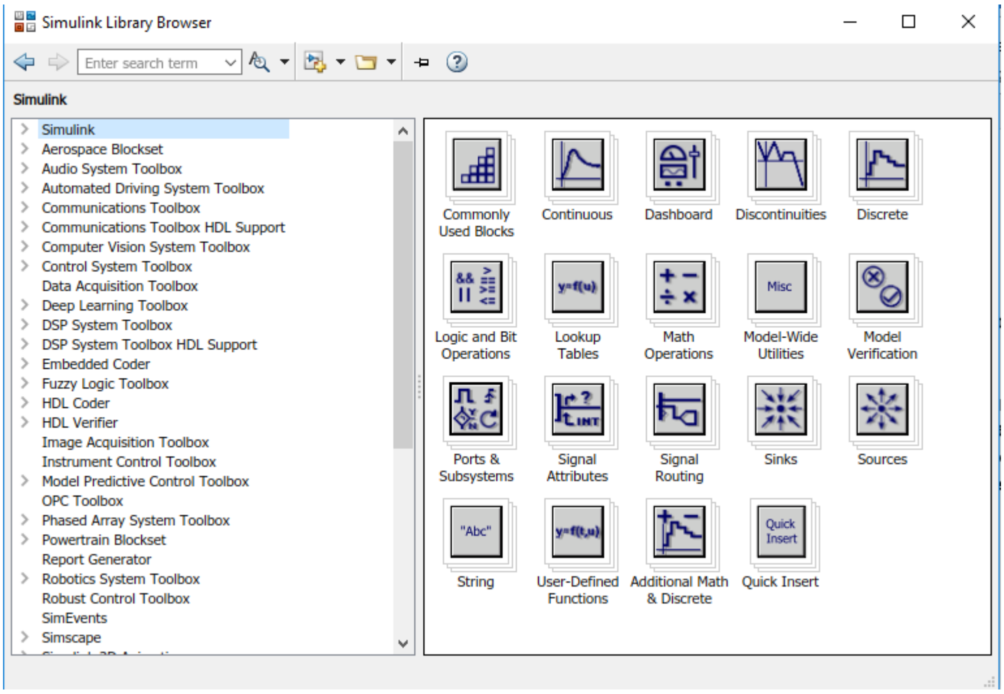


Curva de respuesta exponencial

Se inicializa simulink y el explorador de librerías (“Browser library”).

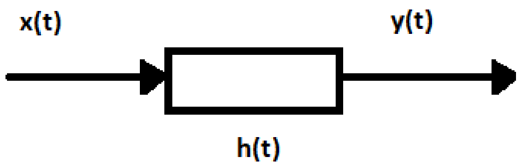


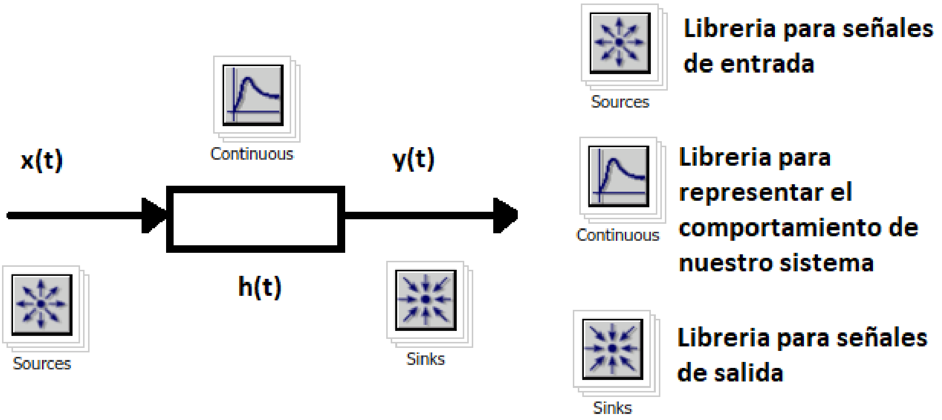
Se visualizará la siguiente ventana:



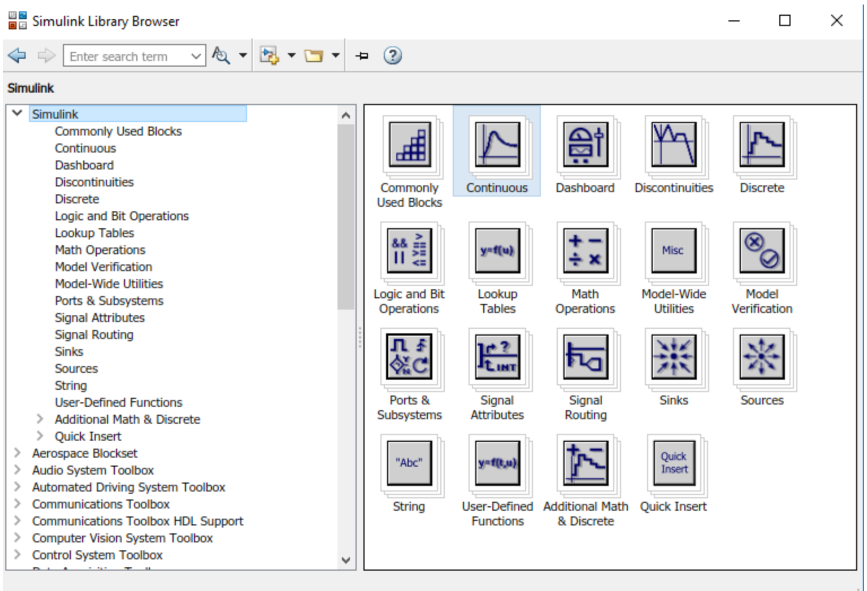
¿Qué se necesita para esta simulación?

En primer lugar, se debe representar gráficamente las relaciones entre las variables de un sistema. Por lo tanto, se buscará una señal de entrada “**x**”, una función que represente la respuesta al impulso del sistema “**h**” y una señal de salida “**y**”. Estos componentes se encuentran dentro de las distintas librerías de simulink. Es decir, que para construir el sistema se analizara en el siguiente esquema:

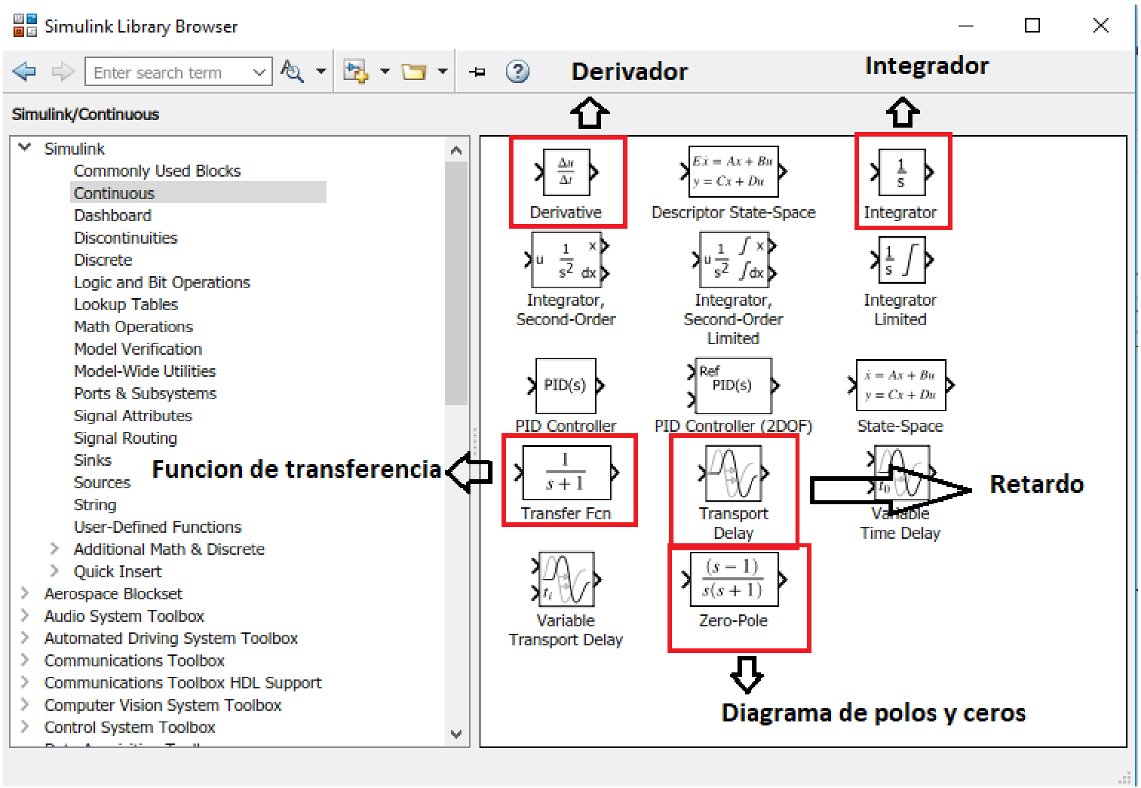




A partir de lo expuesto, se buscarán los distintos componentes. Resulta muy intuitivo obtener primero la respuesta impulsiva del sistema.

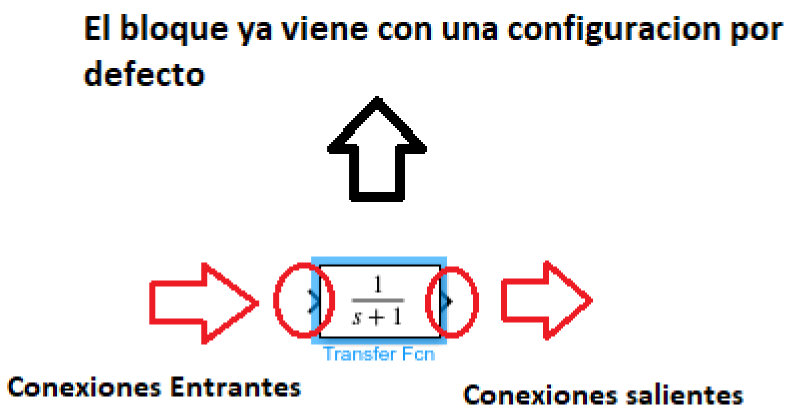


Los bloques que se encuentran disponibles son:



Dentro de la gran cantidad de bloques que se observan, los más simples de analizar e interpretar son los que están marcados en rojo, ya que al verlos rápidamente se puede verificar que función cumplen. Entonces se podría plantear en un primer momento, sin saber demasiado acerca de los bloques, usar alguno de ellos. ¿Cuál se deberían elegir? Los bloques más familiares son "*Transfer Fcn*" y "*Zero-Pole*". ¿Cómo se aplican al proyecto? Solo hay que seleccionarlo y arrastrarlo hasta la hoja de trabajo.

Al realizar dicha operación, en la hoja de trabajo aparecerá el bloque correspondiente con algunos detalles:



El sistema se encuentra caracterizado mediante su función de transferencia en el dominio de S, es decir, mediante la transformada de Laplace. Pero ¿No se había planteado que se buscaba caracterizar al sistema mediante su respuesta impulsiva en el dominio del tiempo **h(t)** para lograr aplicar en la entrada **x(t)**? Sí, es verdad, eso se estableció. Porque como es de esperar, si el sistema es caracterizado por su función de transferencia H(S) se debe ingresar en la entrada X(S), es decir, transformar la entrada x(t) del dominio en el tiempo al dominio de S. De esta manera, se debe respetar la siguiente relación:

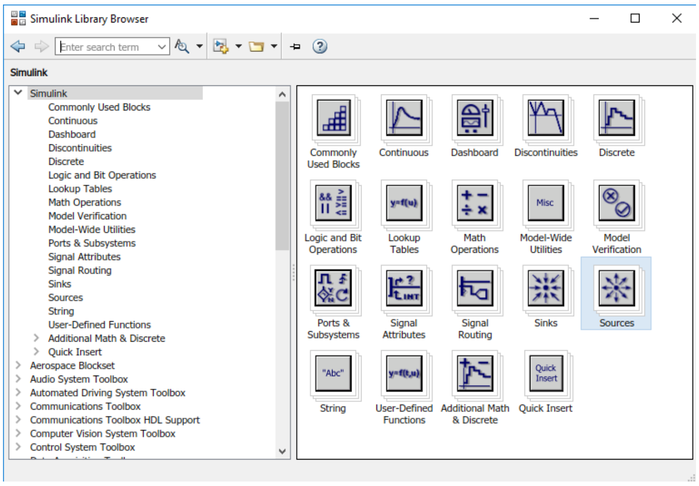
$$y(t) = x(t) * h(t)$$

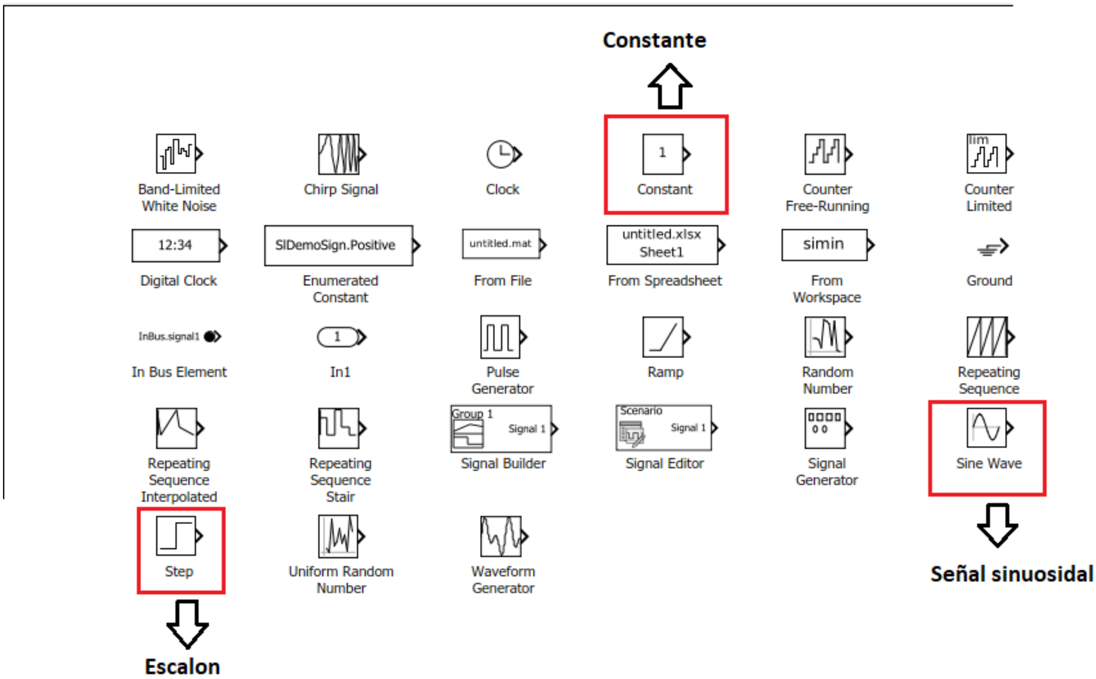
$$Y(S) = X(S) \,.\, H(S)$$

Resulta que simulink es flexible en cuanto a los requerimientos que le exige a los usuarios para armar las simulaciones, pero en su funcionamiento interno respeta las relaciones que se establecieron anteriormente. El software requiere que la entrada se exprese en el dominio del tiempo **x(t)**, lo que es netamente razonable, ya que generalmente se tendrán como dato este tipo de señal y además, para simplificar la construcción del sistema, se requiere la función de transferencia, es decir, la forma clásica con la que se suelen expresar los sistemas. Entonces, queda expresada la entrada en el dominio del tiempo y el sistema en el dominio de S. Parece mezclarse todo, pero simulink al iniciar la simulación, automáticamente construye la respuesta al impulso a partir de la resolución de una ecuación diferencial mediante un método numérico y opera todo el sistema en el dominio del tiempo para entregar en la salida y(t).

En conclusión, si bien parece mezclarse los dominios de trabajo, no hay ningún tipo de inconveniente porque simulink solamente utiliza el dominio temporal para resolver la simulación.

Aclarado esto, se buscan las señales de entrada:

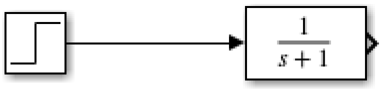




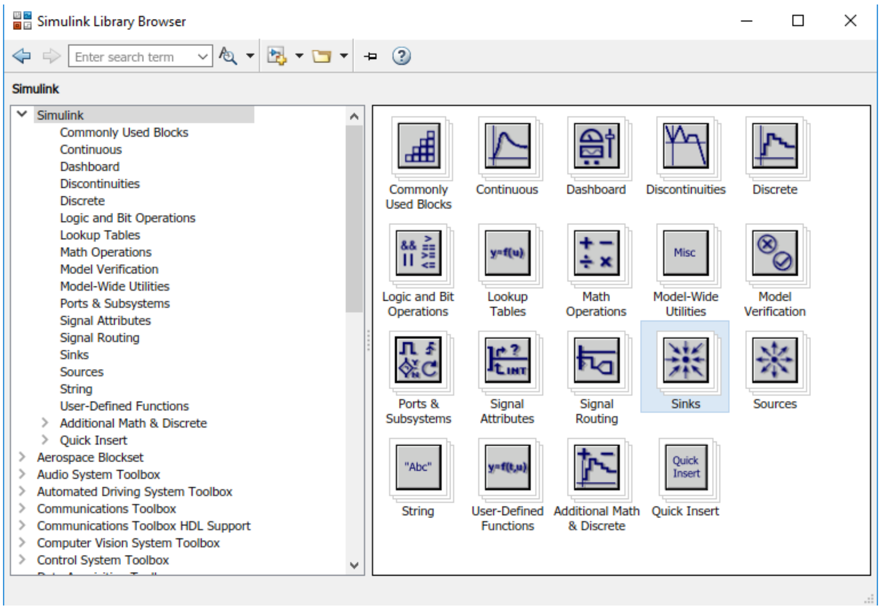
En este caso, a simple vista es más difícil determinar que bloques son más familiares. Si bien se entiende que significa “Signal Generator”, no se sabe cómo opera, ni que señales puede generar. Al igual que “Pulse Generator”. Se conoce que es, pero no cómo se configura ni que parámetros tiene como referencia. Se debería investigar desde el help de Matlab. Así, se podría mencionar algunos bloques más. Sin embargo, hay 3 bloques que a simple vista resultan fáciles de deducir. “Step” , “Sine Wave” y “Constant”. Para el sistema de primer orden es importante la respuesta al escalón, entonces se usará el bloque “step”.

Está presente en la figura anterior la entrada y el sistema, evidentemente se debe realizar alguna operación para que los bloques estén vinculados. ¿Qué se debería hacer? Simplemente cliquear en la salida del escalón y mantener hasta conectar con la entrada del sistema. Si por esas casualidades se suelta antes, se producirá la siguiente flecha:

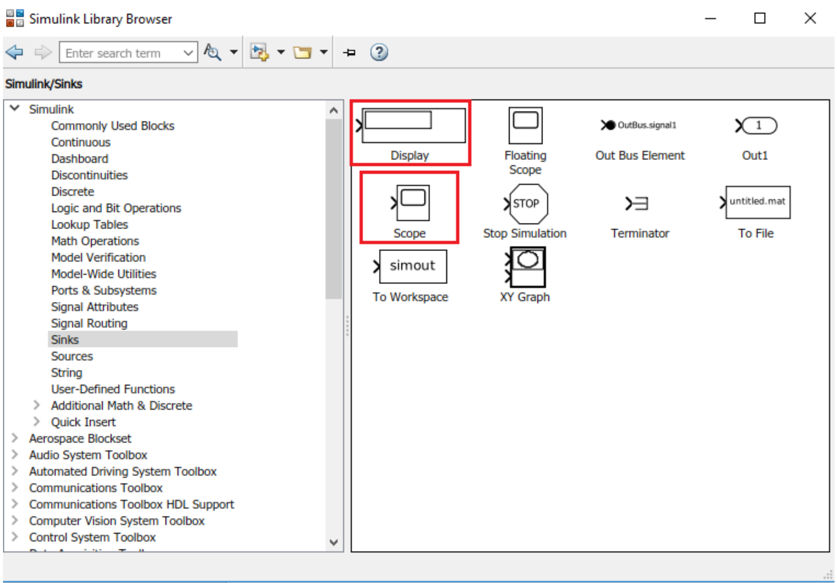
Para continuar y solucionar el problema, se debe cliquear y mantener sobre la punta de la flecha y continuar hasta que se genere la conexión.



¿Estará bien el escalón que se está utilizando? ¿Cómo se conoce que amplitud tiene y donde comienza? ¿Sera un escalón de $1U(t)$ó $1U(t-2)$ ó $0.5U(t)$ ...…? La solución es tratar de graficar dicha señal generada. Es decir, se observará de alguna forma que señal está generando Simulink. ¿Cómo se hace? Hay que buscar en la librería de señales de salida

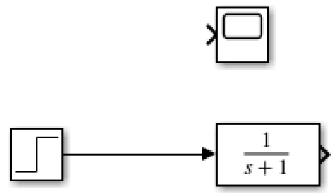


Dentro de la librería “Sinks” se encuentra:

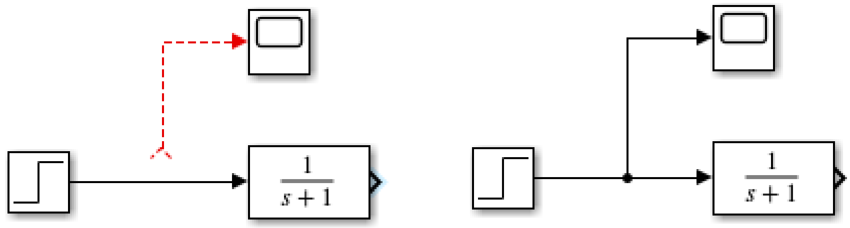


Los bloques más simples de utilizar son “Scope” y “Display”. El primero es un osciloscopio y el segundo es un visor que permite observar momento a momento que valor asume la señal de entrada. Cabe mencionar que son diferentes. Porque mediante “Scope” se observan todos los valores de la señal en el dominio del tiempo, en cambio con “Display” solo se permite visualizar los valores actuales de la señal.

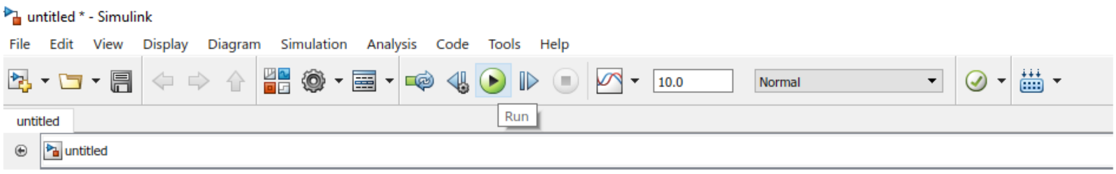
Se usará “Scope” en el proyecto:



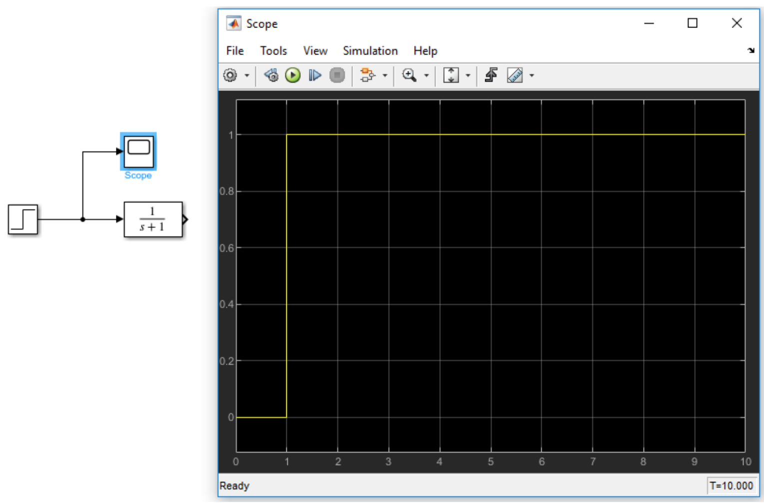
Otra vez se debe realizar una conexión. En este caso, no se observa una salida disponible para el bloque “Step” pero no hay ningún tipo de problema, porque se puede conectar el “Scope” de otra forma.



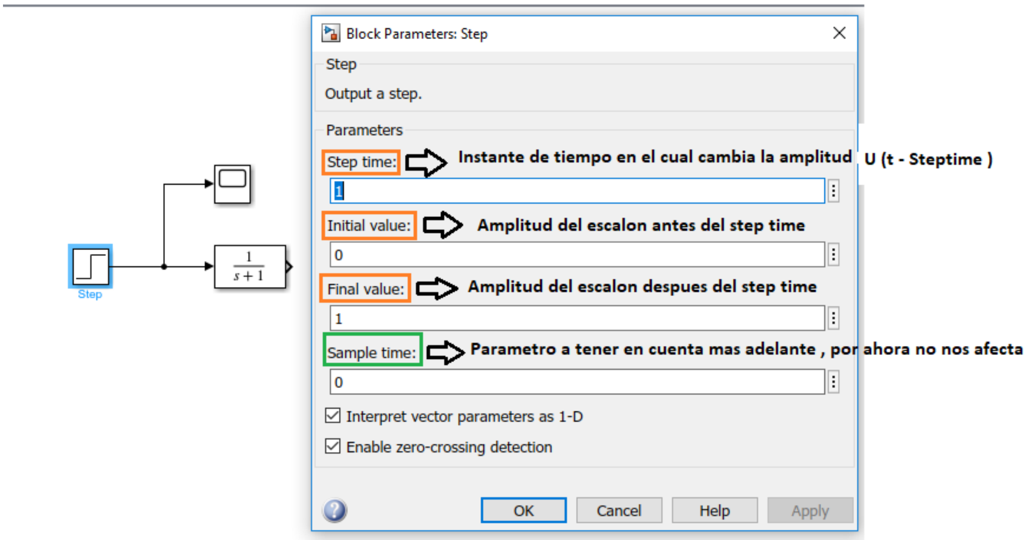
Una vez este todo listo, se va a ejecutar por primera vez la simulacion. Para ello hay que dirigirse al panel superior dentro de la hoja de trabajo e iniciar la simulacion.



Luego de que se termine la simulación, se deberá hacer doble clic sobre “Scope”

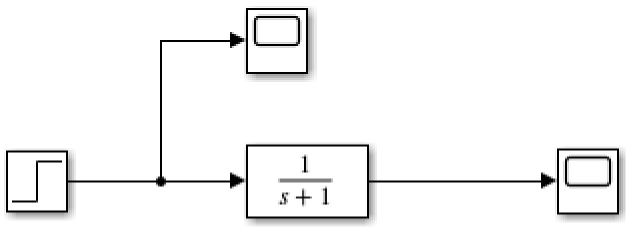


Por las características de la señal generada, se puede definir qué x(t) = 1U(t-1). Entonces, el siguiente paso será observar la configuración del bloque de “Step” para ver si esto es realmente correcto.

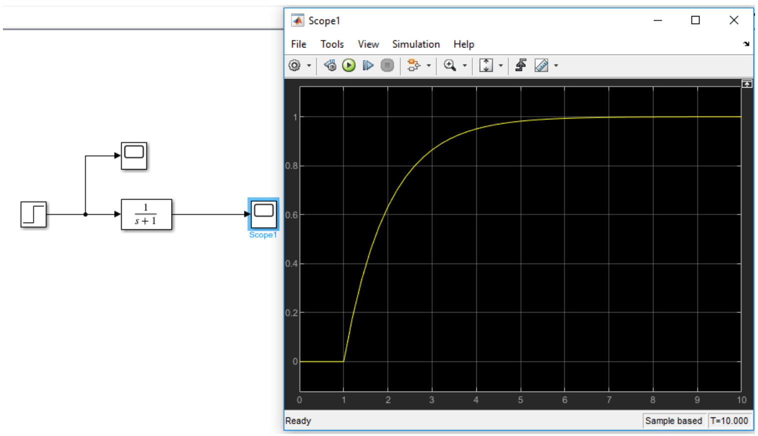


Se verifica que las configuraciones por defecto correspondían a un escalón de la forma: x(t) = 1.U(t-1).

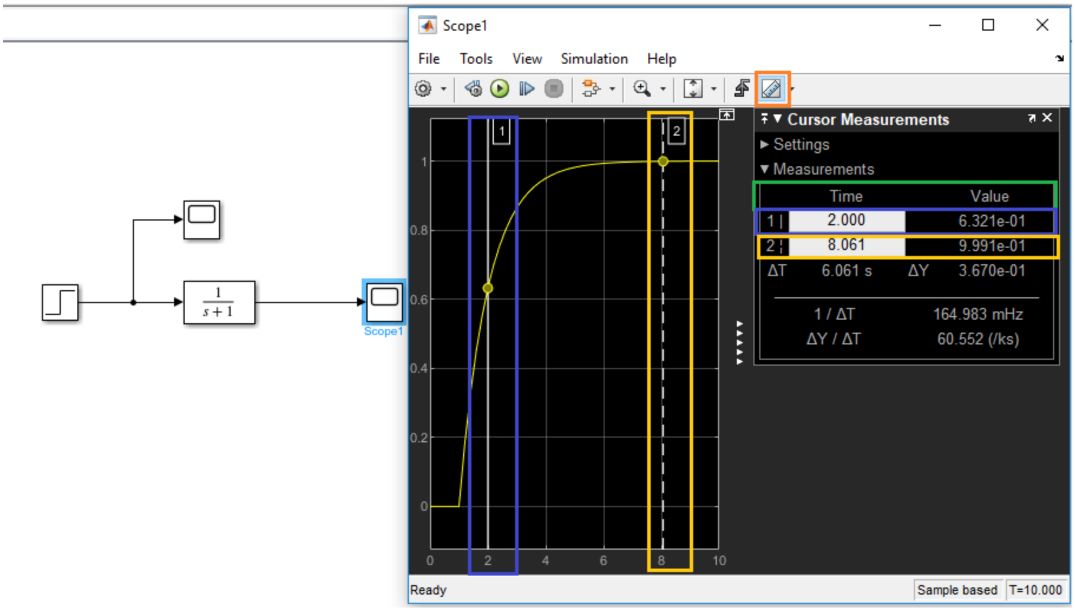
Por último, se agregará al proyecto otro “scope” que permitirá observar la salida del sistema:



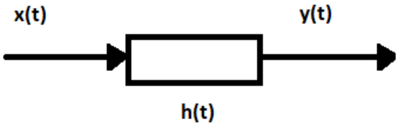
Al realizar otra simulación nuevamente y observar el “scope” de la salida.

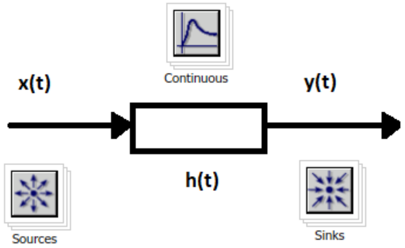


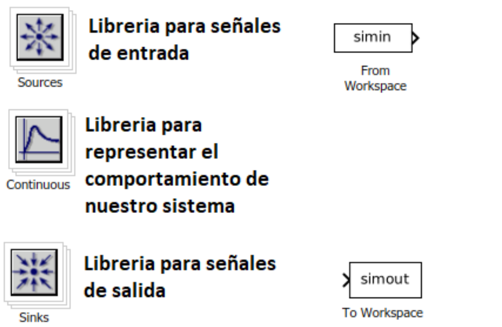
De acuerdo al conocimiento sobre los sistemas LTI de primer orden, se puede observar que la salida contiene un comportamiento característico correspondiente a una entrada del tipo escalón. Sin embargo, no se sabe si la salida está realmente respetando los valores de amplitud en los tiempos correspondientes. En "τ", la amplitud de la salida debe alcanzar un 63 % de su valor máximo. ¿Cómo se mide? aplicando una herramienta para medir dentro de las opciones de "Scope"



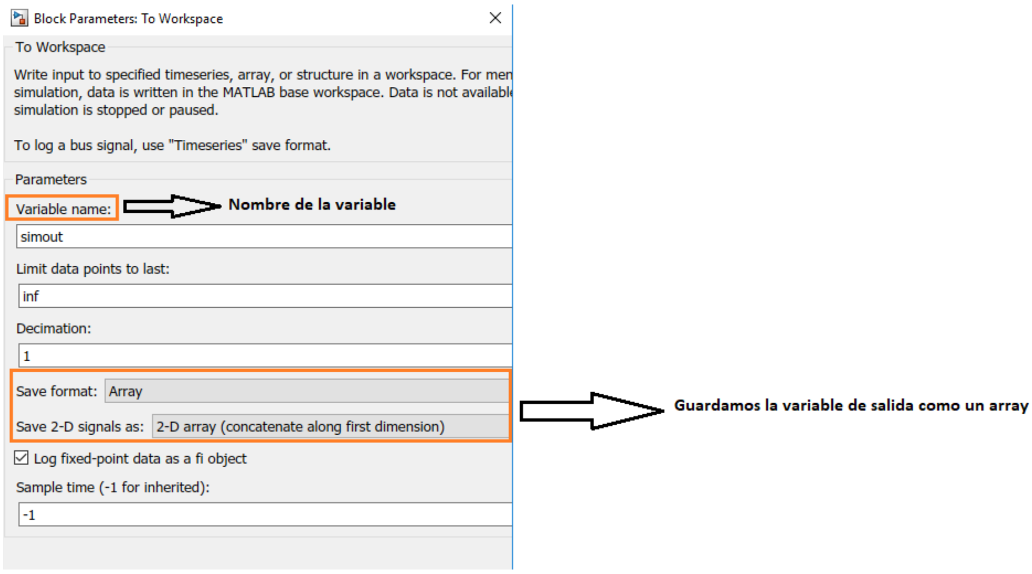
La simulación está funcionando bien y está logrando representar la salida de un sistema LTI de primer orden frente a una señal de entrada del tipo escalón. Como conclusión, si la simulación responde bien ante una entrada del tipo escalón, se puede probar con otras señales de entrada. Pero antes de entrar en detalle sobre eso, no hay ningún vector para manipular desde Matlab tanto de entrada como de la salida. Quedo todo en el ámbito de la simulación. ¿Tiene solución? Si, se puede generar variables de salida a Workspace en Matlab y además se puede importar variables para representar señales de entrada.



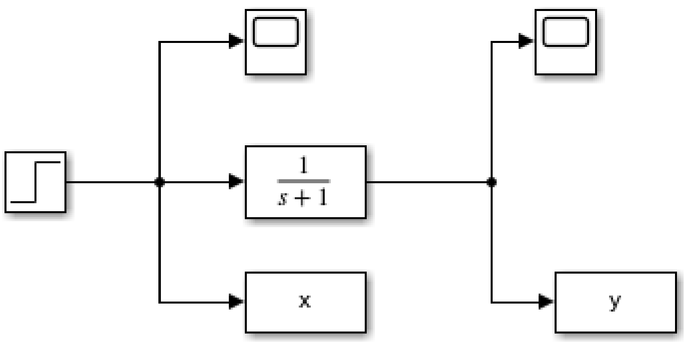




La configuración que necesita el bloque "To workspace" es:



De manera tal que se puede cambiar el sistema a simular de la siguiente forma:

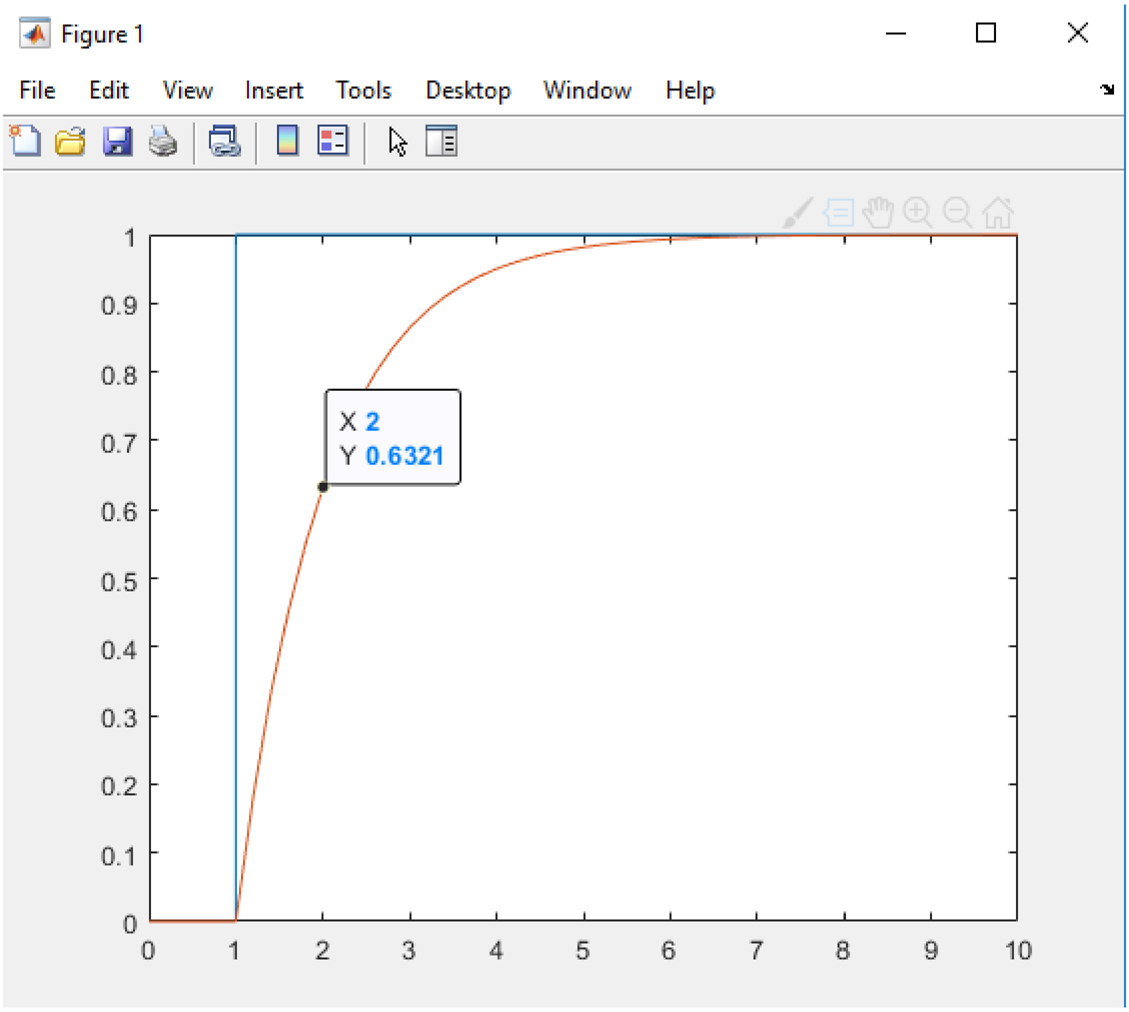


Al realizar la simulación se generan en workspace de Matlab 3 vectores. El primero representa el tiempo de simulación, el segundo es la señal de entrada x(t), que ahora está presente en workspace y el tercero es la señal resultante y(t). Entonces se podría graficar directamente desde la línea de comandos de Matlab.

Workspace

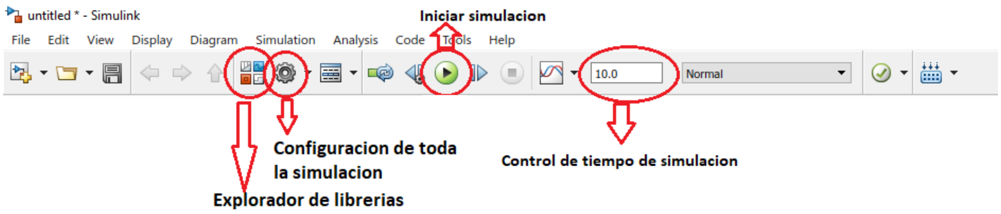
Name	Value
tout	53x1 double
x	53x1 double
y	53x1 double

```
>> plot(tout,x)
>> hold on
>> plot(tout,y)
>>
```

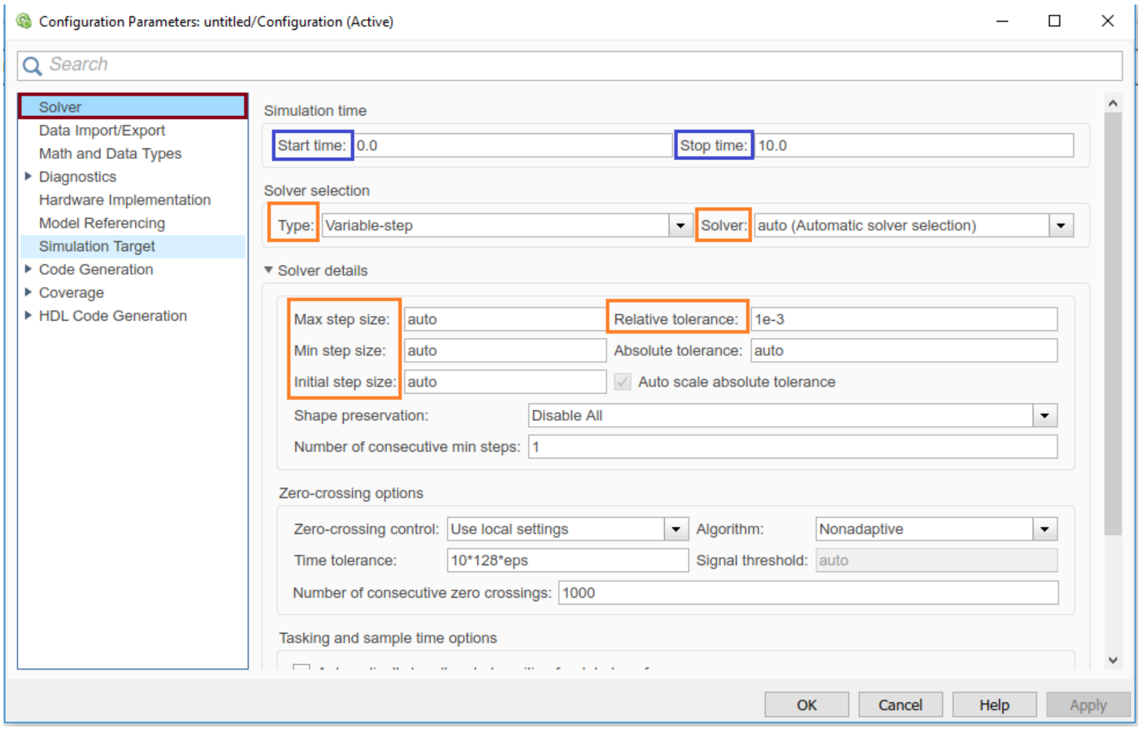


Al observar la figura , se encuentra que el tiempo de simulacion resulta ser muy grande. ¿Por que? Porque los instantes de tiempo mayores e iguales a 6 segundos no presentan ningun tipo de informacion. Las caracteristicas de los sistemas LTI establecen que a partir de 5 “T” la respuesta al escalon presenta variaciones muy pequeñas respecto al valor maximo y la respuesta al impulso es practicamente cero. Como el sistema simulado tiene un “T” igual a 1 seg y el escalon generado comienza en el instante de tiempo t=1 segundo, 5. “T” equivalen a 6 segundos, por lo tanto , para la simulacion resulta ser suficiente un intervalo de tiempo comprendido entre 0 y 6 segundos.

Para modificar estos parametros, se debe ingresar en la configuracion de la simulacion:



Una vez dentro, observaremos que las primeras dos casillas corresponden a especificaciones de inicio y fin de simulacion:



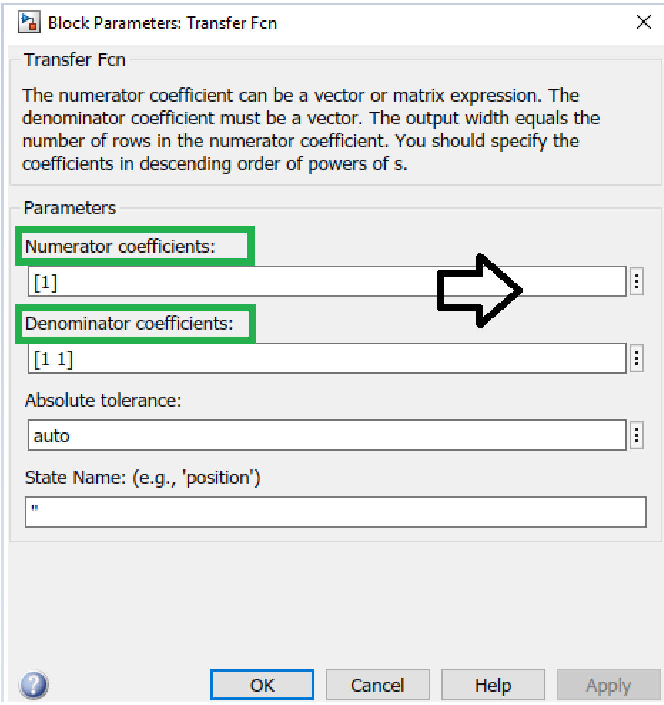
El segundo grupo de casillas esta relacionado a la manera en que simulink resuelve las operaciones matematicas en la simulacion. En el caso de la casilla del tipo “Type” se especifica el “paso” de la simulacion, es decir, si es variable o fijo y para la casilla “ Solver ”, el metodo numerico implementado.

¿Por qué estos parametros son importantes? Porque simulink si bien es una herramienta que permite simulacion de sistemas LTI en tiempo continuo, no deja de ser un sofware que trabaja con muestras discretas y mediante algoritmos matematicos logra muy buenas aproximaciones a los resultados de operaciones en tiempo continuos. Para que se entienda muy bien lo planteado, cuando se ejecuta el comando plot(tiempo, variable) lo que realiza es una interpolacion de las muestras discretas asociadas a una señal de informacion. La computadora no puede procesar intervalos de tiempo infinitesimales para construir realmente la señal en tiempo, la aproxima. Esa aproximacion es mejor o peor dependiendo la cantidad de muestras. Entonces es fundamental elegir bien dicho parametro para poder observar fenomenos en el dominio del tiempo de una manera adecuada.

En el caso de dejar la casilla del tipo “Type” en variable, deberemos especificar el paso minimo y el paso maximo que puede manejar simulink para cumplir con la tolerancia de error que se especifica en la casilla de al lado.

Sin embargo, no se entrara tan en detalle con estos parametros , por ahora, los dejaremos variables ya que como observamos anteriormente, las simulaciones responden de manera excelente a los resultados conceptuales esperados.

Por otro lado, ya se podria estar planteando , simular muchas variantes para un sistema LTI de primer orden. ¿Qué se necesita? En primer lugar, modificar la funcion de transferencia del sistema. Para ello, deberemos cambiar las configuraciones del bloque “*Transfer Fcn*”.



Ambos polinomios deben expresarse de forma completa con todos sus coeficientes desde el orden mas alto hasta el termino independiente

La construccion de la forma prototipo de los sistemas LTI de primer orden es:

$$H(s) = \frac{1/_{\tau}}{s + 1/_{\tau}}$$

Por lo tanto, al conformar los vectores que representan al polinomio del numerador y del denominador se tendra:

Num=[0 , $1/_{\tau}$] → 0.S + $1/_{\tau}$

Den= [1 , $1/_{\tau}$] → 1. S + $1/_{\tau}$

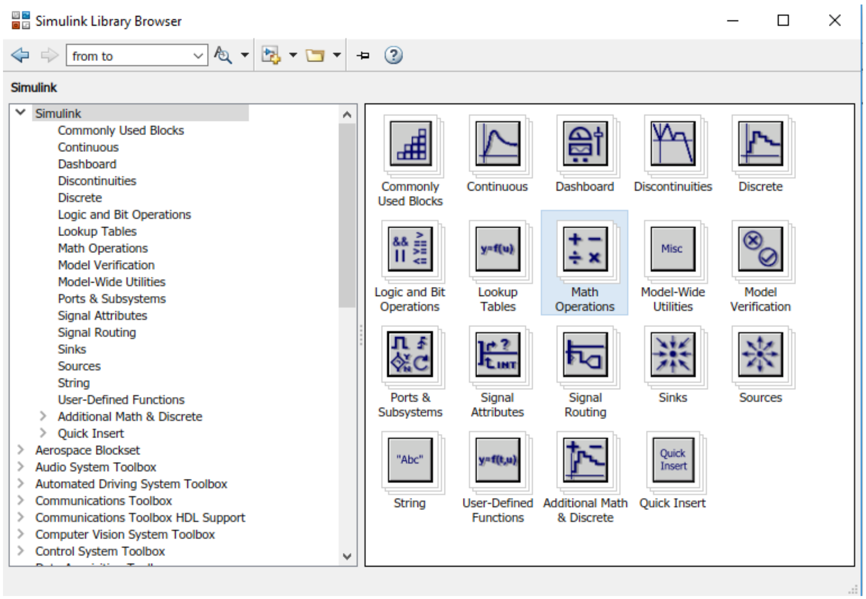
Modificando el valor de "τ" se obtendran diferentes comportamientos asociados al sistema de primer orden. ¿Qué ocurre si se quieren hacer muchas pruebas y de forma rapida cambiando los parametros del sistema? Existen dos opciones:

1- Modificar la configuracion de "*Transfer Fcn*" de acuerdo a los parametros de cada sistema a resolver y configurar el tiempo de simulacion.
2- Parametrizar la configuracion de "*Transfer Fcn*" en funcion de una variable que corresponda con el parametro "τ" y el tiempo de simulacion tambien. Asi desde la consola de matlab, se asigna el valor numerico a la variable con la cual se parametrizo y se inicia la simulacion.
   De simulacion a simulacion el unico cambio que se debe realizar es el valor numerico de la variable asignada.

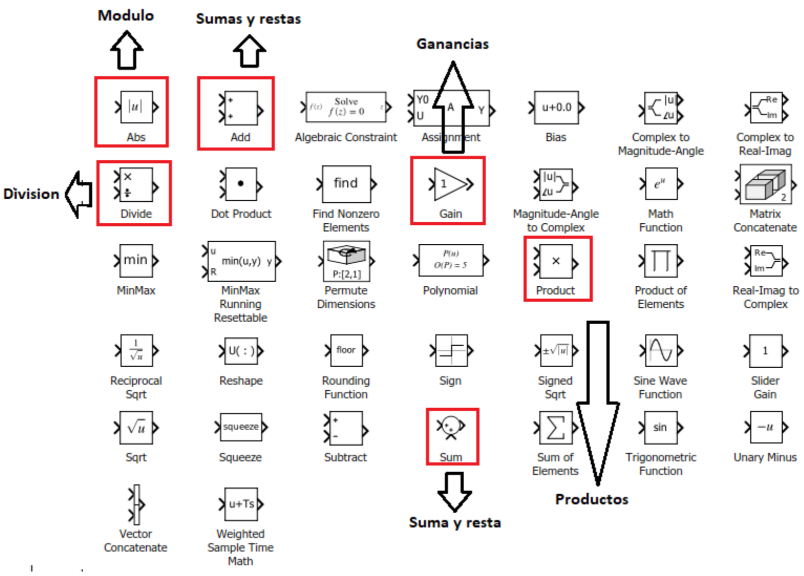
Ya se podrian simular multiples respuestas al escalon unitario para un sistemas LTI de primer orden. ¿Cómo se simula la respuesta al impulso? Se debe construir como señal de entrada una delta dirac.

Para crear esta señal se debe operar matematicamente con señales que ya estan definidas por bloques desde simulink o bien se podria exportar desde workspace. En el primer caso, se podria construir un impulso mediante la resta de dos escalones y luego aplicar una ganancia para que el area de la señal resultante sea igual a la unidad.

Para ello se debe ingresar en browser Library y optar por seleccionar operadores matematicos:

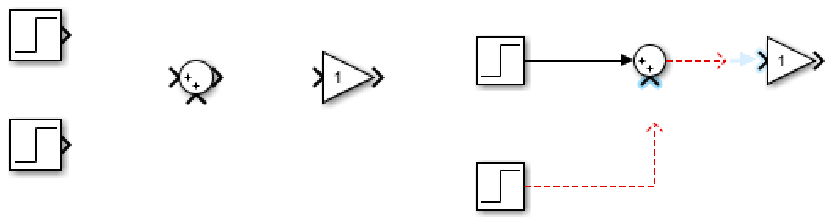


Se obtendran los siguiente bloques:

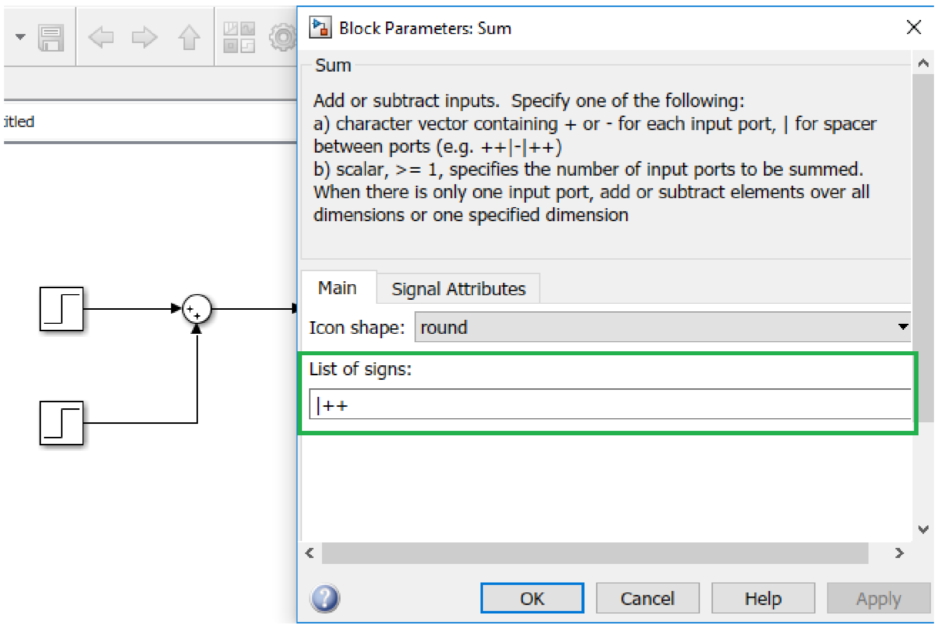


Como puede observarse existen muchas operaciones matematicas que pueden utilizarse. Sin embargo para este laboratorio se utilizaran los bloques : “Sum” y “Gain”.

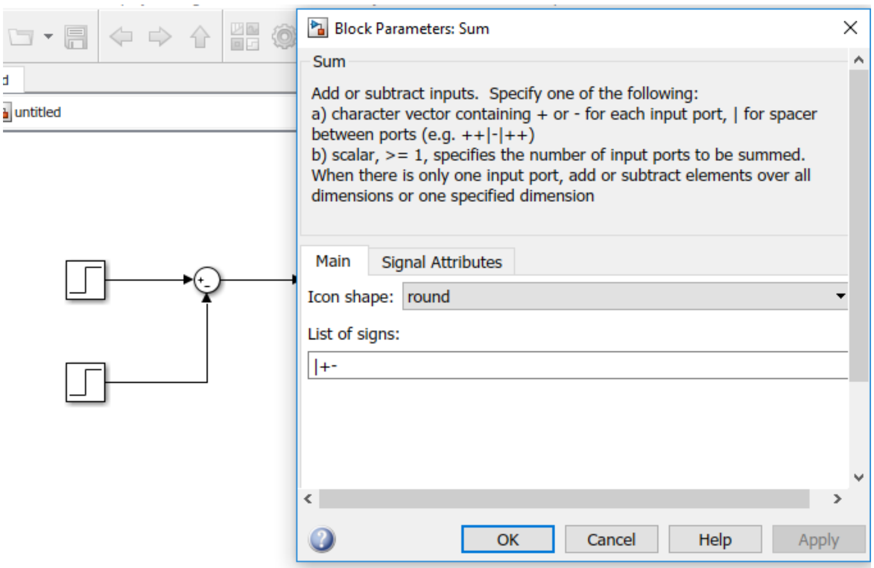
El procedimiento para generar un impulso es el siguiente:



Hay que generar una resta en lugar de una suma, por lo tanto se debera modificar las configuraciones del bloque “Sum”.

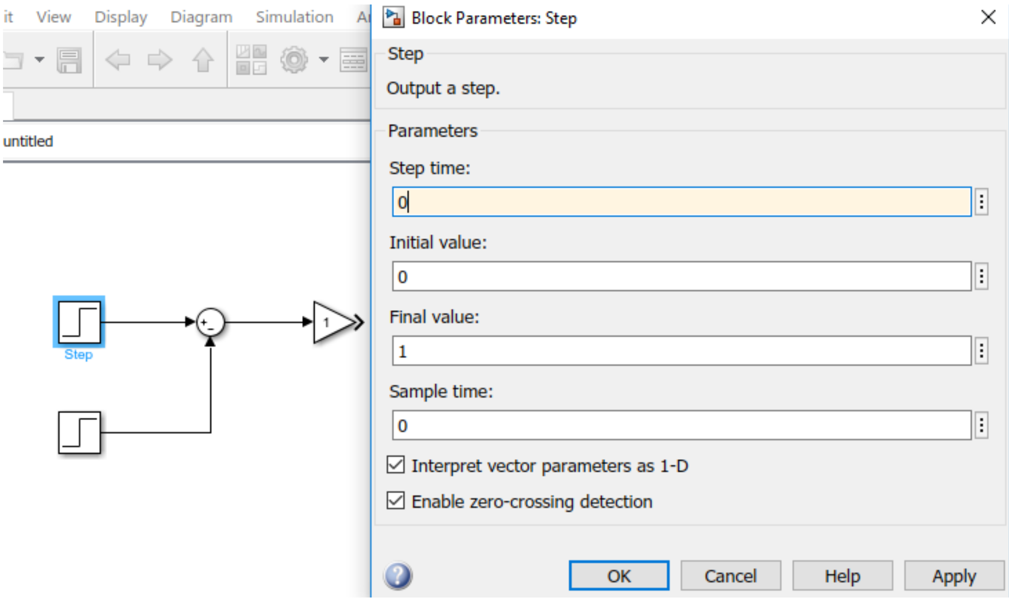


Como se puede observar en la figura anterior, al modificar la casilla “List of signs”, cambiando la configuracion por defecto de (| + +) se modifica la manera en la cual se tratan las entradas. Si la configuracion resultante termina siendo (| + -) la operación del bloque es una resta como se observa en la siguiente imagen:

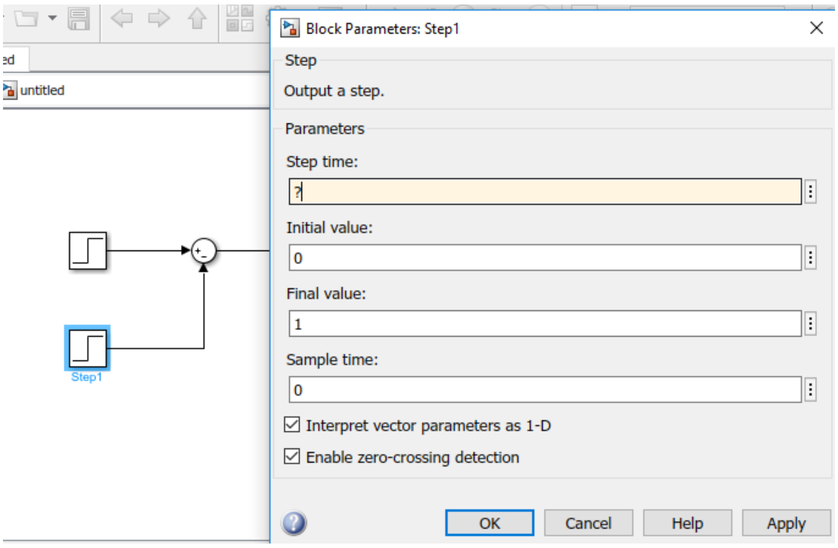


Tambien pueden agregarse mas entradas mediante la siguiente configuracion (| + + +) para lograr tres señales de entrada, si se desean mas, podria extenderse dicha configuracion hasta las n señales requeridas.

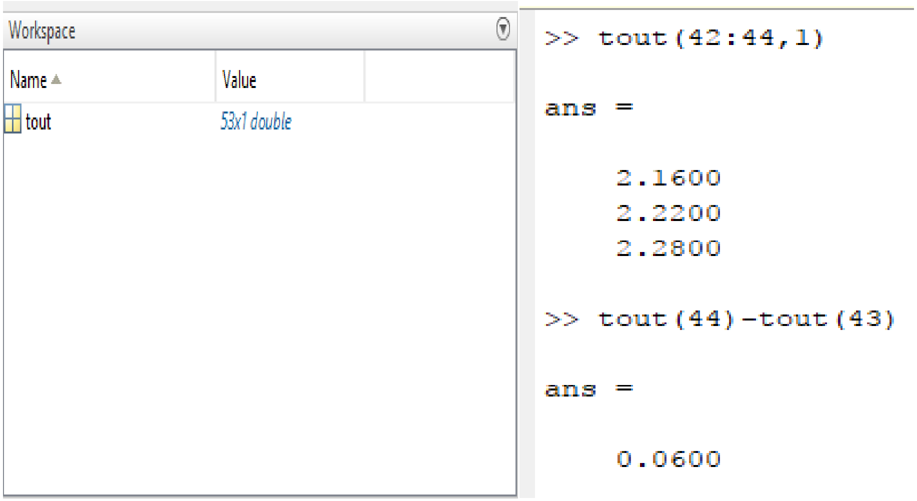
Luego se configura cada escalon unitario.



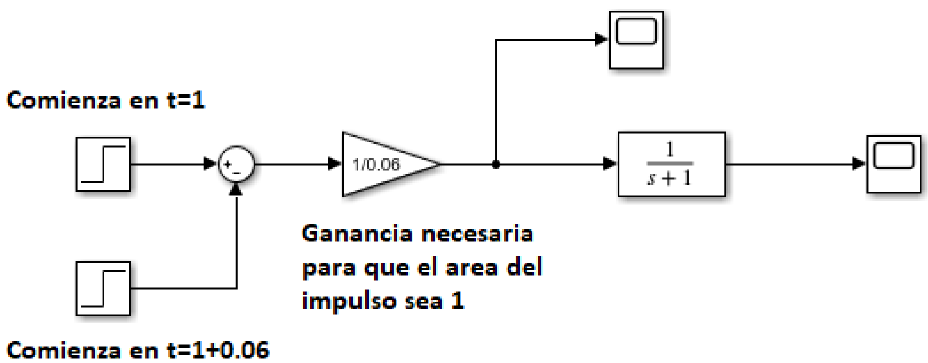
Para el escalon que debe realizar la resta ¿Qué configuracion se elige? Se sabe que el ancho de un impulso debe ser practicamente nulo, pero en la simulacion no se puede lograr dicha condicion. ¿Solucion? Averiguar el paso minimo de simulink para generar una señal cuyas caracteristicas desde el punto de vista de la simulacion sea un impulso, aunque practicamente se sabe que es una aproximacion.



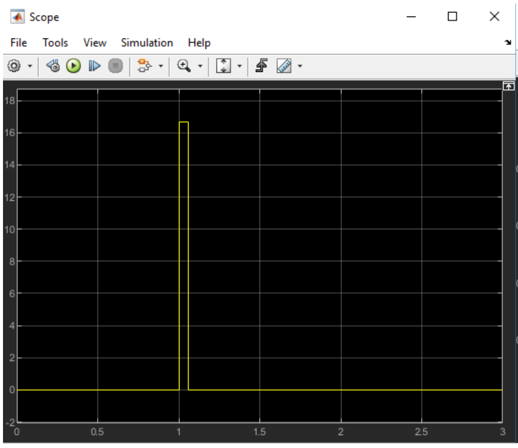
Se genera una simulacion sencilla para obtener el vector tiempo “tout” en workspace y luego se opera de la siguiente manera:

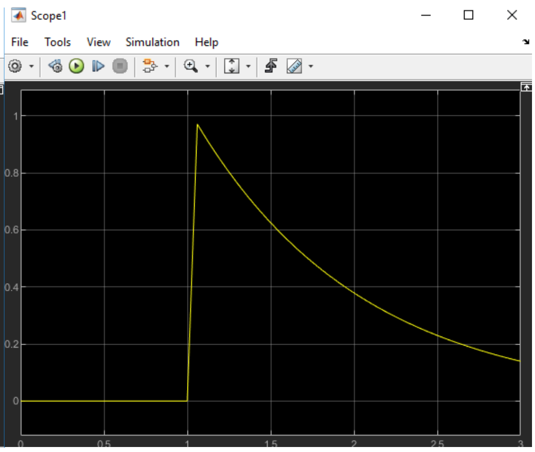


Se ha encontrado el paso minimo de la simulacion que sera asociado al ancho de la señal generada mediante la resta de los escalones. Se aplica la configuracion y se obtiene los siguientes resultados.



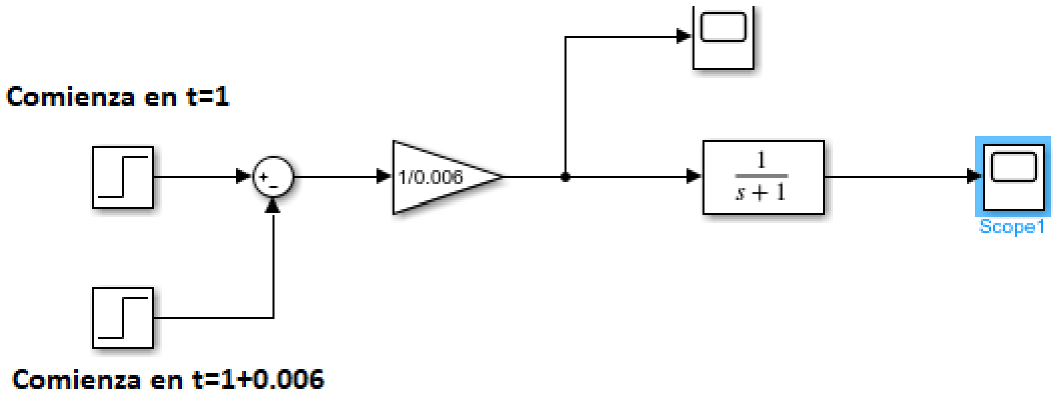
Al observar la entrada **x(t)** y la salida **y(t)** , se encuentra que la aproximacion no cumple con las espectativas. Es decir, al utilizar los tiempos de simulacion proporcionados por simulink el sistema tiene una representacion bastante mala. La amplitud de la respuesta al impulso para t=0 deberia alcanzar el valor de $^1/_\tau$ y escalar de forma vertical, sin embargo se observa una pequeña pendiente.

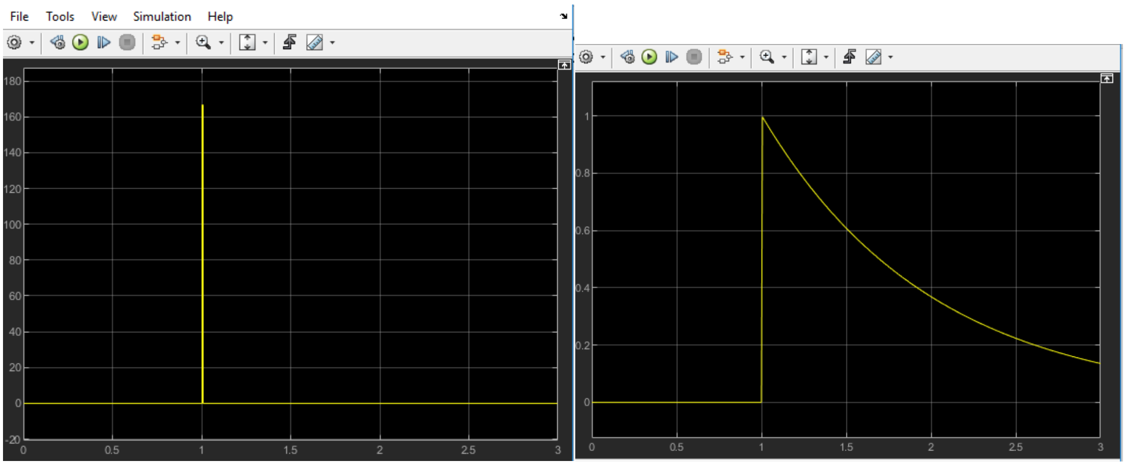




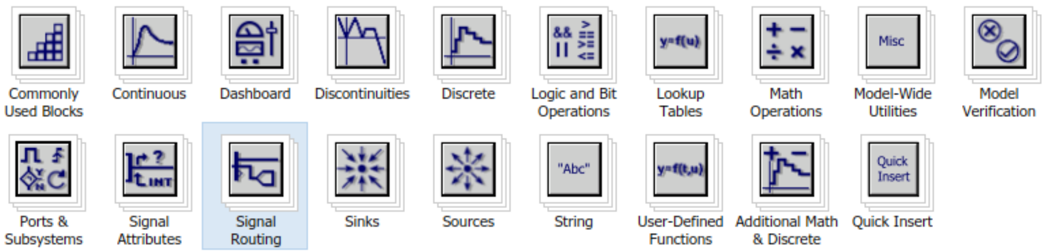
Como el paso esta configurado automaticamente para que simulink detecte la cantidad de muestras necesarias para representar correctamente las señales, forzaremos a que la simulacion aumente la cantidad de muestras y por lo tanto reduzca el paso, para tratar de generar una buena aproximacion al impulso.

Se considerara, que el ancho del pulso que representa al impulso es de 0.006 , por lo tanto , la ganancia deberia ser de $^1/_{0.006}$.

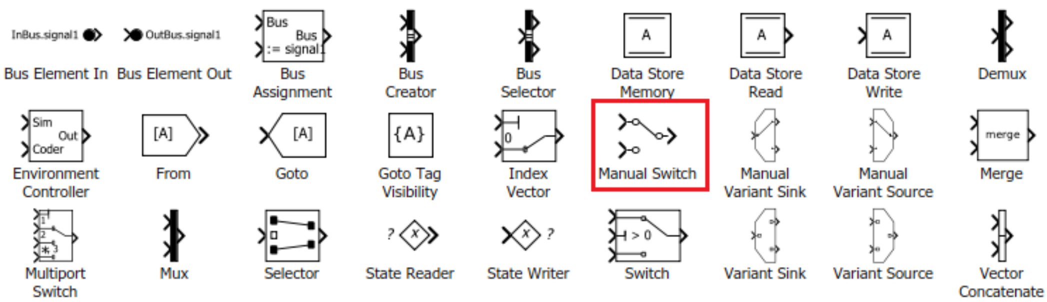




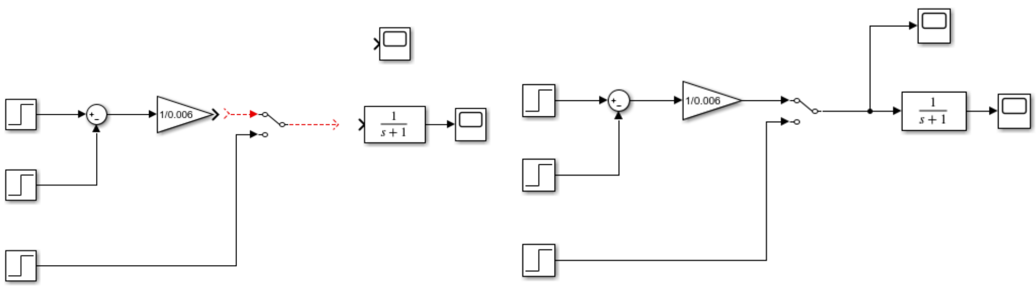
¿Que ocurre si se desea aplicar mas de una señal de entrada al sistema de forma simple y sin tener que duplicar la cantidad de bloques? Se podria agregar un “Switch” que se encuentra dentro de la librería “signal Routing”. Este bloque permite conectar dos entradas y conmutar para seleccionar la salida.



Dentro de la librería encontramos:



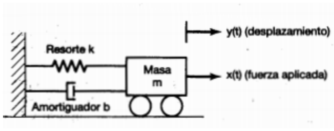
Al aplicarlo al sistema obtenemos el siguiente esquema:



De esta manera se tiene la simulación preparada para dos señales de entrada, el impulso y el escalón. Lo mismo podría aplicarse para múltiples señales de entrada.

Sistemas LTI Continuo de segundo orden

En el caso de armar sistemas LTI de segundo orden, se debe recordar las siguientes características:



$$\frac{d^2y(t)}{dt^2} + \left(\frac{b}{m}\right)\frac{dy(t)}{dt} + \left(\frac{k}{m}\right)y(t) = \frac{1}{m}x(t).$$

$$\frac{d^2y(t)}{dt^2} + 2\zeta\omega_n\frac{dy(t)}{dt} + \omega_n^2 y(t) = \omega_n^2 x(t).$$

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

$\omega_n = \sqrt{\frac{k}{m}}$ Frecuencia natural NO amortiguada, ω_n

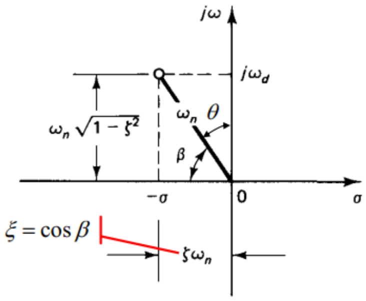
$\zeta = \frac{b}{2\sqrt{km}}$, Relación de amortiguamiento, ξ

$$s_1, s_2 = -\xi\omega_n \pm j\omega_n\sqrt{1-\xi^2} = -\sigma \pm j\omega_d$$

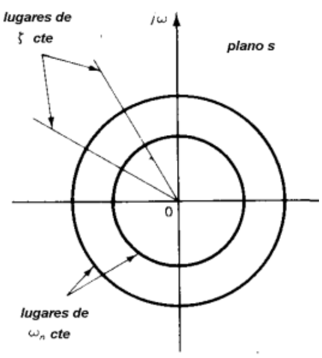
Constante de tiempo inversa

Frecuencia natural amortiguada

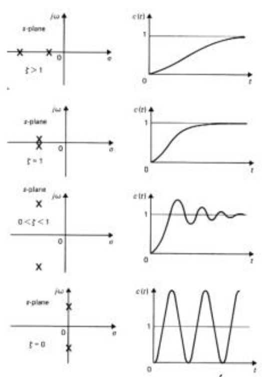
Interpretación geométrica de los parámetros de la F.T.



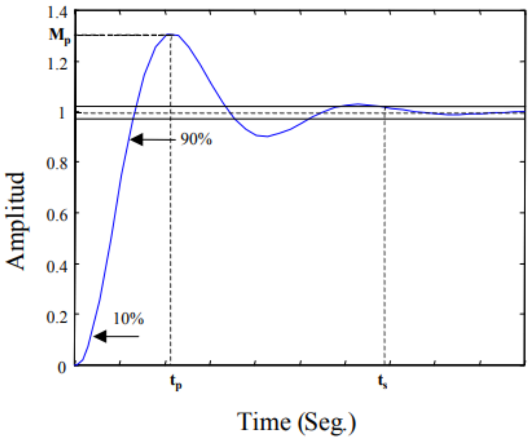
En base a la relación de amortiguamiento ξ constante y la frecuencia natural no amortiguada, ω_n, constante se establecen los correspondientes lugares geométricos en el plano s.



- $\xi > 1$ (s_1 y s_2 reales distintos, parte real -) →SOBREAMORTIGUADO
- $\xi = 1$ (s_1 y s_2 reales iguales, parte real -) →CRÍTICAMENTE AMORTIGUADO
- $0 < \xi < 1$ (s_1 y s_2 conj. complejos, parte real -) →SUBAMORTIGUADO
- $\xi = 0$ (s_1 y s_2 sobre el eje imaginario)



Para el caso de sistemas subamortiguados



El porcentaje de sobrepaso (overshoot) es una medida del valor de sobrepaso que tiene la respuesta del sistema por sobre la amplitud de la entrada escalón.

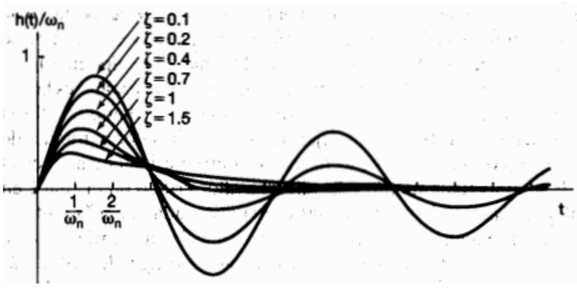
$$M_p = 1 + e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}}$$ (Sobrepaso en tanto por uno)

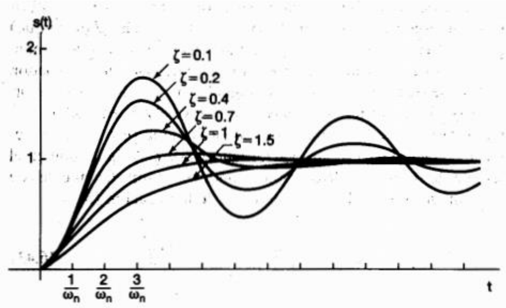
$$P.O = 100 e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}}$$ (Porcentaje de sobrepaso o overshoot)

Tabla I. Porcentaje de sobrepaso vs. coeficiente de amortiguamiento

ζ	0.9	0.8	0.7	0.6	0.5	0.4	0.3
P.O	0.2	1.5	4.6	9.5	16.3	25.4	37.2

Las respuestas al impulso y al escalón caracterizadas por el sobrepaso:





El tiempo de respuesta máxima (time to peak). Este es el tiempo en que se produce la máxima amplitud de salida.

$$t_p = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}}$$

El tiempo de subida (rise time o T_r) es el tiempo que toma la respuesta para subir desde el 10% al 90% de la amplitud del escalón de entrada. El tiempo de subida para un sistema de segundo orden es aproximado por la siguiente expresión.

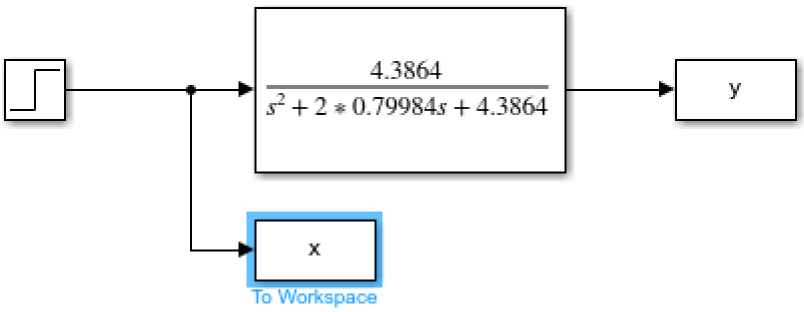
$$T_r = \frac{2.16\zeta + 0.6}{\omega_n} \qquad 0.3 \le \zeta \le 0.8$$

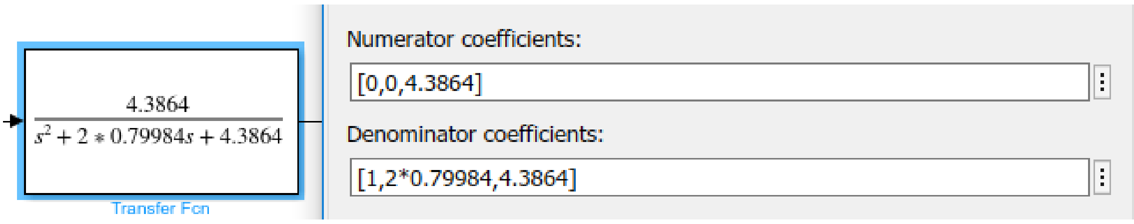
Se creará un sistema masa resorte cuya frecuencia natural No amortiguada corresponda a una señal oscilante con periodo 3 seg. Por lo tanto, a partir de la siguiente ecuación $Tsp = \frac{2\pi}{\omega_n}$ se obtiene que "ω_n" es $\frac{2\pi}{3}$ rad/seg y se desea que el tiempo de asentamiento sean aproximadamente 5 segundos, por lo tanto mediante la siguiente ecuación $Ts = \frac{4}{\xi\omega_n}$, se encuentra que "$\xi$" es 0,3819.

$$H(s) = \frac{4.3864}{s^2 + 2 * 0.79984\ s + 4.3864}$$

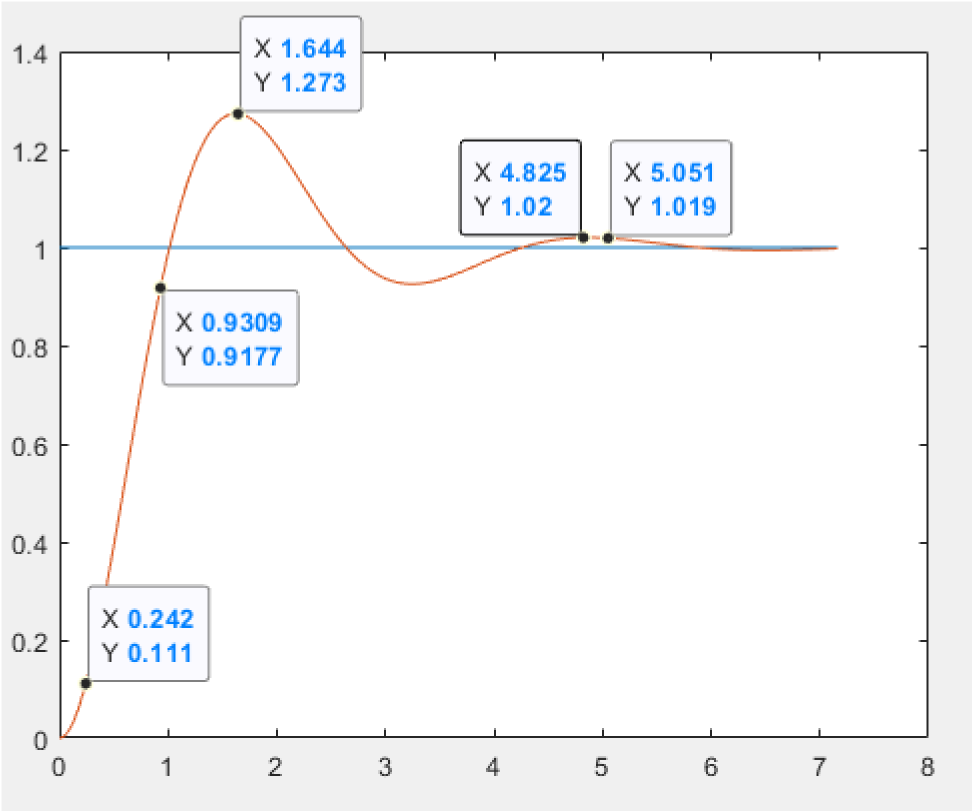
$$\xi\ \omega_n = 0.79984;\ {\omega_n}^2 = 4.3864$$

En simulink se genera el siguiente esquema:

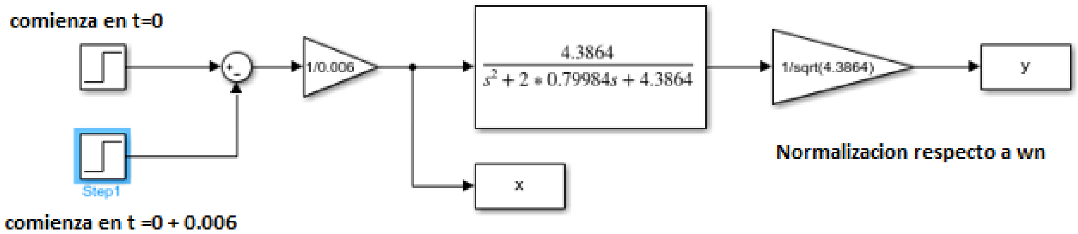


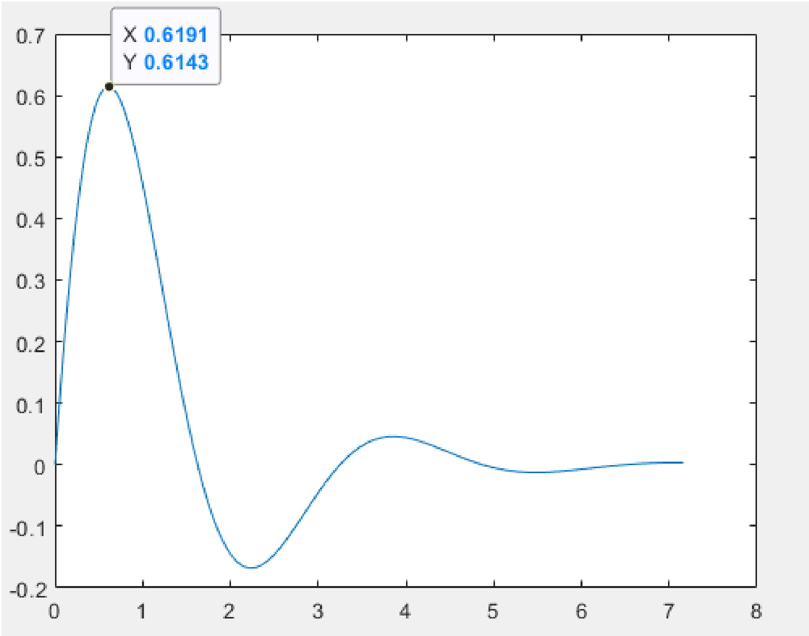


Al comparar **x(t)** con **y(t),** el sistema LTI de segundo orden responde de la siguiente forma**:**

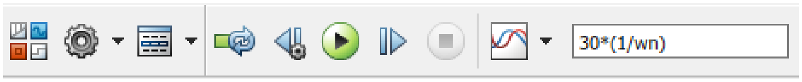


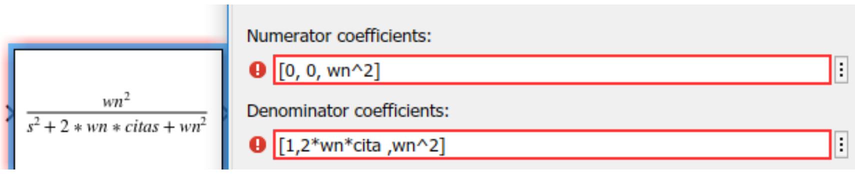
Se observa que la salida responde a todos los conceptos teóricos esperados con respecto a un sistema LTI de segundo orden. Ahora se aplicará como señal de entrada un impulso:



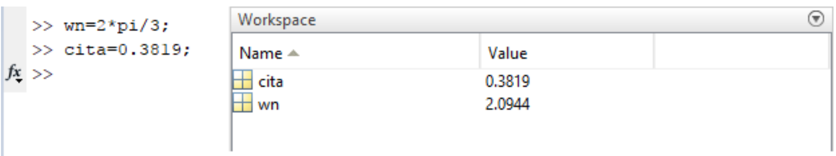


Se podría llevar a cabo un montón de simulaciones de una manera mucho más sencilla si se lleva a cabo una parametrización del sistema respecto a "ξ " y "ω_n".

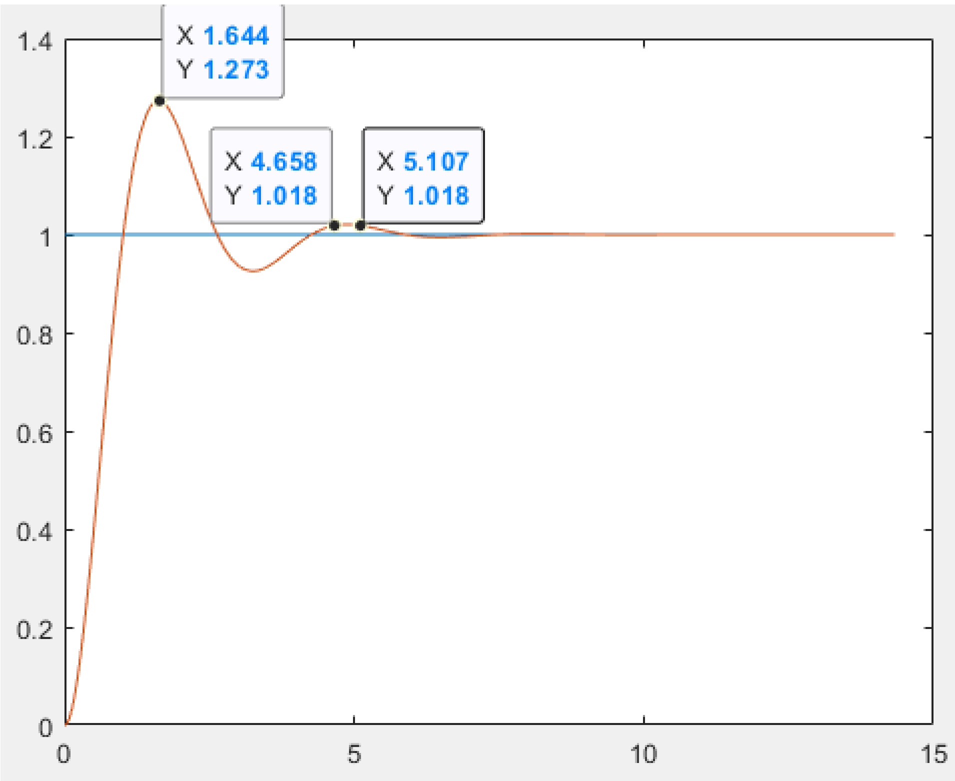




Simulink alerta al usuario que no reconoce dichas variables. Desde "workspace" se establecen los valores correspondientes a las variables.



Al comparar **x(t)** con **y(t),** el sistema LTI de segundo orden parametrizado responde de la siguiente forma**:**



Solo es necesario cambiar dos variables desde la consola de comandos para probar cada una de las alternativas.

Sistema LTI discreto de primer orden

Por último, se realizará una simulación para sistemas LTI discretos de primer orden.

$$y[n] - ay[n-1] = x[n],$$

Aplicando transformada zeta

$$y(z) - a.z^{-1}y(z) = x(z)$$

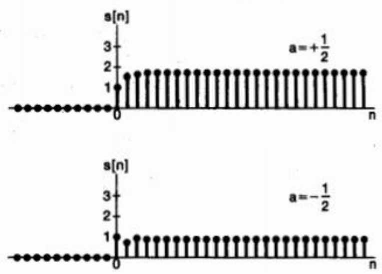
$$H(z) = \frac{y(z)}{x(z)} \Rightarrow \boxed{H(z) = \frac{1}{1 - a.z^{-1}}}$$

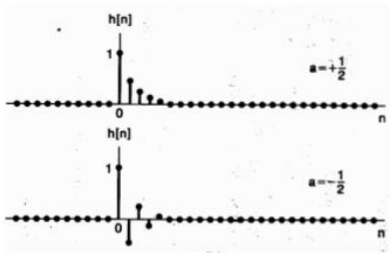
la respuesta al escalón del sistema es

$$s[n] = h[n] * u[n] = \frac{1 - a^{n+1}}{1 - a}u[n],$$

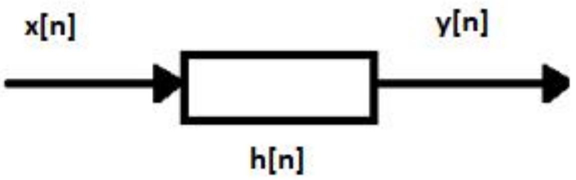
respuesta al impulso

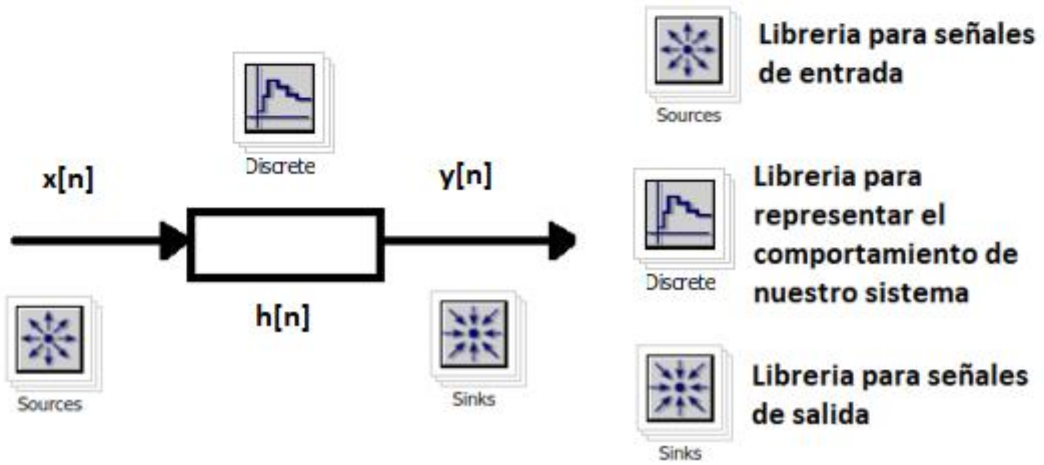
$$h[n] = a^n u[n],$$





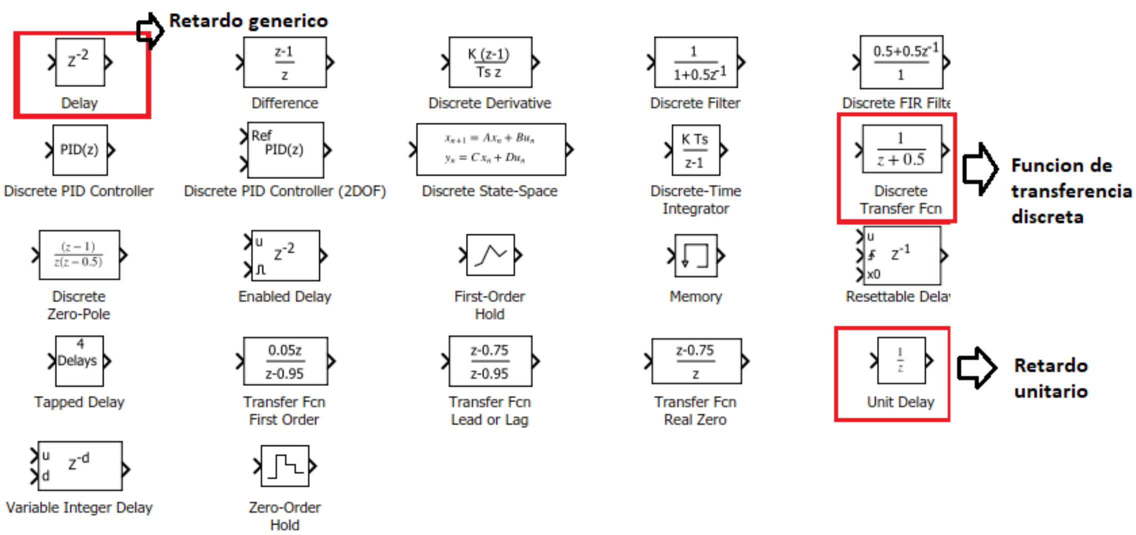
En este caso, se planteará el siguiente escenario:





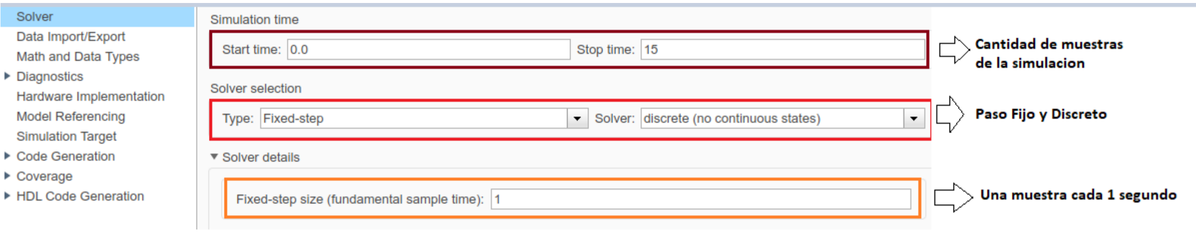
Como se puede apreciar en la imagen, simulink solo utiliza una librería para representar las señales, independientemente si se trata de sistemas continuos o sistemas discretos. En cambio, para representar las respuestas impulsivas de los sistemas, existen dos librerías distintas. "Discrete" para sistemas discretos y "Continuous" para la parte continua.

Al inspeccionar la librería "Discrete" se encuentra:

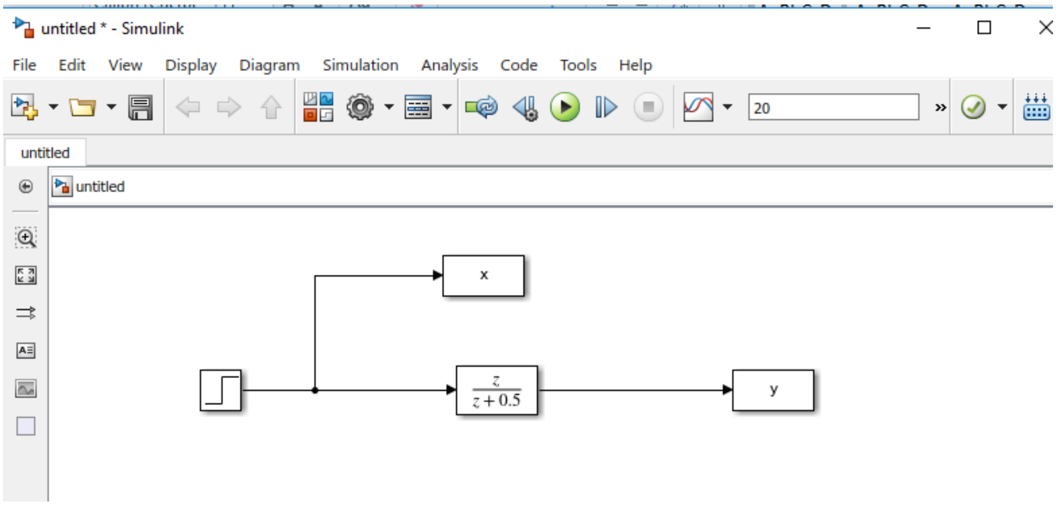


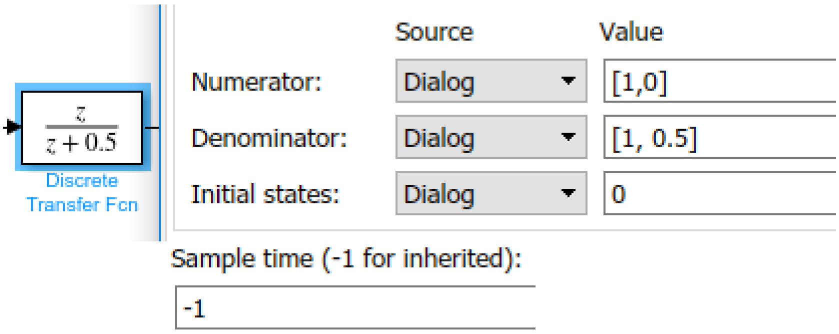
Hay que aclarar que, para trabajar con sistemas discretos además de utilizar las librerías correspondientes, se deben realizar configuraciones particulares. Las mismas están netamente relacionadas con el paso que debe utilizar la simulación, el cual, será discreto.

En la pestaña de configuración se establecerán los siguientes parámetros:

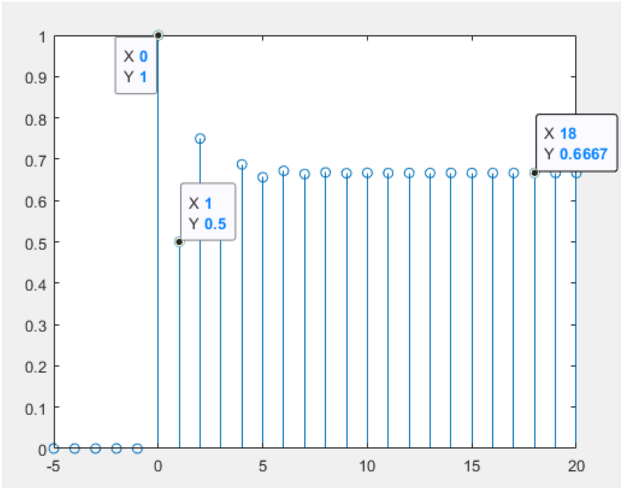


En la hoja de trabajo se arma el siguiente esquema utilizando los bloques básicos “*Step*”, “*Discrete Transfer Fcn*” y “*To workspace*”:

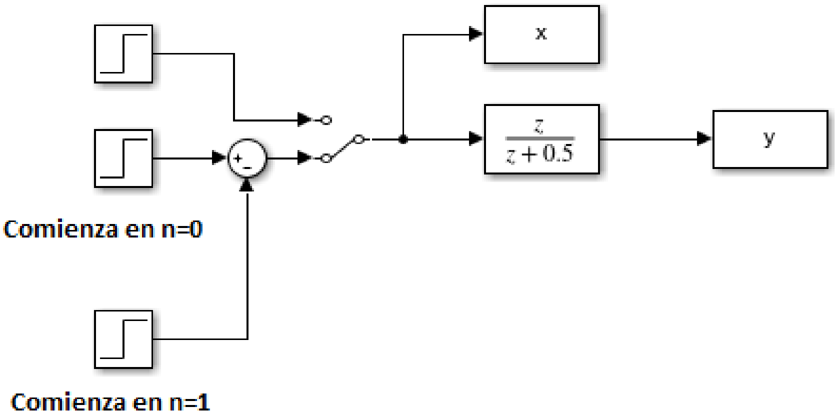




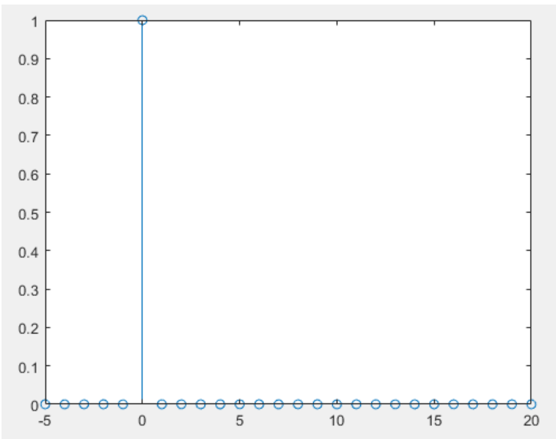
Se obtienen y grafican los resultados de la simulación cuando la señal de entrada resulta un escalón unitario:

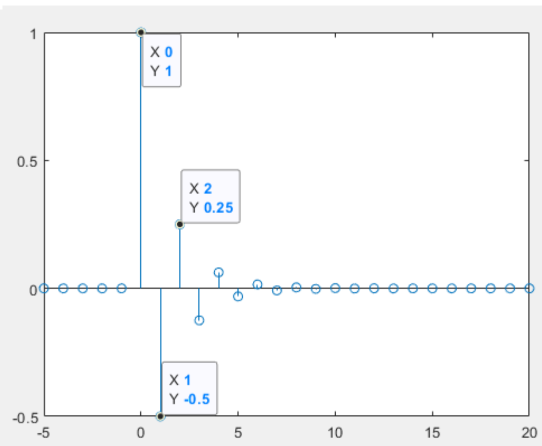


Para obtener la respuesta impulsiva se plantea el siguiente esquema:



Se verifican las variables “x” (entrada) e “y” (salida) del sistema:





Al igual que el resto de los sistemas presentados anteriormente, el sistema LTI discreto de primer orden puede parametrizarse y hacer diversas simulaciones.