

# SISTEMAS ELECTRÓNICOS Y AUTOMÁTICOS

curso 2006-2007

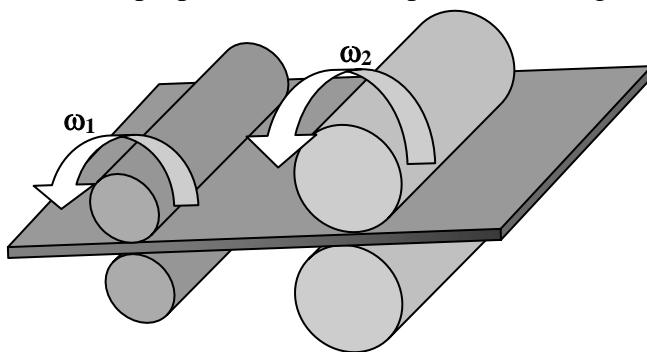
## PRÁCTICA 1: Modelado de Sistemas en el Espacio de Estados

### 1. Objetivos.

- Realizar la representación interna de sistemas continuos y discretos.
- Conocer los comandos de Matlab para transformar un sistema en representación externa a interna, y viceversa.
- Simular sistemas en el espacio de estados utilizando Simulink.
- Implementar nuevos esquemas en Simulink para representar sistemas en el espacio de estados.

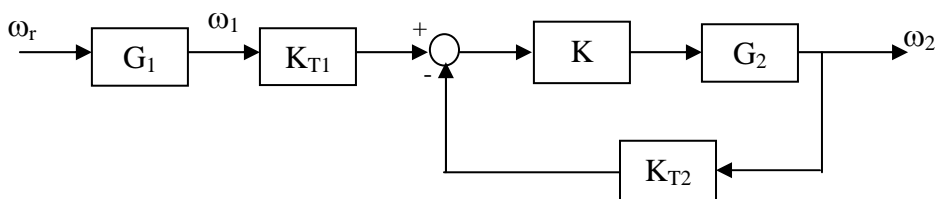
### 2. Descripción del sistema: Tren de laminación.

Un tren de laminación está formado por dos grupos de rodillos, el segundo de los cuales debe girar a una velocidad proporcional a la del primero, ver figura 1.



**Figura 1.** Tren de laminación compuesto por dos grupos de rodillos.

El diagrama de bloques del modelo simplificado de un tren de laminación se muestra en la figura 2.



**Figura 2.** Diagrama del bloques del tren de laminación.

En el diagrama de bloques:

- $\omega_r$  es la velocidad de referencia para el primer grupo de rodillos,
- $\omega_1$  es la velocidad real de giro del primer grupo de rodillos, y
- $\omega_2$  es la velocidad real de giro del segundo grupo de rodillos.

Las funciones de transferencia del primer y segundo grupo de rodillos son, respectivamente:

$$G_1(s) = \frac{10}{s+1}$$
$$G_2(s) = \frac{80}{(s+20)(s+2)}$$

$K_{T1}$  y  $K_{T2}$  son captadores de velocidad (tacómetros) que generan una señal proporcional a la velocidad que miden. Asumiremos  $K_{T1} = K_{T2} = 0.1$  \* (última cifra DNI alumno +1).

La constante K determina la relación entre la velocidad del primer grupo de rodillos y el segundo. Se considerará  $K = 5$ , lo que produce que el segundo grupo de rodillos gire a una velocidad  $\omega_2$  mitad que la del primero  $\omega_1$ .

### 3. Modelado de sistemas continuos.

Se va a modelar el sistema continuo descrito en el apartado anterior en representación interna y en representación externa. El objetivo es comprobar que la respuesta del sistema considerando diferentes representaciones es la misma.

#### 3.1. Modelado en representación externa.

**Ejercicio 1.** Calcular la función de transferencia del sistema descrito en el apartado 2.

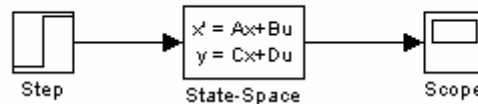
**Ejercicio 2.** Realizar la simulación del sistema en Simulink en representación externa considerando como entrada un escalón unitario. Obtener la gráfica de la respuesta del sistema.

#### 3.2. Modelado en representación interna.

Se va a demostrar la equivalencia entre la representación externa y la interna. Así mismo se va a comprobar que se puede modelar el mismo sistema en el espacio de estados utilizando diferentes ternas de matrices. Para ello se va a simular el sistema utilizando Simulink y se va a comprobar que la salida del sistema es la misma independientemente del modelo seleccionado. Para realizar la simulación en Simulink en representación interna del sistema

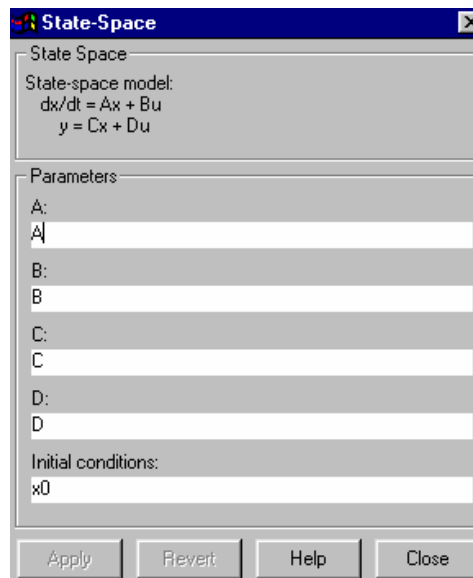
$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

se utiliza el bloque *State – Space* de la librería *Linear*, ver figura 3.



**Figura 3.** Esquema de Simulink para simular el sistema en el espacio de estados.

Los parámetros de este bloque corresponden a las matrices del modelo en el espacio de estados, junto a las condiciones iniciales, ver figura 4. Habitualmente definiremos las matrices como variables en Matlab antes de realizar la simulación, por lo que se establecerá los nombres de las variables como parámetros.



**Figura 4.** Parámetros del bloque de espacio de estados.

**Ejercicio 3.** Modelar el sistema expresado por la función de transferencia obtenida en el ejercicio 1 utilizando: (i) la forma canónica de fase (forma canónica controlable) y (ii) la forma canónica diagonal. Simular cada modelado del sistema en Simulink y comprobar que la respuesta del sistema coincide en los dos casos con la obtenida cuando se utiliza la representación externa. Considerar que el estado inicial es nulo.

Nota: Para calcular la forma canónica diagonal se recomienda utilizar la función **residue** de Matlab.

» **[R,P,K]=residue(B,A)** encuentra los residuos, polos y términos directos de la expansión en fracciones del cociente  $B(s)/A(s)$ , es decir:

$$\frac{B(s)}{A(s)} = \frac{R(1)}{s - P(1)} + \frac{R(2)}{s - P(2)} + \dots + \frac{R(n)}{s - P(n)} + K(s)$$

**Ejercicio 4.** Obtener la representación interna del sistema utilizando la función de Matlab **tf2ss**. Simular el sistema y comparar su respuesta con las obtenidas en el ejercicio 3. Considerar, al igual que en el ejercicio 3, que el estado inicial es nulo.

» **[A,B,C,D] = tf2ss(num,den)** calcula la representación en el espacio de estados:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

del sistema:

$$G(s) = \frac{\text{num}(s)}{\text{den}(s)}$$

Nota: La función **tf2ss** realiza una representación en el espacio de estados en forma canónica controlable (forma canónica de fase). Sin embargo esta representación no coincide exactamente con la definición de forma canónica controlable. Para demostrar que la representación obtenida está en forma canónica controlable hay que invertir el orden de las variables de estado. Sea:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

el sistema definido en forma canónica controlable. Si se definen nuevas variables de estado de la siguiente manera:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & \dots & 1 \\ 0 & \dots & 1 & 0 \\ \vdots & \ddots & & \vdots \\ 1 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \\ \vdots \\ w_n(t) \end{bmatrix} \Leftrightarrow x(t) = Tw(t)$$

entonces la nueva ecuación de estado:

$$\begin{aligned}\dot{w}(t) &= T^{-1}ATw(t) + T^{-1}Bu(t) \\ y(t) &= CTw(t) + Du(t)\end{aligned}$$

también está en forma canónica controlable. Esta representación es la que utiliza la función de Matlab **tf2ss**.

**Ejercicio 5.** Obtener la función de transferencia del sistema a partir de la representación interna utilizando la función **ss2tf** de Matlab. Comprobar que siempre se obtiene la misma función de transferencia independientemente del modelo de estados considerado.

» `[num,den]=ss2tf(A,B,C,D)` calcula la función de transferencia:

$$G(s) = \frac{num(s)}{den(s)}$$

del sistema:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

#### 4. Modelado de sistemas discretos.

En el apartado anterior se ha modelado el sistema continuo tanto en representación externa como en el espacio de estados. En este apartado el objetivo es obtener el modelo discreto en representación interna de un sistema continuo.

##### 4.1. Discretización de sistemas continuos en representación interna.

La discretización de un sistema continuo se realiza añadiendo a la entrada del sistema un bloqueador y a su salida un muestreador.

Considerando el sistema continuo:

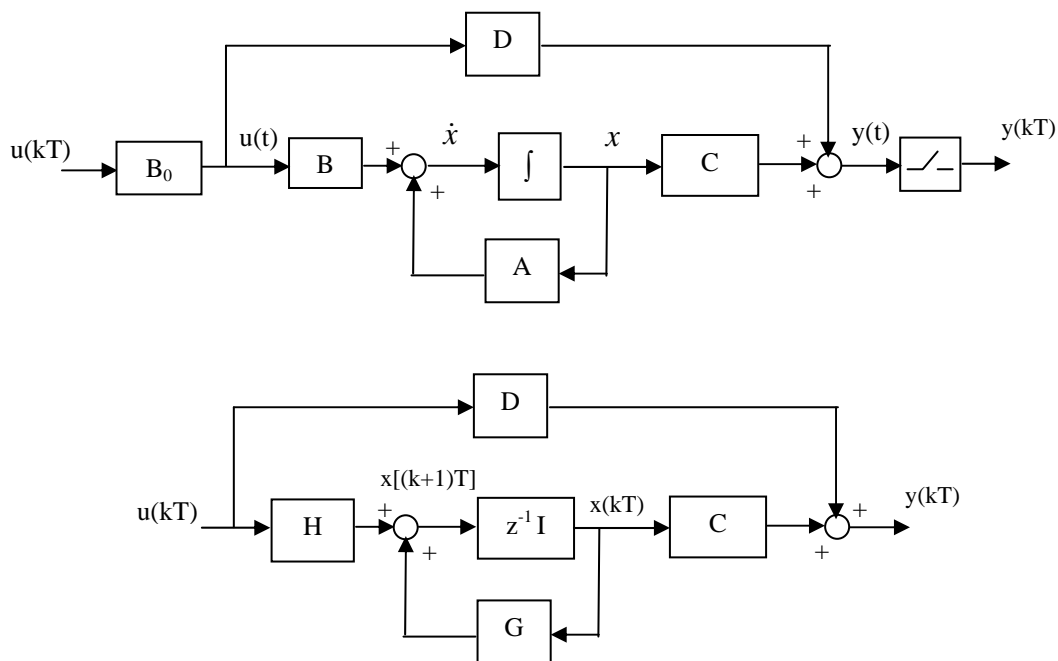
$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

el sistema discreto que sustituye al conjunto bloqueador de orden cero, sistema continuo y muestreador, ver figura 5, viene dado por:

$$\begin{aligned}x[(k+1)T] &= G(T)x(kT) + H(T)u(kT) \\ y(kT) &= Cx(kT) + Du(kT)\end{aligned}$$

donde

$$\begin{aligned}G(T) &= e^{AT} \\ H(T) &= \left( \int_0^T e^{A\lambda} d\lambda \right) B\end{aligned}$$

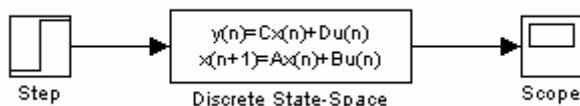


**Figura 5.** En la parte superior se muestra el sistema en tiempo continuo con un bloqueador de orden cero a la entrada y un muestreador a la salida. En la parte inferior se muestra el sistema discreto equivalente.

Para discretizar el sistema continuo en representación interna puede utilizarse la función de Matlab **c2d**.

» **[G,H]=c2d(A,B,T)** convierte el sistema continuo en el sistema discreto considerando que existe un retenedor de orden cero en la entrada y asumiendo un periodo de muestreo de T segundos.

Para realizar la simulación en Simulink de un sistema discreto en representación interna se utiliza el bloque *Discrete State – Space* de la librería *Discrete*, ver figura 6. Los parámetros de este bloque corresponden a las matrices del modelo discreto en el espacio de estados, junto a las condiciones iniciales y al periodo de muestreo en segundos.



**Figura 6.** Esquema de Simulink para simular el sistema discreto en el espacio de estados.

**Ejercicio 6.** Discretizar el sistema continuo modelado en la forma canónica de fase utilizando la función **c2d** y simular el sistema discreto en Simulink. Considerar un periodo de muestreo  $T = 0.1$  segundos y que el estado inicial es nulo. Obtener la respuesta del sistema y compararla con la respuesta del sistema continuo.

#### 4.2. Cálculo de la representación interna discreta a partir de la representación externa discreta.

En este apartado se va a calcular el modelo de estado discreto de un sistema continuo a partir de su representación externa discreta. Para ello en primer lugar es necesario discretizar el sistema en representación externa y, posteriormente, obtener la representación interna a partir de la función de transferencia discreta.

Para obtener la función de transferencia discreta a partir de la continua se puede utilizar la función de Matlab **c2dm**.

» **[numd,dend] = c2dm(num,den,Ts,'método')** convierte la función de transferencia continua  $G(s) = \frac{num(s)}{den(s)}$  a la discreta  $G(z) = \frac{numd(z)}{dend(z)}$  tomando como periodo de muestreo  $T_s$  segundos. Si se desea considerar un retenedor de orden cero en la entrada del sistema, el parámetro 'método' debe ser 'zoh'. Para obtener más información acerca de los valores que se pueden asignar a este parámetro, ejecutar *help c2dm* en Matlab.

Una vez que se tiene la representación externa discreta del sistema, mediante la función de Matlab **tf2ss** se calcula la representación interna discreta.

» **[G,H,C,D] = tf2ss(numd,dend)** calcula la representación en el espacio de estados:

$$\begin{aligned} x[(k+1)T] &= G(T)x(kT) + H(T)u(kT) \\ y(kT) &= Cx(kT) + Du(kT) \end{aligned}$$

del sistema:

$$G(z) = \frac{numd(z)}{dend(z)}$$

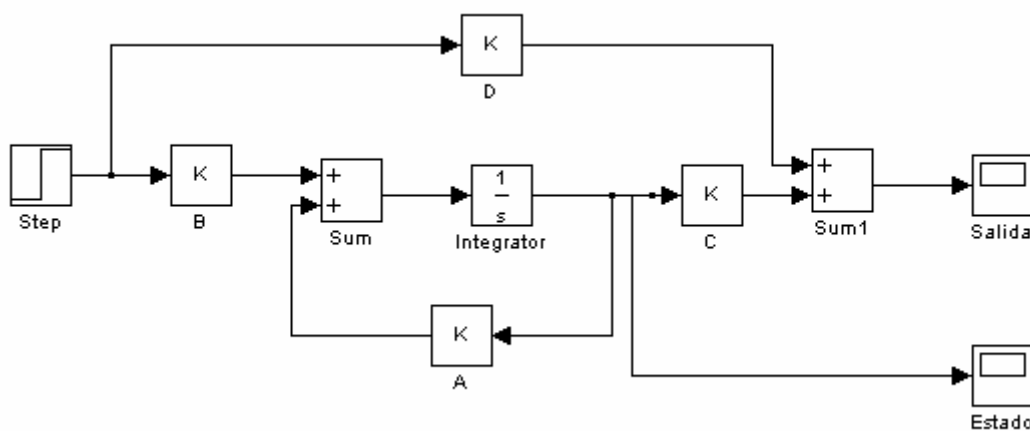
**Ejercicio 7.** Obtener la representación interna discreta del sistema a partir de la representación externa discreta. Asumir un periodo de muestreo  $T = 0.1$  segundos y condiciones iniciales nulas. Comparar el modelo obtenido con el calculado en el ejercicio 6. Comprobar que la respuesta de ambos sistemas coinciden.

### 5. Implementación en Simulink de nuevos esquemas de representación interna.

El bloque *State-Space* de la librería *Linear* y el *Discrete State-Space* de la librería *Discrete* permiten simular el comportamiento de sistemas en representación interna continuos y discretos, respectivamente. Ambos bloques proporcionan la salida del

sistema. Sin embargo, en el espacio de estados, además de la salida del sistema, interesa conocer su estado. Por esta razón se van a implementar en Simulink dos nuevos esquemas en representación interna, en continuo y en discreto, que permiten conocer el estado y la salida del sistema.

**Ejercicio 8.** Implementar en Simulink el esquema en representación interna continua que se muestra en la figura 7. El bloque *Matriz Gain* de la librería *Linear* permite definir cada una de las matrices que aparecen en las ecuaciones de estado y de salida. El bloque *Integrator* de la librería *Linear* se utiliza para implementar la integral. Hay que destacar que en este bloque debe especificarse el vector de estados inicial.

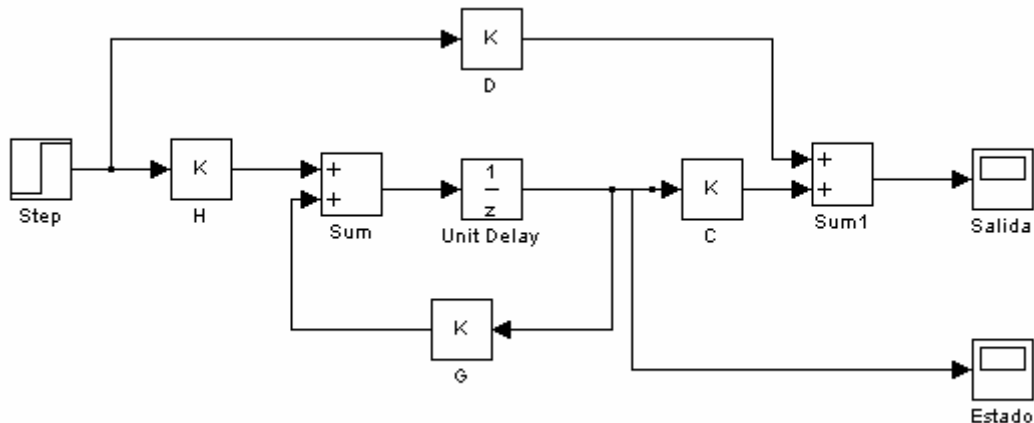


**Figura 7.** Representación en el espacio de estados de un sistema continuo.

Simular el nuevo esquema ante una entrada en escalón y obtener las gráficas de la respuesta y el estado del sistema considerando el modelo en forma canónica de fase. Comprobar que la respuesta del sistema coincide con la que se obtiene al utilizar el bloque *State-Space* de la librería *Linear* de Simulink.

**Ejercicio 9.** Implementar en Simulink el esquema en representación interna discreta que se muestra en la figura 8. El bloque *Unit Delay* de la librería *Discrete* se utiliza para introducir un retardo de un periodo de muestreo. Hay que destacar que en este bloque debe especificarse el periodo de muestreo (en segundos) y el vector de estados inicial.



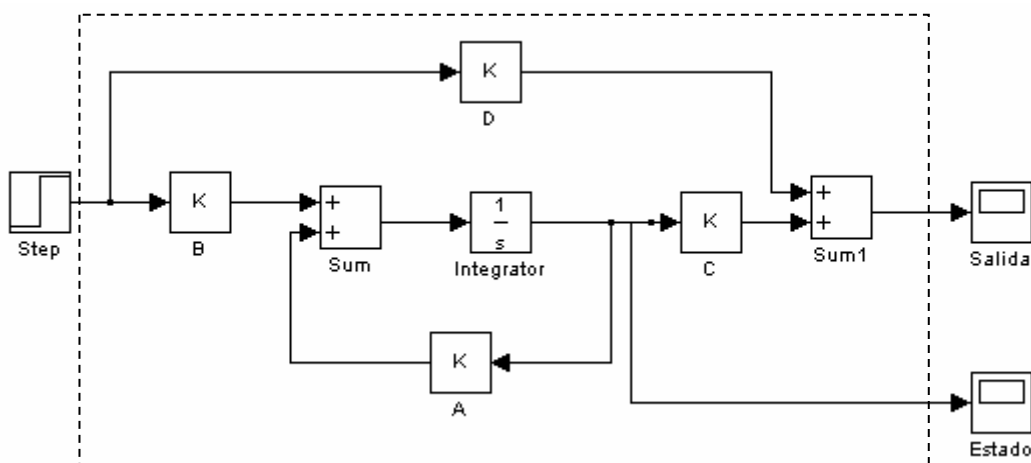


**Figura 8.** Representación en el espacio de estados de un sistema discreto.

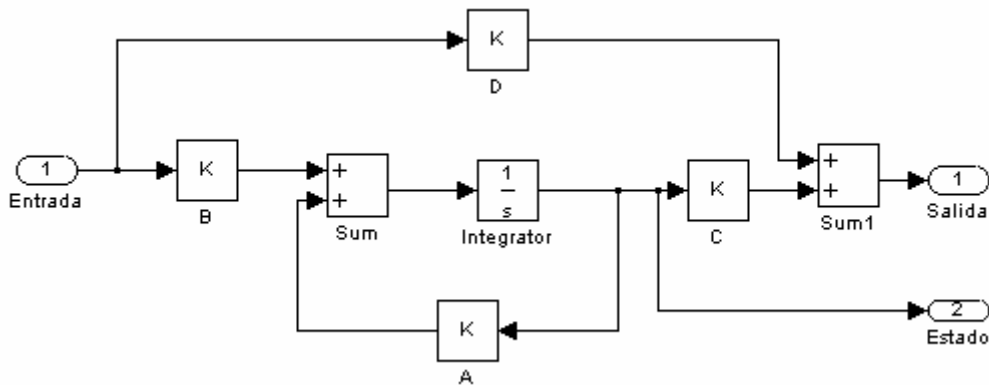
Simular el nuevo esquema ante una entrada en escalón y obtener las gráficas de la respuesta y el estado del sistema discreto correspondiente al sistema continuo simulado en el ejercicio anterior. Comparar las gráficas del estado y de la respuesta de ambos sistemas. Comprobar que la respuesta del sistema coincide con la que se obtiene al utilizar el bloque *Discrete State-Space* de la librería *Discrete* de Simulink.

**Ejercicio 10 (optativo).** En Simulink existe la posibilidad de crear nuevos bloques a partir de bloques existentes mediante la creación de subsistemas. Así mismo se pueden asignar a los bloques creados cajas de diálogos en las que introducir los parámetros que utiliza el subsistema. Con el objetivo de utilizar los nuevos esquemas implementados en próximas prácticas se va a crear un subsistema para cada esquema y se le va a asignar una máscara para introducir los parámetros asociados a cada esquema. En el caso del modelo de estados continuo deben realizarse los siguientes pasos:

- 1) Seleccionar los bloques correspondientes al espacio de estados implementado, ver figura 9, y crear un subsistema (*Edit* → *Create Subsystem*). La apariencia del subsistema creado debe ser como la que se muestra en la figura 10.

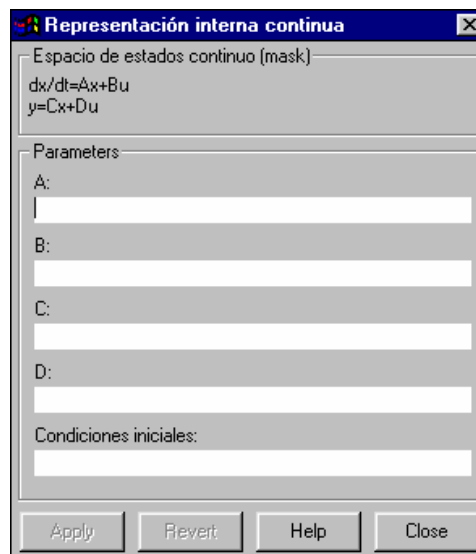


**Figura 9.** Área con los bloques a seleccionar para crear el subsistema.

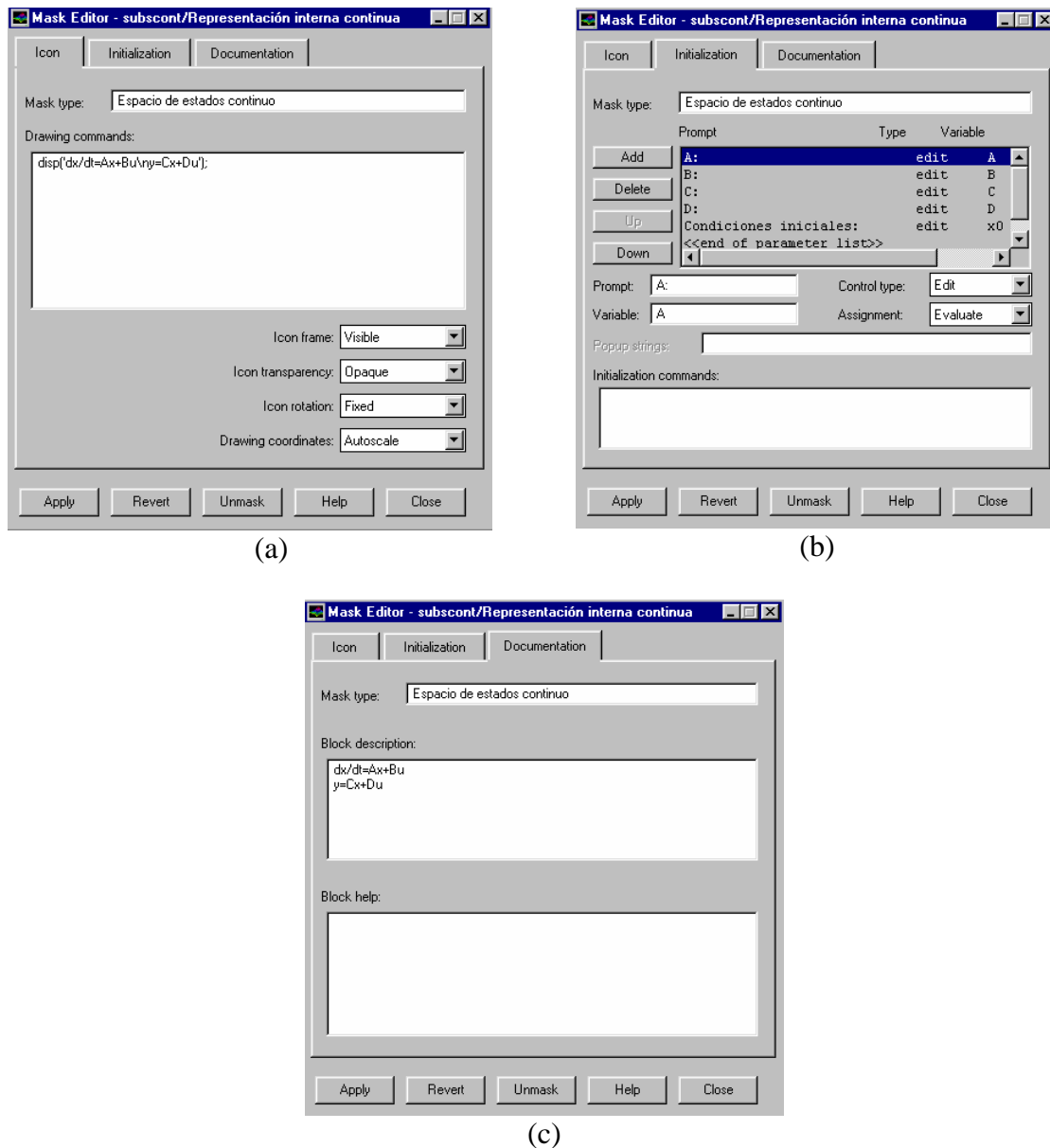


**Figura 10.** Subsistema correspondiente a la representación interna continua.

- 2) Crear una máscara para el subsistema. Para ello debe ejecutarse la opción del menú de Simulink *Edit* → *Mask Subsystem*, que abre el editor de máscaras. En el editor de máscaras se especifican los parámetros que utiliza el subsistema, de forma que al pinchar dos veces sobre el subsistema aparecerá una caja de diálogo que pedirá los valores de dichos parámetros. Así mismo se define la apariencia del bloque del subsistema creado. La apariencia final de la caja de diálogo asociada al subsistema creado debe ser similar a la que se muestra en la figura 11. Los valores que deben introducirse en cada una de las ventanas del editor de máscaras se muestran en la figura 12.



**Figura 11.** Caja de diálogo con los parámetros del subsistema creado.



**Figura 12.** Valores a introducir en cada una de las ventanas del editor de máscaras.

Nota: Una vez que se ha creado la máscara, para modificarla hay que seleccionar la opción del menú *Edit* → *Mask*. En el caso de que se quiera acceder al subsistema creado hay que ejecutar *Edit* → *Look Under Mask*.

Realizando los dos pasos especificados el modelo en representación interna continua será como el que se muestra en la figura 13.

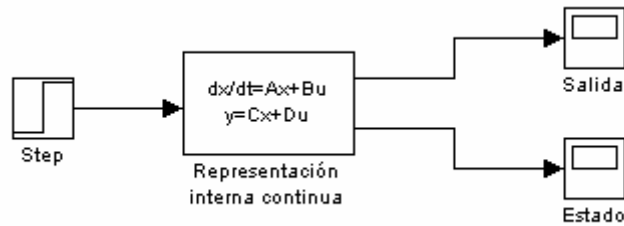


Figura 13. Modelo en representación interna continua.

Comprobar el funcionamiento correcto del subsistema creado.

**Ejercicio 11 (optativo).** Siguiendo el mismo procedimiento explicado en el ejercicio anterior, crear un subsistema, con su correspondiente máscara, para el modelo en representación interna discreta creado en Simulink. Comprobar el funcionamiento correcto del subsistema.

### Importante

- Debe entregarse un informe detallando cada uno de los ejercicios realizados en la práctica.
- ~~Fecha de entrega: hasta el día 25 de enero de 2007~~