

DISEÑO DE UNA PRÁCTICA DE CONTROL BORROSO PARA UN LABORATORIO DOCENTE

L.M.Bergasa, D.Lillo, R.F.Flores, E. Bueno, L. Boquete, M.A. Sotelo

Departamento de Electrónica de la Universidad de Alcalá
Campus Universitario s/n. 28871 Alcalá de Henares (MADRID)

Tfno: (91) 8854810-15 Fax: (91) 8854804-99

E-mail: bergasa@depeca.alcala.es

RESUMEN

En la presente comunicación se describe el diseño de una práctica de control borroso para su impartición en un laboratorio docente. Dicha práctica se realiza en la asignatura de Control Neuronal y Borroso ubicada como materia optativa en 5º curso de Ingeniería Electrónica. La citada asignatura tiene una carga docente de 6 créditos de los cuales 1.5 créditos se dedican a prácticas de laboratorio en donde el alumno puede experimentar con los conceptos teóricos explicados en clase.

1. INTRODUCCIÓN

Para la impartición de prácticas en la asignatura de Control Neuronal y Borroso se ha fabricado un prototipo experimental consistente en un péndulo giratorio sobre el que se realiza un control de posición borroso que se ejecuta en un PC [1][2][3]. Para llevar a cabo el control de dicha planta se ha diseñado un interfaz hardware con el PC. Con estos elementos se propone al alumno el diseño de un controlador borroso empleando la herramienta de MATLAB "Fuzzy Logic Toolbox"[4].

Para afrontar este laboratorio al alumno se le suponen conocimientos previos en el manejo de MATLAB y SIMULINK, los cuales les han sido aportados en asignaturas anteriores de control electrónico. En cuanto al manejo de las herramientas específicas de control neuronal y borroso, previamente a la realización de esta práctica el alumno realiza 2 prácticas dirigidas en donde se familiariza con el manejo de las herramientas de MATLAB "Fuzzy Logic" y "Neural Network". La primera lleva por título: "Identificación neuronal de un motor de continua" y la segunda "Simulación de un controlador borroso para un péndulo invertido en una dimensión". De esta forma, cuando el alumno se enfrenta a la práctica que se propone ya tiene los conocimientos necesarios para abordarla con éxito.

2. DISEÑO DE LA PLANTA

Se ha fabricado una maqueta experimental en metacrilato consistente en un péndulo giratorio. La planta está formada por una varilla con una junta libre en un extremo y sujeta a una pequeña plataforma mediante dos muelles en equilibrio, que son los que le proporcionan la flexibilidad a la junta. La plataforma sobre la que se encuentra la varilla es giratoria. Los giros de la misma se consiguen gracias a una serie de engranajes, que conforman una reductora conectada con un motor de continua, ubicado en una plataforma fija más grande que soporta todo el sistema [5][6].

La varilla oscilará ante las posibles perturbaciones introducidas externamente. La misión del controlador será evitar las oscilaciones de la varilla ante las perturbaciones manteniendo ésta en su punto de equilibrio. En la figura 1 se aprecian dos vistas del sistema.

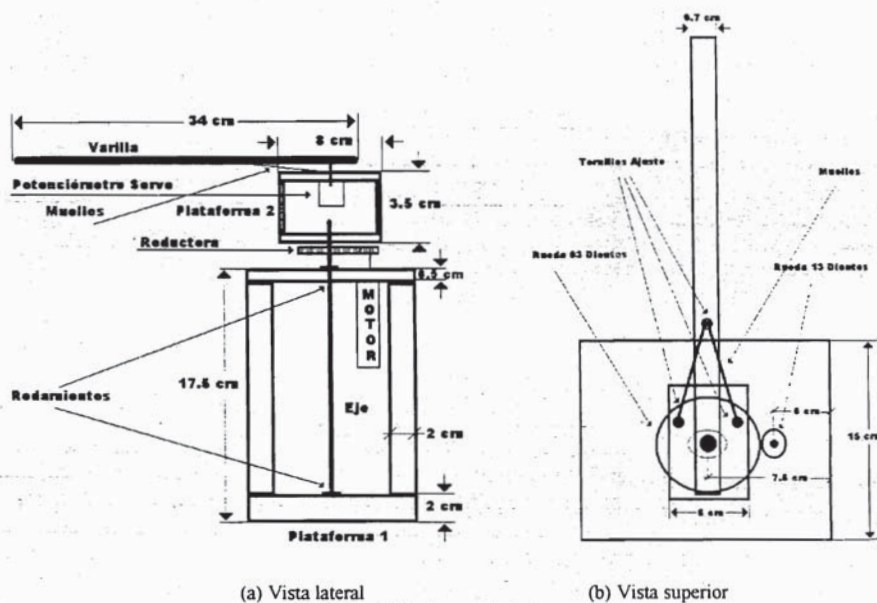


Figura 1. Vistas de la planta diseñada

3. DISEÑO DE LA TARJETA DE INTERFAZ HARDWARE

Se ha diseñado una interfaz hardware para comunicar la planta con el puerto paralelo del PC. Mediante esta interfaz se tiene acceso desde el PC a los sensores ubicados en la planta y se pueden mandar consignas de velocidad al motor de continua, como se puede ver en la figura 2.

Se ha realizado la excitación del motor mediante un puente en H y un C.I. específico para excitación de puentes como es el HIP4082. La generación de las señales PWM necesarias son

conseguidas mediante un contador y un comparador, de forma que se puede indicar desde el PC el valor del ciclo de trabajo codificado. Las señales procedentes de los sensores son acondicionadas para ser enviadas por el puerto paralelo al PC. Se ha colocado un potenciómetro de precisión solidario a la junta flexible, que junto a un ADC nos proporciona el ángulo codificado de la varilla en cada momento. Además, el motor de continua posee un encoder óptico incremental que junto a un contador permite conocer la velocidad del mismo.

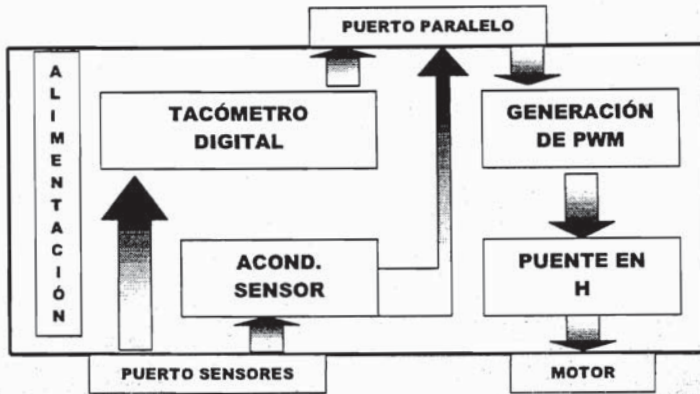


Figura 2. Interfaz hardware con el PC

El protocolo de comunicaciones de la placa con el puerto paralelo del PC se realiza según las conexiones que se presentan en las tablas 1,2 y 3. El bus de datos es bidireccional, de forma que cuando se programa como salida, en las líneas D0..D7 se manda el valor de consigna codificada en 8 bits (Vcodi). Cuando se programa como entrada, en el bus de datos se obtiene el valor codificado en 8 bits del ángulo de la varilla (Poti). En los 5 bits del bus de estado se obtiene el valor codificado de la velocidad de giro del eje del motor (Wcod). Dentro del bus de control, la señal GIRO es una señal de entrada que indica el sentido de giro del motor, #RD es la señal de lectura del ADC que se debe activar a nivel bajo cada vez que se desee leer un dato, #WR es la señal de inicio de conversión del ADC, también activa a nivel bajo, y por último LATCH es la señal de carga de la consigna para la generación de la señal PWM, activa en flanco de subida.

BUS DE DATOS

PIN	2	3	4	5	6	7	8	9
Etiqueta	D0	D1	D2	D3	D4	D5	D6	D7
Salida	Vcod0	Vcod1	Vcod2	Vcod3	Vcod4	Vcod5	Vcod6	Vcod7
Entrada	Pot0	Pot1	Pot2	Pot3	Pot4	Pot5	Pot6	Pot7

Tabla 1. Conexiones del bus de datos del puerto paralelo

BUS DE ESTADO

PIN	11	10	12	13	15	X	X	X
Etiqueta	#S7	S6	S5	S4	S3	X	X	X
Entrada	Wcod4	Wcod3	Wcod2	Wcod1	Wcod0	X	X	X

Tabla 2. Conexiones del bus de estado del puerto paralelo

BUS DE CONTROL				
PIN	17	16	14	1
Etiqueta	#C3	C2	#C1	#C0
Entrada	GIRO	#RD	#WR	LATCH

Tabla 3. Conexiones del bus de control del puerto paralelo

4. PRÁCTICA A REALIZAR

El alumno dispone de la planta y del interfaz hardware con el PC y se le plantea el diseño de un controlador borroso que establezca las oscilaciones de la varilla ante perturbaciones externas en un tiempo mínimo. El esquema de control se muestra en la figura 3.

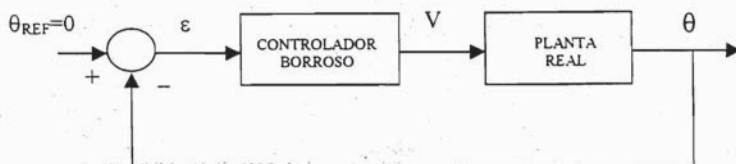


Figura 3. Diagrama de bloques del controlador a diseñar

θ indica el ángulo de la varilla respecto a su posición de equilibrio, ϵ el error de posición en ángulos y V la tensión de excitación del motor.

Para el diseño del controlador inicialmente se realizará una simulación del mismo que permita ajustar sus parámetros. Posteriormente se aplicará el controlador diseñado sobre la planta real y se comprobará las diferencias entre el control simulado y el real, realizando las modificaciones oportunas sobre el controlador real para que el control sea adecuado. Para la realización de estos objetivos se proponen 4 fases: en la primera se realizará la comunicación con el interfaz hardware, en la segunda el alumno deberá hacer una identificación neuronal de la planta, en la tercera diseñará una simulación del controlador y por último implementará el controlador sobre la planta real.

4.1. Comunicación con el interfaz hardware

Inicialmente el alumno deberá comunicarse con el interfaz hardware, a través del puerto paralelo del PC, según el protocolo de comunicaciones definido anteriormente. Para ello realizará un programa en C, ejecutable bajo DOS, que permita hacer las siguientes operaciones:

- Envío de distintas señales de consigna al motor DC (triangular, cuadrada, senoidal, procedente de un fichero)
- Lectura del ángulo de la varilla y almacenamiento de los datos obtenidos en un fichero.
- Posibilidad de usar distintos períodos de muestreo en las adquisiciones.

4.2. Identificación neuronal de la planta

Para poder realizar simulaciones del controlador borroso es necesario disponer de un modelo de la planta. Se recurre a técnicas neuronales de identificación como aplicación práctica de los conceptos explicados en teoría. Para ello el alumno deberá obtener la respuesta de la planta ante distintos estímulos (señales rampa, cuadradas, senoidales, etc) con el software desarrollado en la fase anterior. Con los ficheros de datos obtenidos de la planta real se deberá identificar la misma utilizando la herramienta de MATLAB "Neural Network Toolbox".

El alumno deberá emplear diferentes arquitecturas de redes neuronales para obtener el modelo de la planta (redes Elmann, RBFs, Madaline, etc), entrenando la red con una señal triangular y probando la bondad de la misma (capacidad de generalizar) con una señal cuadrada.

En la memoria se deberá entregar como resultado la topología de red que mejor aproxime al sistema real, es decir, que minimice el error cuadrático medio (SSE) entre el modelo neuronal y el sistema real.

4.3. Simulación del controlador

Una vez identificada la planta se diseñará el controlador borroso mediante las herramientas gráficas de "Fuzzy Logic Toolbox" de MATLAB, definiendo las variables de entrada y de salida del controlador, las funciones de pertenencia de dichas variables y las reglas de actuación. Esta herramienta genera como salida una matriz de datos con extensión ".fis" que implementa el controlador.

A continuación se realizará un diagrama completo del lazo de control bajo SIMULINK. Se debe añadir el bloque del modelo de la planta identificado con técnicas neuronales y el bloque del controlador borroso "Fuzzy Logic Controller", donde se le pasa como parámetro "fismat". Para asignar la matriz "controlador.fis" a la variable "fismat" hay que introducir en el fichero de inicialización de SIMULINK la orden: `fismat=readfis('controlador.fis')`.

Se introducirá un bloque que simule una perturbación externa y se comprobará la bondad del controlador diseñado ante dicha perturbación analizando la señal de posición de la varilla de manera que se minimice la amortiguación de la misma.

En la memoria de la práctica se deberá entregar el esquema de SIMULINK dando como resultados las gráficas de ángulo de la varilla ante distintas perturbaciones. Opcionalmente se puede diseñar un entorno gráfico para representar una animación de la simulación.

4.4. Controlador sobre la planta real

Completada la fase de simulación, para la implementación real del controlador se utilizará la herramienta "Fuzzy Inference Engine" suministrada por MATLAB, con la que se puede realizar programas ejecutables totalmente autónomos. Esta herramienta lee una matriz FIS y un archivo de datos de entrada y da la salida defuzzificada del sistema borroso de cada uno de los vectores de entrada. Se suministra al alumno una plantilla en código C basada en la herramienta comentada, que deberá personalizar para implementar su controlador. En la figura 4 se muestra un organigrama de la plantilla, donde aparecen las distintas funciones que

la componen. El alumno debe completar la declaración de variables y las funciones de leer, escribir y de ajuste del periodo de muestreo. Una vez desarrollado el programa ejecutable bajo DOS se compila y se ejecuta pasando como parámetro la matriz FIS. Posteriormente se comprobará la bondad del controlador en condiciones reales, modificando los parámetros oportunos, siguiendo los pasos anteriormente descritos, hasta obtener un controlador adecuado.

El alumno deberá mostrar el funcionamiento de su controlador al profesor y deberá presentar en la memoria las gráficas de la posición de la varilla real ante varias perturbaciones de entrada. Asimismo realizará un estudio comparativo entre los resultados obtenidos de la simulación y de la planta real.

5. EVALUACIÓN

El alumno deberá entregar una memoria del trabajo realizado donde se contemple el desarrollo de todos los puntos requeridos. La evaluación se realizará en función de la memoria entregada y del seguimiento realizado por el profesor en las clases de laboratorio. La experiencia en el primer año de inclusión de esta práctica ha sido muy positiva, mostrando los alumnos un gran interés por la misma debido a la gran cantidad de disciplinas que manejan (teoría de control borroso, identificación neuronal, simulación con SIMULINK, creación de un entorno gráfico con MATLAB, programación en C, diseño de un controlador real) y a que se puede pasar de una concepción teórica de un controlador a su aplicación práctica sobre una planta real en un corto tiempo.

6.- BIBLIOGRAFÍA

- [1] Lofti A. Zadeh. "Fuzzy sets and applications". Ed. John Wiley and Sons. 1987.
- [2] John Yen, Reza Langari, Lofti A. Zadeh. "Industrial applications of fuzzy logic and intelligent systems". Ed. IEEE Press. 1995.
- [3] Robert J. Marks. "Fuzzy logic technology and applications". IEEE Technology Update Series. 1994.
- [4] "Fuzzy Logic Toolbox User's Guide" de MATLAB
- [5] J. Holgado. "Controladores en lógica borrosa". Ed. Universidad de Cádiz. 1995.
- [6] J. Holgado. "Estudios monográficos de automática avanzada". Ed. Universidad de Cádiz. 1995.



Figura 4. Plantilla del código C