SIMULAB: UN ENTORNO DE SIMULACION IDONEO PARA LA DOCENCIA EN AUTOMATICA.

F. Morilla, A. Pérez de Madrid

Dpto de Informática y Automática, UNED
Avda Senda del Rey s/n, 28040 Madrid
Tfn: 91-3987156, Fax: 91-5446737, E-mail: Fernando.Morilla@human.uned.es

RESUMEN.

Este documento, que incluye un apartado dedicado a la presentación de SIMULAB y otro a la descripción de un ejemplo de aplicación, no pretende servir al lanzamiento de SIMULAB ni a los intereses de MathWorks, tampoco contiene ningún estudio comparativo de SIMULAB con otros lenguajes de simulación, simplemente pretende transmitir a la comunidad de docentes en Automática, amplíamente representada en estas Jornadas, que SIMULAB nos parece un entorno idóneo para la docencia en Automática. Del que todos sacaremos buen partido si unimos esfuerzos en el desarrollo de bloques, subsistemas o sistemas con la clara ídea de difundirlos. Por otro lado, los autores se ofrecen, en nombre del Departamento, a desarrollar la labor de coordinación necesaria para que todos los usuarios de SIMULAB salgan beneficiados de ello.

INTRODUCCION.

MATLAB de MathWorks Inc., que tiene sus orígenes en 1984, ha alcanzado un gran nivel de difusión no sólo en entornos universitarios sino también en entornos industriales. Todo ello debido no sólo a que hay versiones para casi todas las máquinas, sino también por las características que ofrece y porque ha conseguido adaptarse a las necesidades de sus usuarios.

En el Departamento de Informática y Automática de la UNED somos usuarios habituales de MATLAB desde el año 1987, siempre nos hemos preocupado de sacarle el mayor rendimiento (en la investigación y en la docencia) y de difundirlo; a nuestros alumnos de Automática a traves de prácticas en el laboratorio y en el entorno multimedia "HyperAutomática", a nuestros alumnos del Programa de Doctorado como herramienta muy importante en la realización de los trabajos de investigación, al mundo industrial a través de los cursos de formación impartidos a

empresas, a profesores de bachillerato a través del curso de verano "Computadores y Matemáticas".

Somos, por tanto, buenos conocedores de la evolución que ha experimentado MATLAB, que por un lado ha facilitado su uso y por otro lado ha aportado nuevas herramientas en el análisis y diseño de sistemas de control. Prueba de ello es la cantidad de "Toolboxes" existentes en la actualidad, véase la lista incluida a continuación, donde entre paréntesis se indican las versiones que estamos utilizando en el Departamento. A éstos se ha unido recientemente SIMULAB, de esta manera MathWorks se adentra en el mundo de la simulación de sistemas no lineales en competencia directa con otros paquetes típicos de simulación.

Familia de productos MATLAB:

SIMULAB (v 1.2 , mayo 1992)
μ-Analysis and Synthesis Toolbox (abril 1991)
Control System Toolbox (v 3.03, febrero 1991)
Signal Processing Toolbox (v 2.08, febrero 1991)
System Identification Toolbox (v 3.0 julio 1991)
Robust-Control Toolbox
Spline Toolbox
Optimization Toolbox
Chemometrics Toolbox
MMLE3 State-Space Identification Toolbox
The Block Diagram Toolbox (marzo 1991)
The Control Kit (junio 1991)

SIMULAB.

El nuevo programa de MathWorks para simulación (modelización y análisis) de sistemas dinámicos no lineales fué presentado en 1990, con el nombre de SIMULAB para computadores personales y con el nombre de SIMULINK para estaciones de trabajo. Desde mayo de 1992 está disponible la versión 1.2. El programa está soportado sobre la versión de MATLAB para X Window y para Macintosh y es similar a los "Toolboxes" de MATLAB, en el sentido que añade características específicas a éste, al mismo tiempo que aprovecha toda la potencia de MATLAB.

SIMULAB está especialmente pensado para el análisis y diseño de sistemas de control, aunque es totalmente útil en el estudio de cualquier sistema: lineal, no lineal, continuo, discreto, o híbrido. La principal diferencia respecto a los otros "Toolboxes" es que SIMULAB ofrece un entorno gráfico de ventanas, en el que los sistemass se construyen o se modifican haciendo uso del ratón y en el que el usuario dispone de unas capacidades gráficas ilimitadas. De esta forma, se reduce considerablemente el tiempo empleado en la construcción y prueba de un modelo del sistema y se dispone de una forma más natural e intuitiva para describirlo.

El usuario puede crear sus modelos a partir de una librería de componentes básicos y de los nuevos bloques que vaya generando, simplemente copiándolos de una ventana a otra y estableciendo las conexiones oportunas. Entre los tipos de bloques que ofrece SIMULAB se encuentran los siguientes:

Fuentes: reloj del sistema, constante, fichero de datos, variable de MATLAB, generador de señales periódicas (sinusoidal, cuadrada, triangular, secuencia repetitiva), generador de señales aperiódicas (escalón, pulso, ruido blanco, barrido en frecuencia).

Sumideros: fichero de datos, variable de MATLAB, registrador gráfico (en ventana específica, en ventana gráfica de MATLAB), registrador gráfico XY, analizadores gráficos específicos (densidad espectral de potencia, analizador de respuesta en frecuencia, función de correlación, función de autocorrelación).

Discretos: retardo unitario, retenedores (de orden cero y de primer orden), bloque genérico descrito en el espacio de estados o por su función de transferencia, bloque discretizado.

Lineales continuos: sumador, amplificador, integrador, derivador, bloque genérico descrito en el espacio de estados o por su función de transferencia.

No lineales continuos: valor absoluto, multiplicador, función genérica, histéresis, zona muerta, limitador de cambio, tabla (de una y de dos variables de entrada), relé, saturación, selector, retardo, integrador con saturación, cuantificador, memoria, transformación de coordenadas, función de MATLAB, "S-function".

Conexiones: entrada, salida, multiplexor, demultiplexor.

Filtros: analógicos y digitales (paso baja, paso banda, rechazo de banda, paso alta) en sus distintas aproximaciones (Butterworth, Chebychev, elíptico, FIR, IIR).

Controladores: PID, PID con saturación del término integral, matriz de ganancias, controladores óptimos de diferentes tipos (filtro de Kalman, realimentación de estados).

Estimadores no recursivos: para modelos AR, ARX, ARMAX, OE, BJ (Box-Jenkins) y para el modelo general usando el método de error de predicción.

La figura 1 muestra la ventana con los bloques elementales más utilizados en la construcción de sistemas. La figura 2 recoge un ejemplo de utilización de distintos bloques elementales, concretamente la estructura interna del controlador PID con saturación del término integral que ofrece el propio SIMULAB como un bloque controlador elemental.

El análisis del sistema, mediante simulación, se puede efectuar desde el menú de simulación asociado a todo modelo en SIMULAB, desde la línea de comandos en MATLAB o desde cualquier programa en MATLAB. En la simulación se puede elegir uno cualquiera de los cinco siguientes métodos de integración: lineal, Runge-Kutta de 3er ó 5º orden, predicción-corrección de Gear o de Adams, Euler. La figura 3 muestra la ventana con el menú de parámetros de

simulación, durante la simulación no es posible modificar los parámetros generales, pero si es posible efectuar cambio en los parámetros específicos de cada bloque, dotando así a SIMULAB de un entorno interactivo muy útil para la realización de pruebas.

Adicionalmente a la simulación es posible: obtener un modelo lineal del sistema frente a determinadas perturbaciones, obtener las condiciones de equilibrio del sistema a partir de distintas condiciones iniciales y utilizar toda la potencia de MATLAB para el análisis de datos o para el análisis y diseño del modelo lineal.

Para que un conjunto de bloques, como es el caso del controlador PID, se pueda utilizar posteriormente de forma cómoda, como un subsistema, SIMULAB facilita la dotación de identidad, mediante un icono y una máscara que llevan asociados: una etiqueta, una función de transferencia o gráfica, un mensaje de ayuda en la utilización del bloque, el diálogo con el usuario, y la ejecución de una serie de comandos en la inicialización. El diálogo con el usuario sirve para que si el bloque, como es este caso, tiene asociado una serie de parámetros, se active una ventana de petición de parámetros cada vez que el bloque es utilizado. Véase como ejemplo la figura 4 que representa la ventana de diálogo del usuario con el bloque controlador PID de la figura 2.

Todo sistema presente en una ventana de SIMULAB tiene una representación interna en un fichero "*.m" que constituye la "S-function" del sistema. Esta función tiene un papel relevante durante toda la simulación, recibe el instante de tiempo t, el vector de estados x, el vector de entradas u, un "flag" y los parámetros específicos de la función, y debe facilitar según sea el "flag": información del sistema (nº de estados continuos y discretos, nº de salidas, nº de entradas, nº de raíces discontinuas, si existe o no camino directo de la entrada a la salida, el orden de los estados), información del cambio en los estados dx/dt ó x(n+1), información de la salida, información del próximo instante de actualización si el sistema es discreto, información sobre las raíces del sistema. La creación de una "S-function" se puede hacer desde el propio SIMULAB mediante el diagrama de bloques, o desde un programa en MATLAB en C o en Fortran. Cumpliendo con esta estandarización, el usuario puede hacer que SIMULAB crezca indefinidamente.

EJEMPLO.

A continuación se muestra un ejemplo, en el que se ha utilizado SIMULAB para el desarrollo de un bloque estimador de parámetros recursivos por el método de los mínimos cuadrados (RLS). En este desarrollo se podría haber hecho uso de las funciones de estimación recursivas incluidas en el "System Identification Toolbox", sin embargo por motivos didácticos se ha preferido

hacer el desarrollo utilizando bloques elementales de SIMULAB y generando varios bloques del tipo "S-function" en MATLAB. En una primera versión sólo se considera el algoritmo básico de mínimos cuadrados para un modelo lineal ARX del sistema, para que nuestros alumnos del curso de doctorado "Control Adaptativo" incorporen otras opciones como son: factor de olvido constante, factor de olvido variable, reinicialización de la matriz de covarianza, traza constante.

El algoritmo de estimación recursiva por mínimos cuadrados queda resumido en las siguientes expresiones

$$\theta(k) = \theta(k-1) + \frac{P(k)\,\phi(k)}{1 + \phi(k)^T P(k)\,\phi(k)} \left[y(k) - \phi(k)^T \theta(k-1) \right] \qquad (1)$$

$$P(k+1) = P(k) - \frac{P(k)\,\phi(k)\,\phi(k)^T P(k)}{1 + \phi(k)^T P(k)\,\phi(k)} \qquad (2)$$

siendo

$y(k) = \phi(k)^T \theta$, el modelo ARX del sistema

$u(k)$ la entrada al sistema

$y(k)$ la salida del sistema

$\theta = [\ a_1 \ldots a_{na}\ b_1 \ldots b_{nb}\ ]$ el vector de parámetros

$\phi(k) = [\ -y(k-1) \ldots -y(k-n_a)\ u(k-1) \ldots u(k-n_b)\ ]$ el vector de regresión

Cuando el usuario solicita la utilización del bloque RLS se le reclaman: el número de parámetros, a estimar, del polinomio numerador y del polinomio denominador, el período de muestreo a emplear, el valor inicial para los elementos de la diagonal de la matriz de covarianza y la estima inicial de los parámetros del modelo. La figura 5 muestra la estructura interna del bloque, que tiene dos entradas (por la puerta 1 se recibe la salida del sistema a modelar y por la puerta 2 se recibe la entrada al sistema) y una salida (el vector de parámetros estimados). En la estructura se han empleado los siguientes bloques elementales de SIMULAB (2 puertas de entrada, 1 puerta de salida, 2 multiplexores y un inversor) y los siguientes bloques del tipo "S-function" (2 bloques "sregre", 1 bloque "scov" y 1 bloque "steta") desarrollados con este fin.

El bloque "sregre" recibe como entrada un escalar y genera un vector con los últimos n valores muestreados de la entrada. Combinando dos bloques "sregre" (con los tamaños adecuados y sincronizados al mismo período de muestreo), un inversor y un multiplexor se consigue formar el vector de regresión $\phi(k)$. El bloque "scov" recibe como entrada el vector de regresión y genera la matriz de covarianza P(k+1), de acuerdo con la expresión (2), presentándola a su salida como un vector columna. El bloque "steta" recibe como entrada un vector con la salida

del sistema y(k), el vector de regresión $\phi(k)$ y las columnas de la matriz de covarianza P(k) y genera el vector de parámetros $\theta(k)$, de acuerdo con la expresión (1).

La figura 6 muestra un sistema de control realimentado con control PID, en el que se ha utilizado el bloque RLS para obtener un modelo discreto del sistema. Desde un entorno como SIMULAB resulta entonces cómodo, hacer cambios en los distintos componentes del sistema en cuestión y observar los efectos en la estimación de parámetros, ya sea a través del registro gráfico o utilizando las herramientas propias de MATLAB.

BIBLIOGRAFIA.

SIMULINK User´s Guide. The MathWorks Inc., March 1992.

SIMULINK Release Notes Version 1.2. The MathWorks Inc., May 1992.

Hang, C. C., K. J. Aström and W. K. Ho. Refinements of the Ziegler-Nichols tuning formula for PID autotuners. IEE Proceedings-D, Vol. 138, No. 2, march 1991, pp. 111-118.

Ljung, L., T. Söderström. Theory and Practice of Recursive Identification. The MIT Press, 1983.

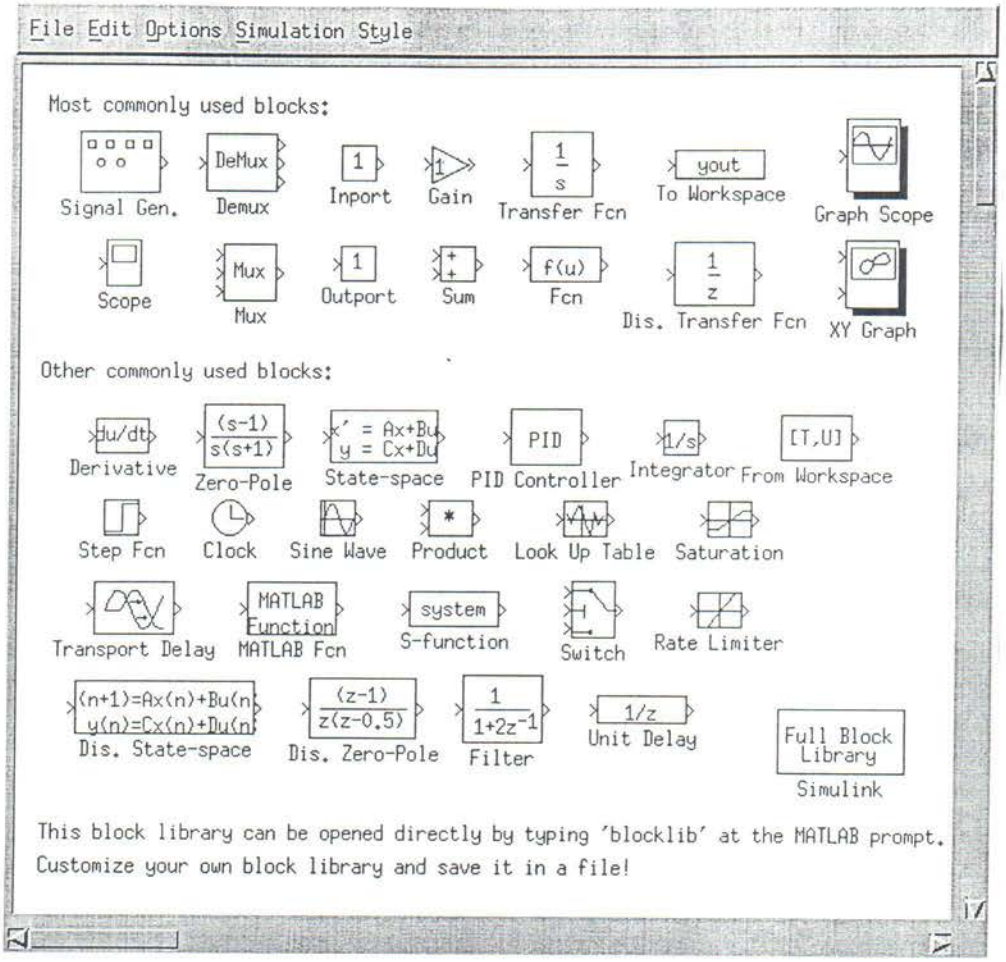

Fig. 1. Ventana 'blocklib' con los bloques elementales más utilizados en SIMULAB.

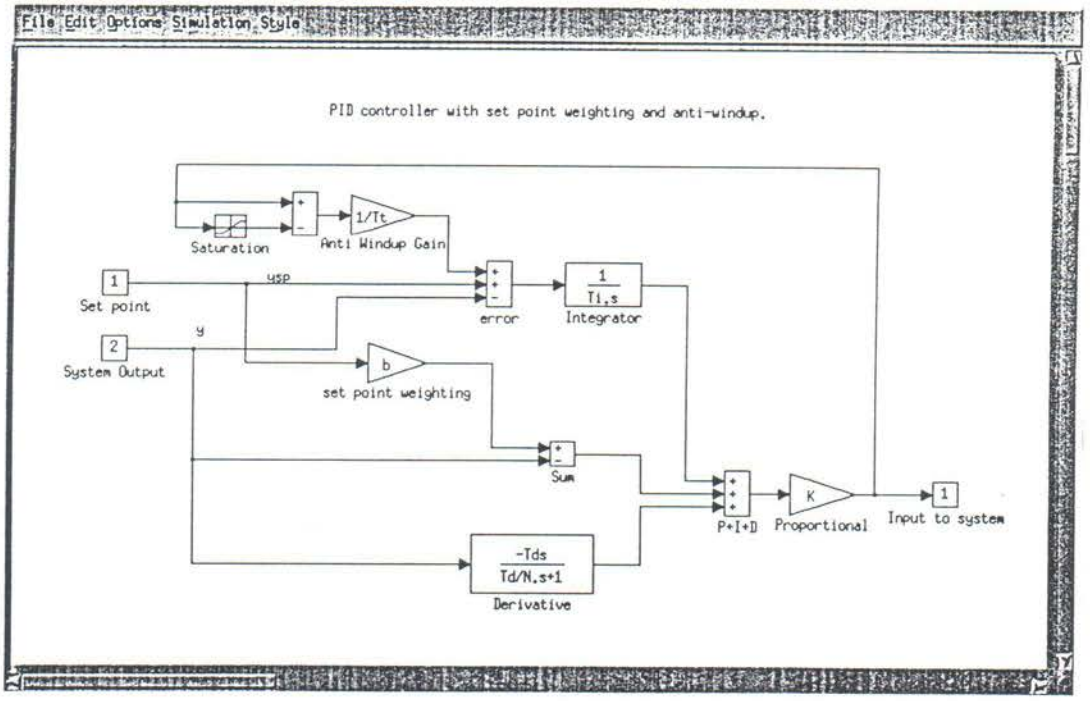

Fig. 2. Estructura interna del bloque 'Set point PID with anti-windup' perteneciente a la librería de controladores de SIMULAB.

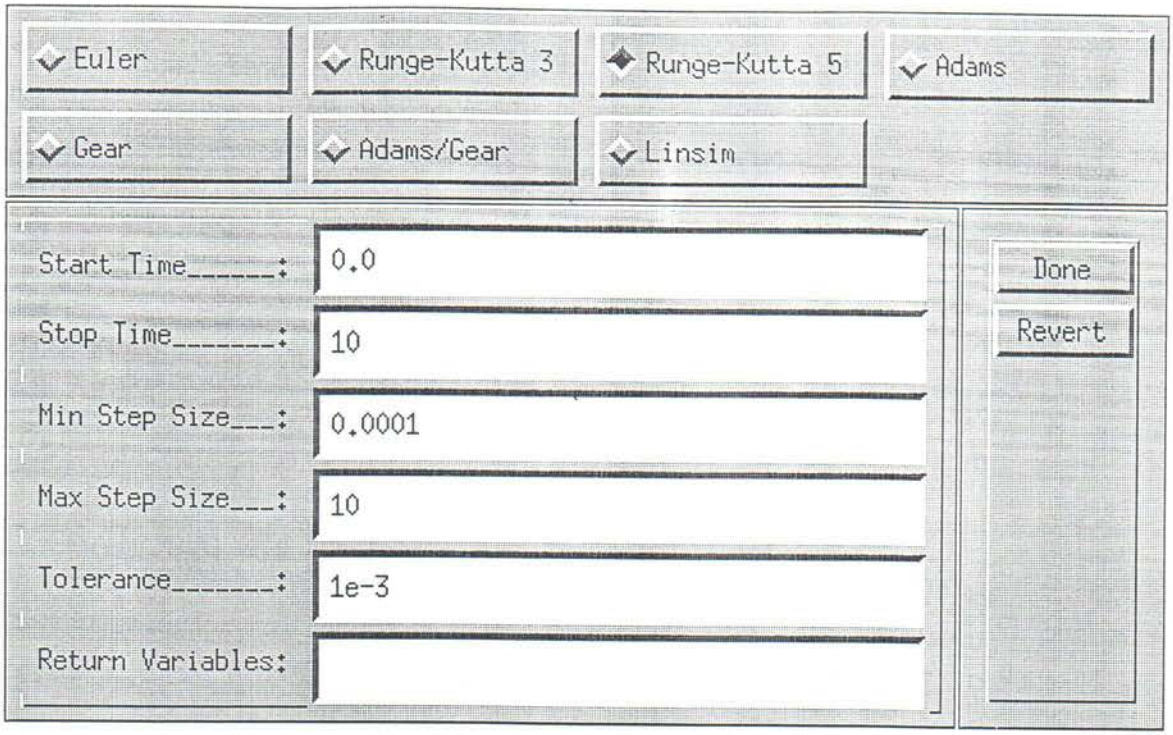

Fig. 3. Ventana para modificación de los parámetros generales de simulación.

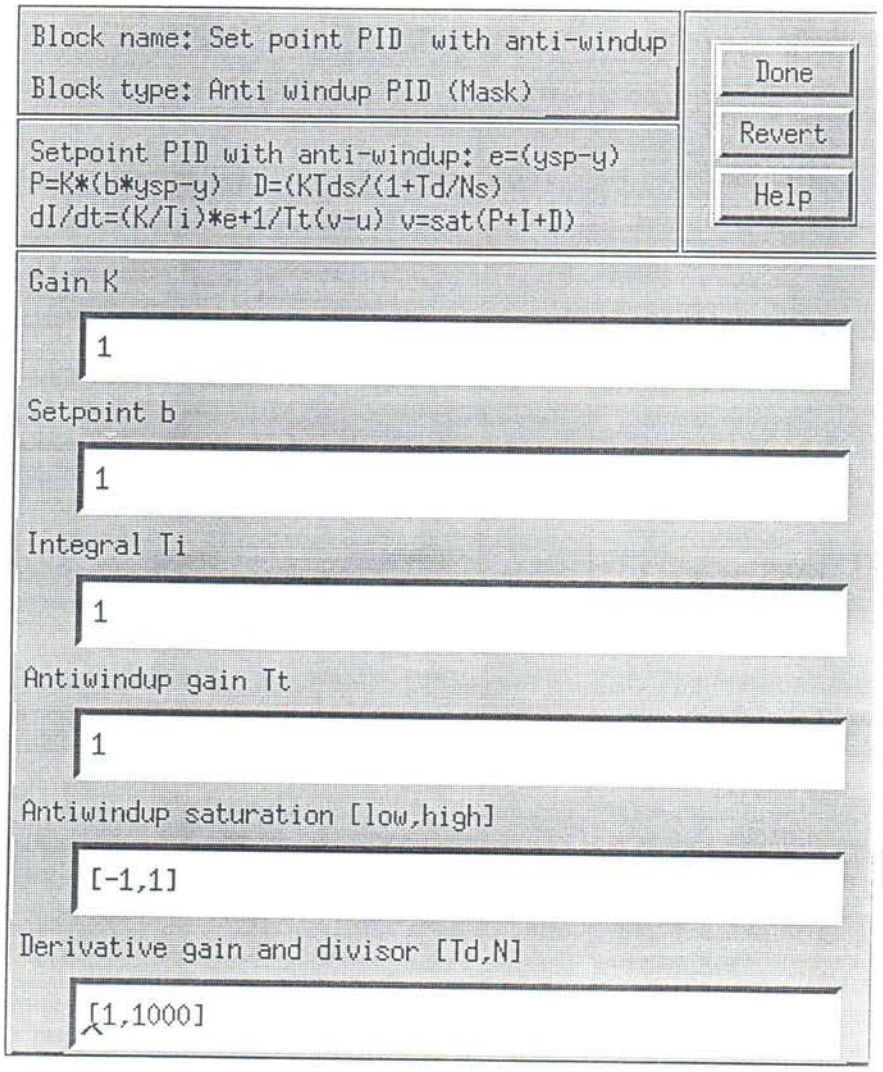

Fig. 4. Ventana de diálogo con el usuario del bloque 'Set point PID with anti-windup'.

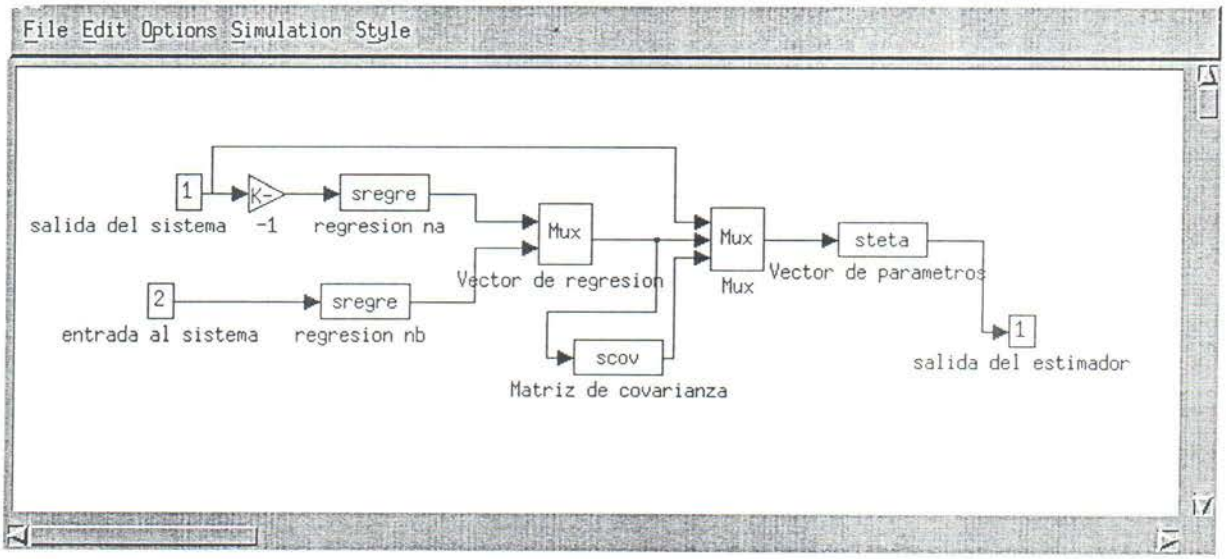

Fig. 5. Estructura interna del bloque 'RLS' desarrollado por el Departamento de Informática y Automática de la UNED.

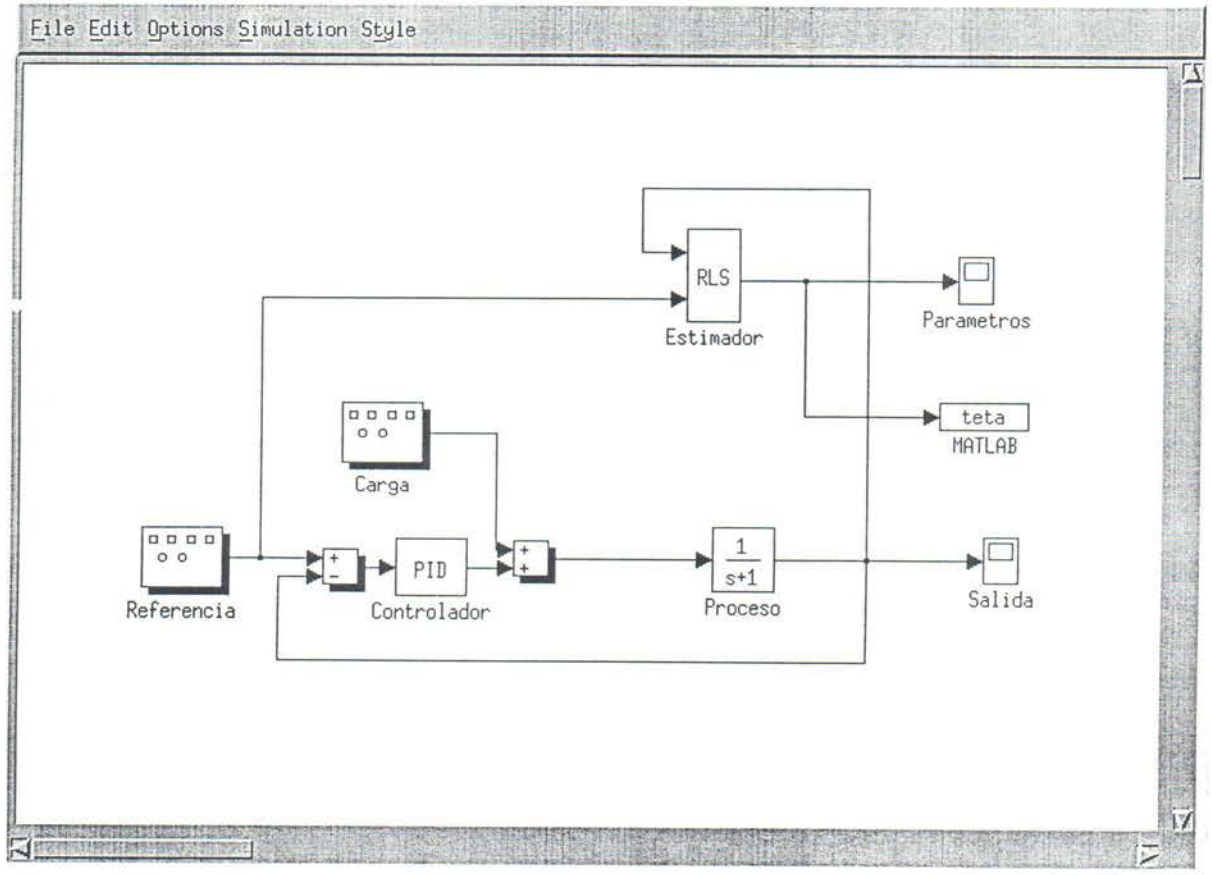

Fig. 6. Ejemplo de sistema de control realimentado en el que se utiliza un bloque 'RLS' para estimación de un modelo del sistema.