

# PROTOTIPADO RÁPIDO PARA PRÁCTICAS DE LABORATORIO DE CONTROLADORES BORROSOS EN MATLAB

C. Betancor Martín<sup>1</sup>, A. Vega Martínez<sup>2</sup> y J.M. Cerezo Sánchez<sup>3</sup>

Universidad de Las Palmas de G.C., Departamento de Ingeniería Electrónica y Automática, Instituto Universitario de Microelectrónica Aplicada

<sup>1</sup>cbetancor@diea.ulpgc.es <sup>2</sup>avega@diea.ulpgc.es <sup>3</sup>jcerezo@diea.ulpgc.es

## RESUMEN

*MATLAB posee una serie de toolbox mediante los cuáles se puede realizar el prototipado rápido para sistemas de control borroso. En este artículo se resumen estos toolbox y qué es lo que se debe disponer para llevar a cabo una práctica de laboratorio para el control borroso de velocidad de un motor de corriente continua e imán permanente.*

## 1. INTRODUCCIÓN

Uno de los problemas que surge a la hora de elaborar prácticas de control es disponer de la planta física a controlar, que normalmente suele tener un coste elevado. Además, se debe disponer de un modelo matemático del proceso a controlar, cuestión que muchas veces supone un problema. En este trabajo se desarrolla una planta con un coste mínimo y se realiza el control borroso aplicando el principio de prototipado rápido de MATLAB. Con ello, el alumno depura el algoritmo de control con la planta física real, llevando a cabo simulaciones en tiempo real. Además, el alumno podrá realizar la identificación experimental de la planta, acercándose al máximo a la realidad.

MATLAB<sup>®</sup>, Simulink<sup>®</sup> y otros productos de *The Mathworks* permiten:

- ✓ Diseñar algoritmos, modelar y simular el sistema en un único entorno integrado.
- ✓ Simular y optimizar el diseño para detectar y corregir errores en las fases iniciales.
- ✓ Probar y validar el diseño en tiempo real.
- ✓ Distribuir el software en un sistema embebido.

Entre diferentes toolbox que posee MATLAB<sup>®</sup> para el diseño de sistemas de control se encuentra *Fuzzy Logic Toolbox* (FLT) [1], el cual proporciona las herramientas para crear y editar sistemas de inferencia borrosa dentro del entorno de computación numérico de MATLAB<sup>®</sup>, pudiéndose integrar en simulaciones dentro de Simulink<sup>®</sup> [2].

El toolbox *Real-Time Windows Target* (RTWT) de MATLAB<sup>®</sup> nos permite realizar simulaciones en tiempo real [3], de forma que el algoritmo de control corre en el entorno MATLAB-Simulink y el proceso a controlar está físicamente disponible.

## 2. DESCRIPCIÓN DEL SISTEMA

Se ha construido la planta a partir de dos motores de los que únicamente se conoce que son de corriente continua (DC) e imán permanente (PM) y como características su tensión nominal de 20 V y su corriente nominal de 1 A. A uno de los motores se le hizo actuar como tal y al otro como generador, acoplando los ejes. El generador, en definitiva, hará las veces de carga mecánica variable al modificar una carga eléctrica conectada a su armadura.

Puesto que el motor tiene un encoder acoplado, se diseñó a partir de él un tacómetro, debido a que el objetivo que se persigue es el control de velocidad; dicho tacómetro se construyó a partir del circuito integrado LM2917N. Si se quiere control de posición se dispone de las dos señales en cuadratura que genera el encoder.

Otro diseño a realizar es el servo-amplificador, ya que a la salida del controlador no se tiene potencia suficiente para atacar a éste. Basado en el circuito integrado OPA549T, se desarrolló dicho amplificador.

Una vez diseñado el tacómetro y servo-amplificador, fue necesario saber el modelo matemático de la planta, para poder diseñar controladores y simularlos de forma rápida en tiempo no real. Por ello, se procedió a la identificación de la planta, un tema de crucial importancia a la hora de disponer de un modelo matemático que se acerque lo más posible a la realidad. La identificación se realizó accediendo al exterior desde Simulink, con RTWT, para aplicar al motor un escalón y recoger la información sobre las revoluciones por segundo (r.p.s.). Por lo tanto, es necesaria una tarjeta de adquisición, que en este caso fue la PCL-818L de Advantech. Una vez creada e identificada la planta, se puede comenzar a diseñar controladores para ésta.

En la Figura 1 se muestra la estructura del sistema de prototipado rápido.

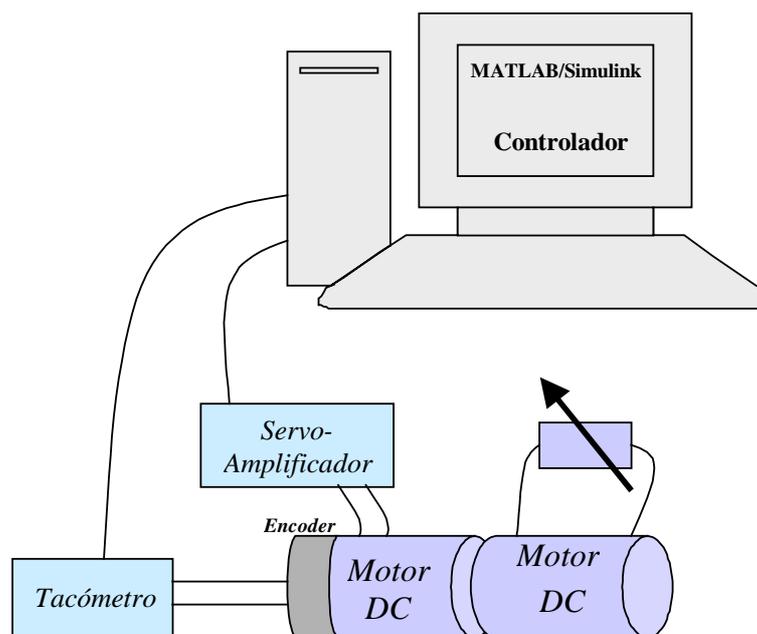


Figura 1. Estructura del sistema para prototipado rápido

### 3. HARDWARE EXTERNO

Dado que a la salida del controlador se dispone de una potencia pequeña, incapaz de atacar al motor, es necesario amplificarla en potencia, función que lleva a cabo el Servo-Amplificador.

El Servo-Amplificador es lineal y está diseñado entorno al circuito integrado *OPA549T* de *Burr-Brown*. Se trata de un amplificador operacional de bajo coste e ideal para atacar a una amplia variedad de cargas debido a la alta precisión que ofrece, incluso para señales de bajo nivel, y a las salidas en corriente y tensión máxima que puede suministrar. Otra de las ventajas que ofrece es que puede ser usado en cualquier configuración que pueda ofrecer un amplificador operacional. Además, posee las siguientes funcionalidades:

- ✓ Fijar la corriente máxima de salida entre 0 A y 10 A.
- ✓ Habilitar o deshabilitar la salida: ahorrando potencia cuando el circuito no trabaje o en sistemas multiplexados.
- ✓ Indicación de shutdown térmico: cuando la temperatura de la unión llega a los 160 °C la salida queda deshabilitada automáticamente hasta que la temperatura baje a 140°.

Para el circuito de realimentación será necesario diseñar un tacómetro. El tacómetro de corriente continua es un generador que produce una tensión proporcional a la velocidad de rotación del eje del motor. El motor tiene incorporado un *encoder* que produce dos señales cuadradas desfasadas 90°. La frecuencia de dicha señal cuadrada es de 1 KHz por cada revolución por segundo (r.p.s.). Para generar una tensión continua proporcional a la velocidad de rotación, se puede incorporar un convertidor frecuencia / tensión (*F/V*). Para esta función se ha elegido al circuito integrado *LM2917N* de *National Semiconductor*.

El *LM2917N* produce una tensión de salida proporcional a la frecuencia de entrada. Además, debido a que el principio de operación de este circuito se basa en la carga y descarga de condensadores, se producirá un rizado en la tensión de salida, siendo mayor cuanto menor sea la frecuencia de entrada.

La tensión de salida elegida para el Tacómetro ha sido de 0.25 V/r.p.s. Para obtener una realimentación unitaria será necesario un amplificador de ganancia 4; este amplificador puede ser implementado por hardware o software en el algoritmo de simulación.

En la Figura 2 se muestra el sistema implementado.

### 4. FUZZY LOGIC TOOLBOX

El interface de usuario gráfico de *Fuzzy Logic Toolbox* (FLT) posee cinco herramientas básicas para el diseño de sistemas de inferencia borroso, de forma que los cambios en cualquiera de ellas afecta de forma automática a las demás (Figura 3):

- 1) *Editor del sistema de inferencia difuso (FIS Editor)*: aquí especificamos el número de variables de entrada y salida del sistema con sus nombres. Aunque FLT no limita el número de entradas, éste puede estar limitado por la memoria del ordenador.

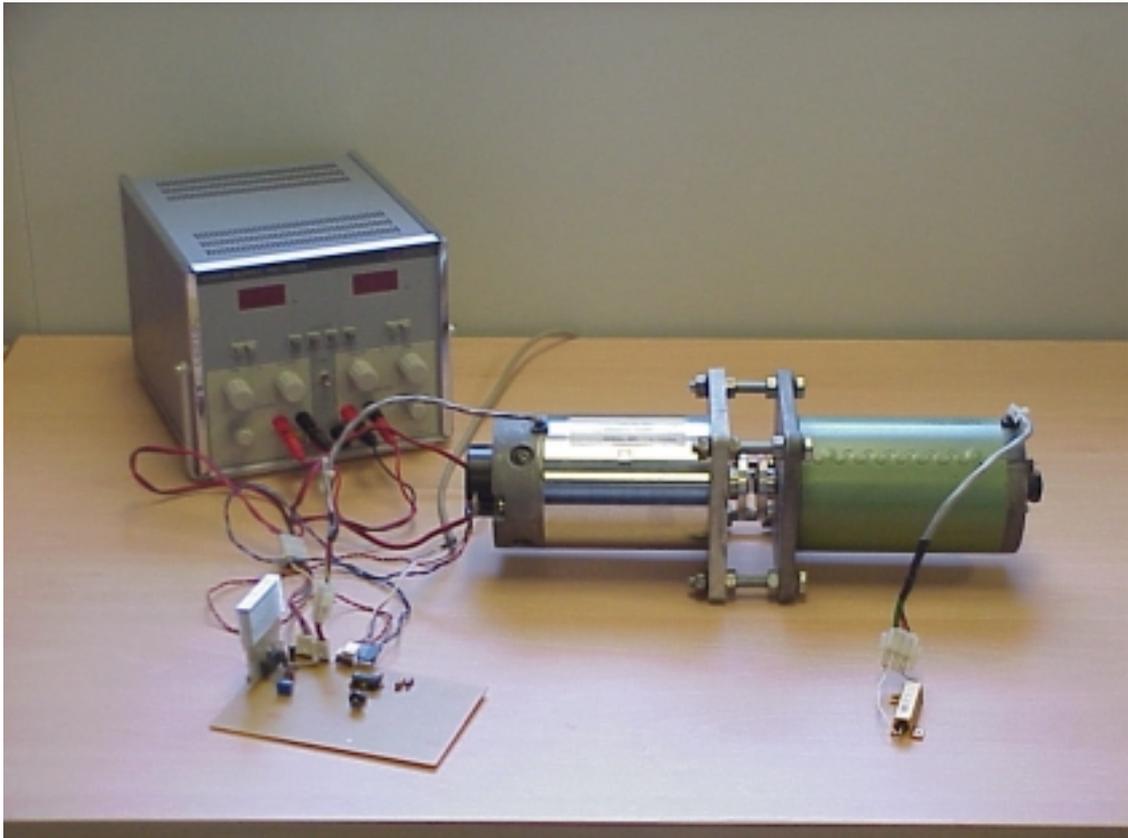


Figura 2. Sistema para prototipado rápido

- 2) *Editor de las funciones de pertenencia (Membership Function Editor)*: para definir las funciones de pertenencia asociadas a cada una de las variables de entrada y de salida.
- 3) *Editor de reglas (Rule Editor)*: para editar la lista de reglas que define el comportamiento del sistema.
- 4) *Visor de reglas (Rule Viewer)*: para observar los resultados, por ejemplo, qué reglas son activas o que influencia tiene cada función de pertenencia en los resultados.
- 5) *Visor de superficie (Surface Viewer)*: nos enseña como la salida depende de las entradas, mostrándonos un mapa de superficie.

Una vez que el sistema de inferencia borroso ha sido creado en FLT, se puede incorporar en Simulink. En la librería *Fuzzy Logic Toolbox* de Simulink se dispone del bloque Fuzzy Logic Controller, el cuál será usado en la construcción del modelo del sistema. En la Figura 4 se muestra el modelo en Simulink de un sistema de control que incorpora un PD-Borroso, observándose el bloque Fuzzy Logic Controller que dispone de una entrada y una salida, por lo que si al sistema de inferencia se le debe introducir más de una entrada será necesario disponer de un multiplexor a la entrada; en el caso de que el sistema de inferencia disponga de más de una salida se dispondrá de un demultiplexor a la salida.



Figura 3. Herramientas de Fuzzy Logic Toolbox

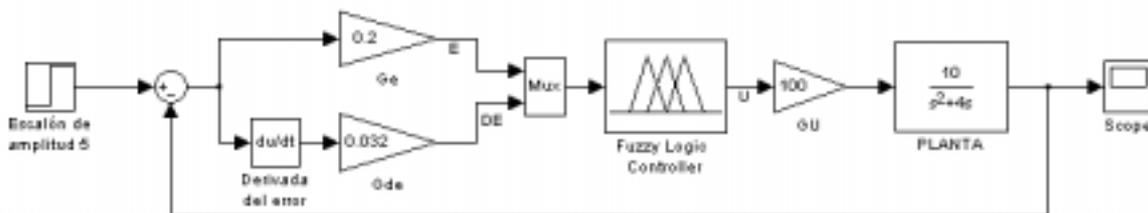


Figura 4. Ejemplo de modelo en Simulink con el bloque Fuzzy Logic Controller

## 5. REAL-TIME WINDOWS TARGET

*Real-Time Windows Target* (RTWT) es parte de una familia de productos software que se pueden usar para crear sistemas de control en tiempo real. En RTWT el entorno de desarrollo y el entorno en tiempo real usan el mismo procesador (*self-targeting system*), pudiéndose instalar en un PC bajo sistema operativo Windows. En definitiva, RTWT supone una solución para el prototipado y testeado de sistemas de control en tiempo real sobre PC.

RTWT permite ejecutar los modelos de Simulink de forma interactiva y en tiempo real con un PC. Interactuando con las tarjetas de E/S más populares ISA, PCI y PCMCIA, proporciona un entorno excelente para hacer prototipos con los diseños de Simulink de forma rápida y efectiva. Gracias a la capacidad de generación de código de *Real-Time Workshop* [4], esta herramienta ayuda a evitar el proceso de hacer código y depurarlo a mano.

RTWT requiere los siguientes productos:

- ✓ *MATLAB*: proporciona las herramientas de diseño y análisis para Simulink.

- ✓ *Simulink*: proporciona el entorno para modelar el sistema en forma de diagramas de bloques. Se puede usar RTWT con la mayoría de los bloques de Simulink excepto los que no corren en tiempo real y los siguientes: Display, To Workspace, To File, y Graph blocks salvo el Scope block. Además, se debe usar el algoritmo de integración con paso fijo (*fixed-step*). Las funciones S en código C son soportadas, pero no las funciones S en código M.
- ✓ *Real-Time Workshop (RTW)*: proporciona las utilidades necesarias para convertir el modelo de *Simulink* en código C y dispone de varios formatos de salida para optimizar el código para sistemas embebidos. Cuando se usan sistemas en tiempo continuo y se genera el código con *RTW*, se debe usar el algoritmo de integración de paso fijo.
- ✓ *Compilador en C*: crea el código ejecutable desde el código C generado por *RTW* y las funciones S en código C que se hayan generado. Se requiere la versión profesional de Microsoft Visual C/C++ o Watcom C/C++.

Para el diseño del sistema de control se pueden seguir los siguientes pasos:

- 1) Usar MATLAB y el FLT (o algún toolbox de control, como el *Control System Toolbox*) para diseñar y seleccionar los coeficientes del sistema para el controlador.
- 2) Usar los bloques de Simulink para modelar de forma gráfica el sistema físico.
- 3) Comprobar el comportamiento del modelo mediante una simulación en tiempo no real antes de crear la aplicación en tiempo real.
- 4) Crear la aplicación en tiempo real. Para ello se usará *RTW* que generará el código C desde el modelo creado en Simulink. Posteriormente el compilador en C creará el ejecutable que correrá con el Kernel de RTWT.
- 5) Correr la aplicación en tiempo real y visualizar y analizar los datos. Los datos obtenidos pueden ser enviados al *Workspace* de MATLAB y usar las funciones gráficas.

Una vez que se ha configurado de forma apropiada en Simulink ciertos parámetros, *RTW* y el compilador en C son transparentes para el diseñador puesto que no necesita modificar ningún parámetro de éstos, salvo que se quiera generar el código de forma diferente. En el modo externo, los cambios en parámetros de Simulink son automáticamente pasados a la aplicación en tiempo real.

En la librería *Real-Time Windows Target* de Simulink se dispone de los bloques típicos de los que disponen las tarjetas de adquisición: entradas/salidas analógicas y digitales, entrada de contador, entrada para encoder y otras entradas y salidas específicas para ciertas tarjetas. Naturalmente, dependiendo de la tarjeta utilizada se dispondrán de unos u otros recursos.

Para las pruebas en tiempo real se ha utilizado la tarjeta de adquisición PCL-818L. En la Figura 5 se muestra un modelo en Simulink de un controlador PID-Borroso donde se han incorporado los drivers para acceder a la tarjeta de adquisición.

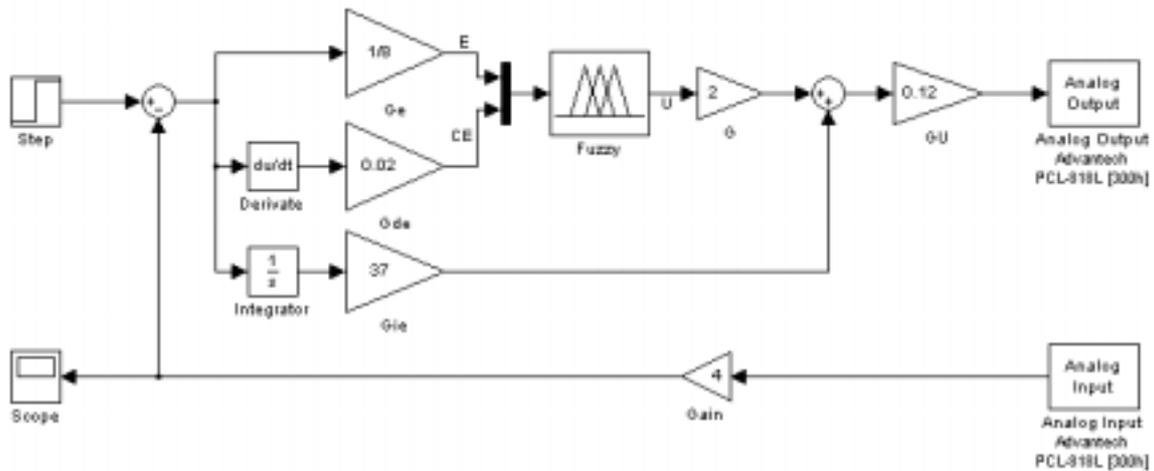


Figura 5. PID-Borroso para simulación en tiempo real

## 6. IDENTIFICACIÓN DE LA PLANTA

Para obtener un algoritmo aproximado del controlador es más rápido realizar simulaciones en tiempo no real con un modelo matemático de la planta.

El término *identificación de procesos* [5] podría ser definido como el conjunto de estudios, teorías y algoritmos que permiten obtener la estructura y los parámetros de un modelo matemático. En sistemas cuyos elementos tienen un comportamiento conocido es posible, en función de las ecuaciones físicas que ligan sus variables, obtener relaciones matemáticas, llegando de este modo a los conocidos como modelos de primeros principios (modelado). En la mayoría de los casos estas ecuaciones no son físicamente deducibles y la identificación exige de un conjunto de experiencias sobre el sistema que permitan la determinación indirecta de las ecuaciones o lo que es igual, de su estructura y de sus parámetros.

La señal más simple que puede utilizarse para la identificación es, sin duda, la función escalón. Un escalón ideal es una señal cuyo tiempo de crecimiento inicial es cero. Resulta obvio que tal señal no puede ser realizada físicamente, pues sería necesaria una energía infinita. Sin embargo, si el tiempo de crecimiento inicial es varias veces más pequeño que la constante de tiempo más pequeña del sistema, el error que se introduce en la identificación puede considerarse despreciable. Con la señal escalón se obtienen modelos sencillos suficientemente exactos, sobre todo en los casos de procesos monovariabes simples y poco perturbados. La respuesta de un proceso tecnológico a la señal escalón puede aproximarse, en muchos casos, mediante un modelo de primer orden con o sin retardo (Ecuación 1):

$$G_p(s) = \frac{K \cdot e^{-T \cdot s}}{(\tau \cdot s + 1)} \quad (\text{Ecuación 1})$$

donde K es la ganancia del sistema, T el tiempo de retardo y  $\tau$  la constante de tiempo.

Para la planta construida se ha aplicado diferentes escalones para diferentes cargas eléctricas, obteniendo la función de transferencia a partir de los resultados obtenidos. En la Figura 6 se

muestra un ejemplo de respuesta para un escalón de 15 V y 10  $\Omega$  de carga eléctrica. A partir de la respuesta y mediante un programa creado en MATLAB se obtiene de forma directa la función de transferencia.

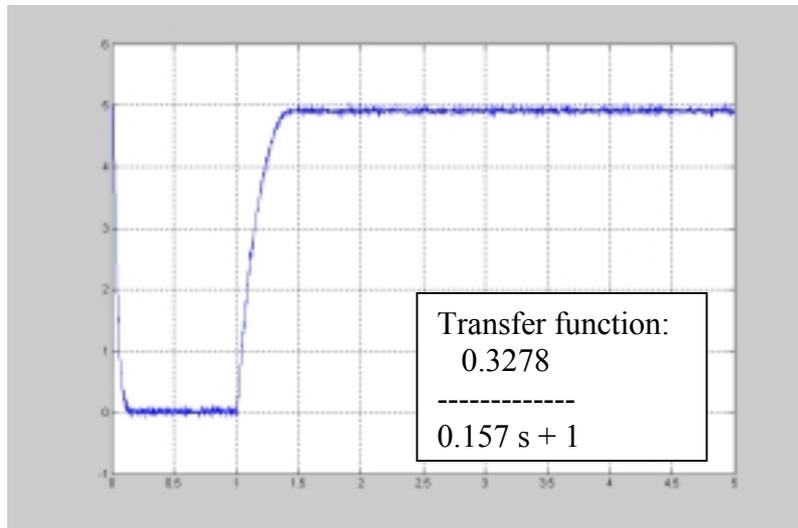


Figura 6. Respuesta de la planta y función de transferencia obtenida

## 7. COMPARATIVA ENTRE PID-CLÁSICO Y PID-BORROSO

Se han diseñado diferentes controladores borrosos para la planta creada y se han comparado sus prestaciones con los homólogos clásicos [6][7][8].

Entre las diferentes pruebas realizadas se presentan aquí los resultados obtenidos para el PID-Clásico y el PID-Borroso. En la Figura 5 se presenta el modelo para el PID-Borroso y en la Figura 7 el modelo para el PID-Clásico.

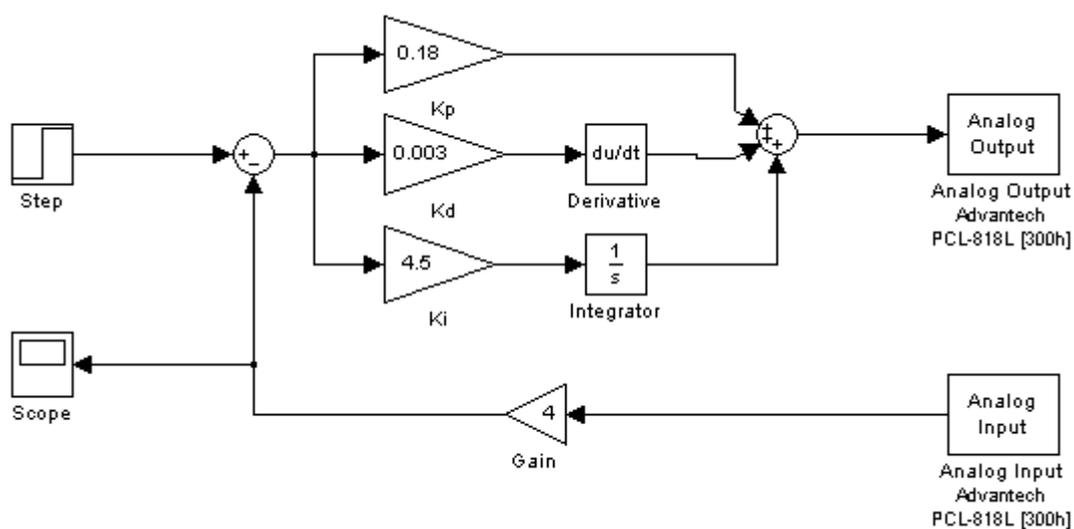
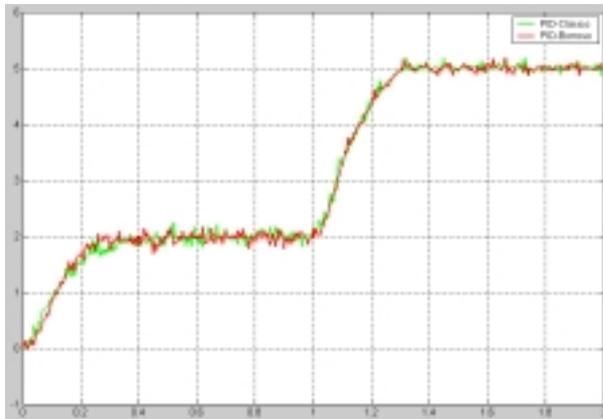
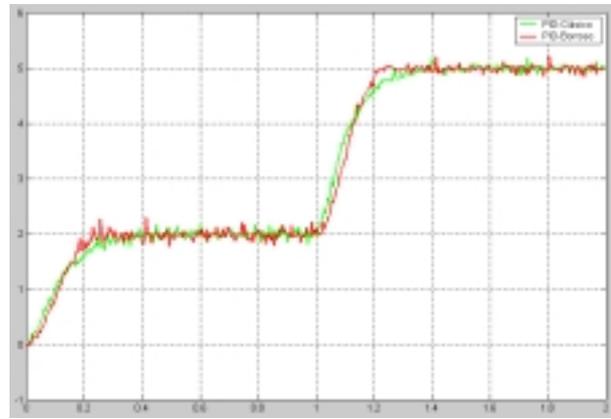


Figura 7. PID-Clásico para simulación en tiempo real

En las Figuras 8 y 9 se presentan las respuestas para cambios en la consigna de 0 a 2 r.p.s. y de 2 r.p.s. a 5 r.p.s. Se observa la mejor adaptación que presenta el control borroso ante cambios en la consigna y en la carga.



*Figura 8. Respuesta obtenida para una carga eléctrica de 10 ohmios*



*Figura 9. Respuesta obtenida sin carga eléctrica*

## 8. CONCLUSIONES

En este artículo se ha presentado el diseño de una planta para desarrollar prácticas de control. Se puede concluir:

- ✓ Fácil implementación de la planta con un coste mínimo.
- ✓ En el proceso de diseño del controlador el alumno deberá realizar los pasos típicos de control: identificación, simulaciones en tiempo no real, simulaciones en tiempo real.
- ✓ Aunque a la planta se le puede aplicar cualquiera de las técnicas de control que se desee, se ha mostrado la posibilidad del control borroso.

## 9. BIBLIOGRAFÍA

- [1] Fuzzy Logic Toolbox User's Guide. © COPYRIGHT 1995 - 2001 by The MathWorks, Inc.
- [2] Using Simulink. © COPYRIGHT 1990 - 2001 by The MathWorks, Inc.
- [3] Real-Time Windows Target User's Guide. © COPYRIGHT 1999-2001 by The MathWorks, Inc.
- [4] Real-Time Workshop User's Guide. © COPYRIGHT 1994 - 2001 by The MathWorks, Inc.
- [5] Identificación y Control Adaptativo. Alberto Aguado Behar y Miguel Martínez Iranzo. Prentice-Hall, 2003.
- [6] Jan Jantzen, "Design of fuzzy controllers", Technical University of Denmark, Department of Automation. Tech. Report no 98-e 864, 30 Sep 1999
- [7] Jan Jantzen, "Tuning of fuzzy PID controllers", Technical University of Denmark, Department of Automation. Tech. Report no 98-H 871, 30 Sep 1998
- [8] M. Mizumoto, "Realization of PID controls by fuzzy control methods", Fuzzy Sets Syst., vol. 70, pp. 171-182, 1995.