Grado: Ingeniería Electrónica de Comunicaciones
Asignatura: Control de Sistemas
Profesores: Victor Manuel Maroto
Eva Besada Portas
Curso: 2020/21

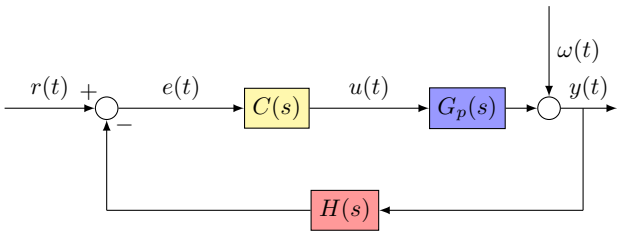
Práctica 1
Modelado y Simulación con Simulink

Objetivos

El objetivo de esta práctica es aprender a utilizar Simulink como herramienta de modelado y simulación de sistemas realimentados. Para alcanzarlo, el alumno realizará las tareas que se enumeran a continuación y que comprenden: 1) la definición de los modelos de diferentes sistemas en Simulink, 2) la simulación/visualización de los resultados desde Simulink y desde la linea de comandos de Matla, y 3) la inclusión de elementos no lineales en el modelo que le dan un comportamiento realista al sistema.

Sistemas en Modelos de Entrada y Salida

El primer sistema que se desea modelar en Simulink en esta práctica se encuentran representado mediante el siguiente esquema y definido a través de las siguientes funciones de transferencia:

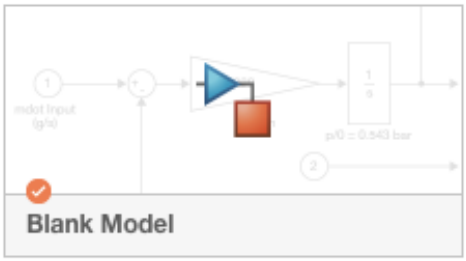


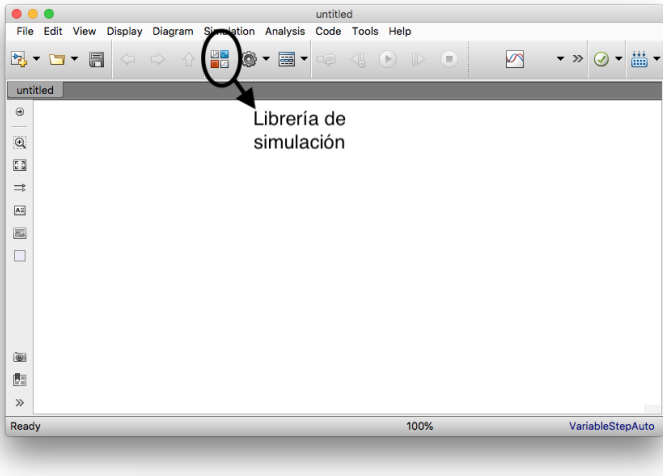
- Planta[1]: $G_p(s) = \frac{1}{s+100}$
- Controlador:
  - Proporcional: $C(s) = K$
  - Integral: $C(s) = \frac{K}{s}$
- Sensor: $H(s) = 1$

[1] Nótese que aunque en las imágenes capturadas aparecen diferentes plantas, el alumno debe resolver la práctica con la $G_p(s)$ propuesta.

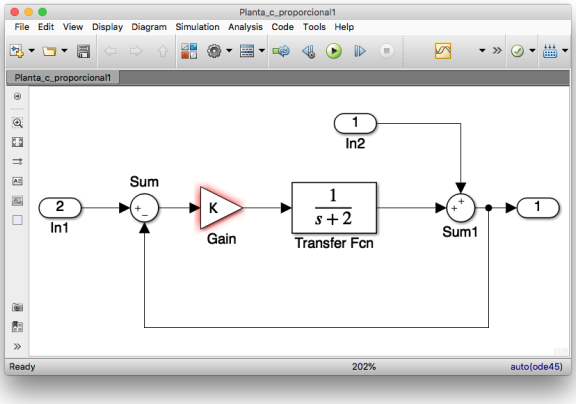
Para modelarlo, primero se abrirá Simulink (a través del icono 

que aparece en el IDE de Matlab) y se creará un nuevo modelo vacio (blank model) Simulink (a través del menu de modelos del navegador de la Librería de Simulación).





A continuación, se definirá el modelo deseado, arrastrando del Navegador los bloques necesarios (en este caso: funciones de transferencia, ganancia, suma, entradas y salida genéricas -bloques in/out-). Acto seguido, se editará la información de cada bloque (haciendo doble click sobre el mismo para acceder a los parámetros modificables) y se realizarán las conexiones necesarias. Finalmente, se grabará el modelo Simulink (semejante al que se muestra en la figura de la derecha para el caso de un controlador proporcional de valor K).



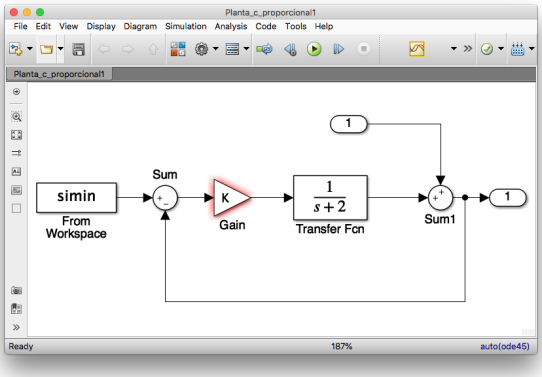
Con el modelo así definido, se puede obtener la matriz de transferencia entre las entradas y la salida con el siguiente programa, en el que en la primera linea se dan valores a las constantes del modelo, en la segunda se solicita la representación en variables de estado de una linealización del modelo (i.e. si el modelo no fuese lineal, la orden lo linealizaria) y en la tercera se hace la transformación de representación (primero se define el modelo en la representación en variables de estado y a continuación se transforma en su representación en matriz de transferencia).

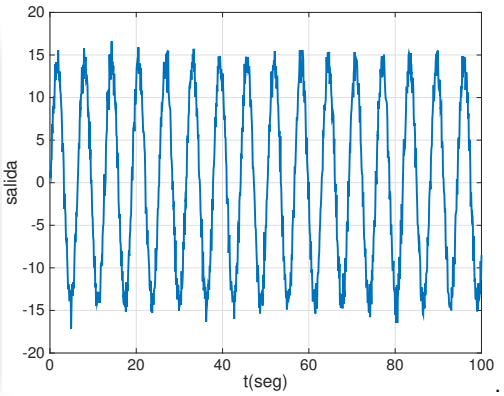
```
K=5;  %Poner el valor que se desea utilizar para resolver el problema
[A,B,C,D]=linmod('Modelo1');  %dlinmod('Modelo',T) para modelos discretos/mixtos
gs=tf(ss(A,B,C,D))            %ss(A,B,C,D,T) para modelos discretos
```

Una vez que se ha definido el modelo Simulink, se puede simular de dos formas diferentes:

- **Desde la linea de comandos.** Para realizar este tipo de simulación, util cuando se desea incluir los resultados de la simulación en un programa que está realizando otras labores (por ejemplo, comprobando

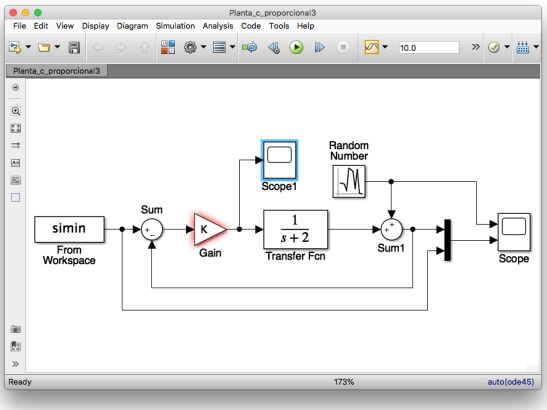
el funcionamiento del modelo ante diferentes valores de ciertos parámetros), es necesario modificar el **Modelo** para incluir las señales de entrada. Por ejemplo, en el caso de la siguiente figura (izquierda) se ha incluido un bloque que permite leer la referencia de la linea de comandos y otro que introduce en la perturbación en la salida una señal de ruido normal. A continuación, se puede escribir un programa como el siguiente, que en la orden sim recibe el nombre del modelo Simulink que desea simular y las propiedades de la simulación (tiempo, configuración) y devuelve los resultado de la simulación (estados y salida). Además, la orden simset permite indicar que las variables que debe elegir el modelo (K, input) deben ser cargadas desde el espacio de trabajo de este script. En este caso, el resultado de la simulación, que se representa en la siguiente figura (derecha) es dibujado por la última linea del script.



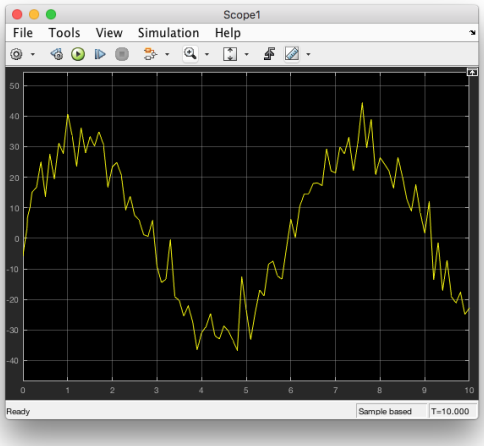


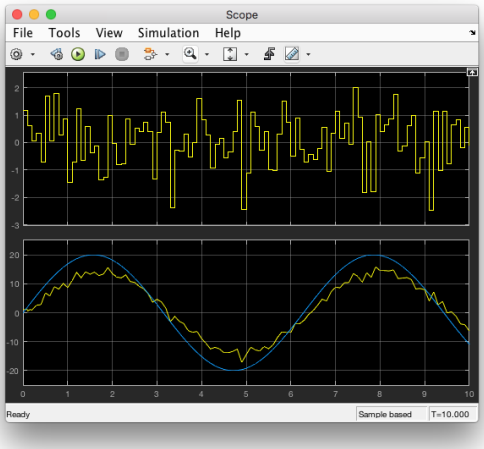

```
K=5;  %Poner el valor que se desea utilizar para resolver el problema
t=[0:0.1:100]';
r=20*sin(t);
simin=[t,r];  %simin es el nombre de la variable que se carga
simopt=simset('SrcWorkspace','current');  %Que coja las variables de aqui
[t,x,y]=sim('Planta_c_proporcional2',100,simopt);
plot(t,y),xlabel('t(seg)'),ylabel('salida');
```

- **Desde Simulink.** Para realizar y visualizar la simulación desde Simulink, además de introducir las señales de entrada como en el caso anterior, es necesario conectar algún elemento de visualización (por ejemplo un Scope) a las señales que se van a visualizar. Además, es posible colocar varias curvas en la misma gráfica (utilizando un Multiplexor) o poner varias gráficas en la misma figura. Como ejemplo, en figura de la derecha se ha añadido un Scope para visualizar la señal de control, y otro para visualizar en dos gráficas, por una parte la señal de referencia y la salida, y por otra la perturbación en la salida.



Con el modelo así construido, la simulación se arranca dandole al botón  y la duración se fija también desde el modelo. Los resultados de la simulación se pueden observar en las ventanas que se abren haciendo doble click sobre los visualizadores. Las dos figuras siguientes muestran, a la izquierda la información recogida en el Scope1 (señal de control) y a la derecha en Scope (perturbación y, señal de referencia y de salida). En el caso en el que se superpogan varias gráficas, es conveniente fijarse en el orden de colores (amarillo, fuxia, ...)

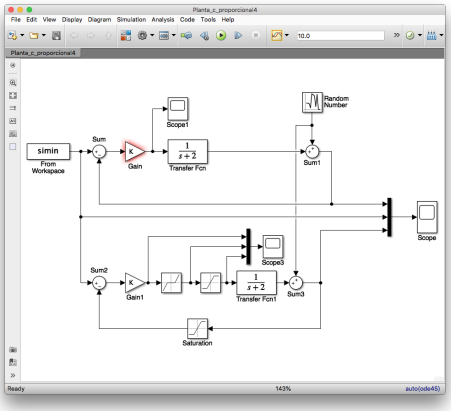




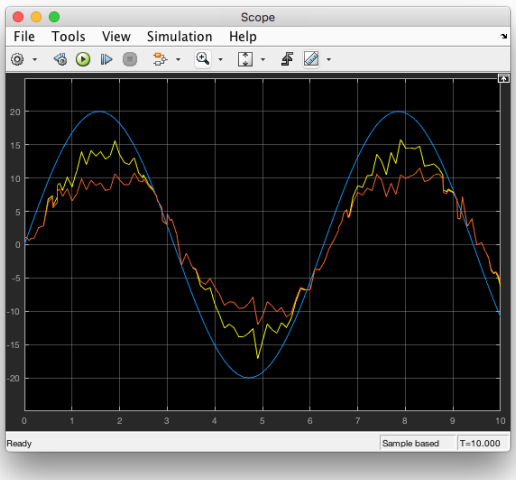
Los iconos que aparecen en las ventanas asociadas a los visualizadores (Scopes) permiten realizar diferentes labores. Destaca el icono de configuración  , que permite elegir cuantas graficas/puertas de entrada tendrá el visualizador (por ejemplo para el Scope se ha cambiado el valor de 1 a 2 para tener dos gráficas), el número de datos consecutivos que se muestran y opcionalmente el nombre de la variable en la que se alamacenarán. También suele ser necesario usar el icono de autoescalado  de los ejes de la figura.

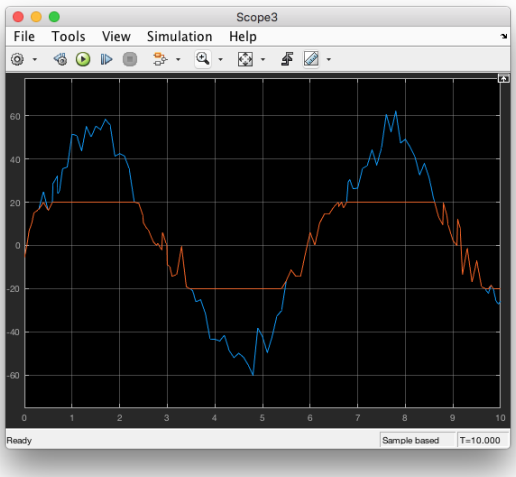
Finalmente, sobre este sistema vamos a introducir algún bloque que añada no linealidades habituales. Por ejemplo, se puede dar el caso que tanto la salida del controlador como la medida (realimentación) se encuentren saturadas (no pueda medirse ni aplicarse una acción que esté fuera de un rango determinado). Además podría suceder que el sistema no respondiera para valores pequeños de la entrada (Zona muerta). Para ver su efecto, se añadirán dos bloques de saturación y un bloque de zona muerta, y se fijarán los rangos de cada uno.

A continuación simularemos simultaneamente el modelo lineal y del saturado (se copia el modelo existente, se pega sobre el modelo Simulink, y se incluyen los nuevos elementos en uno de los dos, tal y como se muestra en la figura de la derecha). Para ver las discrepancias entre los modelos dibujamos: a) en Scope1 la entrada y salidas de ambos, y b) en Scope el valor de la señal que se calcula y el que se aplica (saturado). Se puede analizar el efecto de las no linealidades, ajustando sus parámetros de modo que se observe primero el efecto de cada una de ellas por separado, y después su efecto cuando actúan juntas.



Para poder ver más claro aún su efecto, podemos empezar por ajustar los parámetros del bloque de la perturbación para que no actúe y añadir su efecto en sucesivas simulaciones.





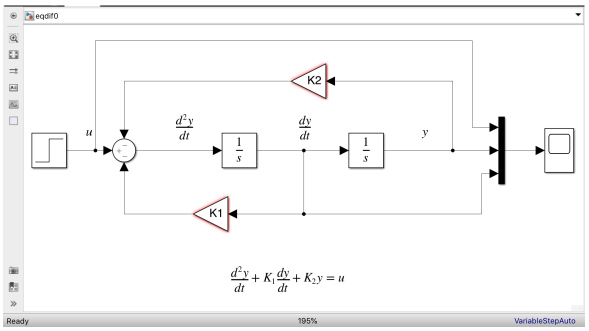
TAREA: En esta parte de la práctica hay que realizar dos tareas. Por una parte, reproducir lo anteriormente mostrado con el controlador ganancia. Y por otra, repetir todos los pasos con el controlador integral. En ambos casos se puede elegir cualquier valor de K.

Ecuaciones diferenciales

Simulink también permite modelar ecuaciones diferenciales. Por ejemplo la ecuación diferencial lineal,

$$\frac{d^2y(t)}{dt} + K_1\frac{dy(t)}{dt} + K_2y(t) = u(t)$$

puede simularse en Matlab mediante el modelo que se muestra en la figura de la derecha, en el que cada bloque $\frac{1}{s}$ se utiliza para observar a la salida la integral de la señal que hay a la entrada del bloque.



TAREA: En esta parte de la práctica hay que realizar las siguientes tareas:

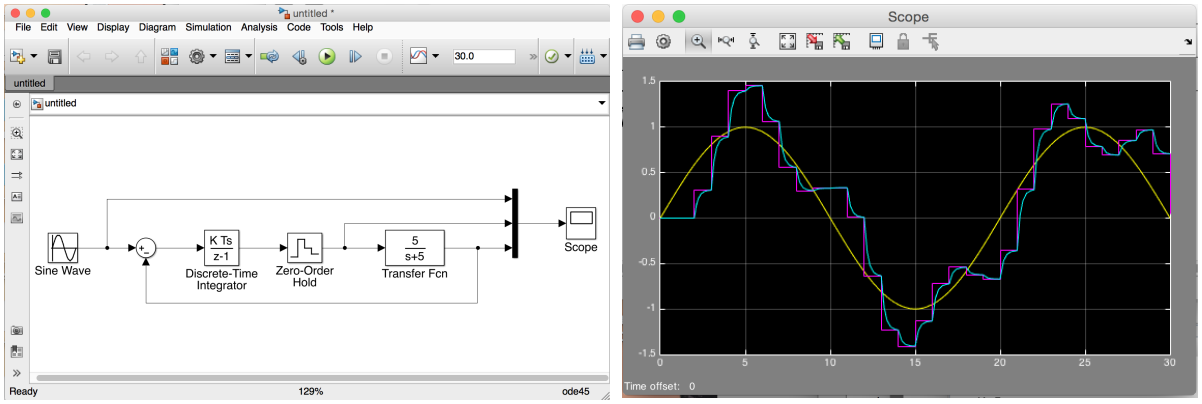
- Reproducir el modelo de la figura, comprobando que efectivamente corresponde a la ecuación diferencial descrita más arriba y analizar los resultados de la salida.
- Obtener la función de transferencia correspondiente a la ecuación diferencial.
- Añadir al modelo los bloques necesarios para modelar la función de transferencia obtenida. Comprobar que ambos modelos dan la misma salida.

- Cambiar la entrada por una entrada sinusoidal y comparar la respuesta para distintas frecuencias de la señal de entrada.
- Modificar el modelo para que pueda simular la ecuación diferencial:

$$\frac{d^2y(t)}{dt} + K_1 \frac{dy(t)}{dt} \cdot e^{-y(t)^2} + K_2 \cos(y(t)) = u(t) \tag{1}$$

Conexion entre bloques continuos y discretos

Por último en el siguiente modelo Simulink se muestra como aplicar la señal de un controlador discreto (en concreto un integrado) a un sistema continuo (utilizando como medio de conexión un retenedor de orden zero - ZOH).



TAREA: El alumno montará el sistema, observará los valores que hay en el Scope, e intentará interpretarlos. Además modificará la ganancia del integrador discreto y el periodo del integrador y del ZOH, y verá como esos cambios pueden afectar a la respuesta del sistema.