Implementación de controladores en Arduino mediante Simulink

Eneko Lerma, Robert Griñó, Ramon Costa-Castelló
Departamento de Ingeniería de Sistemas, Automàtica i Informàtica Industrial (ESAII)
Universitat Politècnica de Catalunya (UPC)
e-mail: eneko.lerma@estudiant.upc.edu, ramon.costa@upc.edu, roberto.grino@upc.edu
Carlos Sanchis
Mathworks, e-mail: Carlos.Sanchis@mathworks.es

Resumen

En este trabajo se presenta una plataforma basada en MATLAB y Arduino para la realización de prácticas de control digital. El trabajo describe los componentes hardware utilizados, los elementos de MATLAB/Simulink y muestra algunos de los resultados preliminares que se están obteniendo en las plataformas experimentales de las que se dispone en el departamento de ESAII de la UPC.

1. Introducción

En el ámbito de la docencia en ingeniería la realización de prácticas experimentales continua siendo de vital importancia para la correcta formación de los estudiantes. Desafortunadamente, la adquisición de equipamiento que permita realizar prácticas experimentales presenta un coste económico que no es siempre fácil de asumir por los centros docentes.

La aparición de plataformas hardware de bajo coste tipo Arduino y Rasberry Pi ha favorecido el desarrollo de entornos de prácticas que pueden ser utilizadas en casi todos los entornos y en particular en el ámbito del control automático [6, 2, 1, 7, 5]. Este tipo de dispositivos pueden ser programados con sus propios entornos de simulación o bien usando entornos de generación automático de código como los incorporados en MATLAB/Simulink[1, 6]. Todo ello ha reducido substancialmente los costes de los equipos de captura, generación y procesado, así como el coste en tiempo necesario para preparar unas prácticas.

En este trabajo se presenta los desarrollos, y análisis, que se han realizado con el objetivo de substituir la plataforma que actualmente se utiliza en los laboratorios del departamento de ESAII en la Escuela Técnica Superior de Ingeniería Industrial de Barcelona (ETSEIB) por una de bajo coste basada en una placa Arduino y entorno MATLAB/-Simulink.

El trabajo se organiza de la siguiente forma, en la sección 2 se presenta en entorno de prácticas actual, la sección 3 analiza la prestaciones de tiempo real ofrecidas por la plataforma propuesta, la sección 4 describe las prácticas experimentales que se realizan, la sección 5 muestra algunos resultados preliminares, y finalmente en la sección 7 se presentan algunas conclusiones.

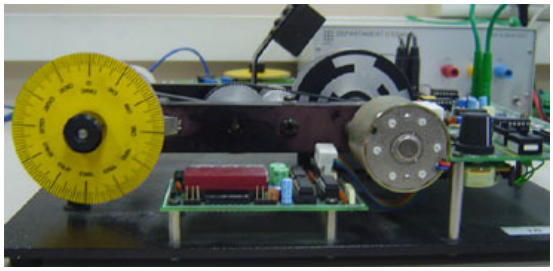

Figura 1: Vista del Prototipo de Prácticas

2. Descripción general del sistema actual

El laboratorio de automática de la ETSEIB está dotado de 12 servosistemas de posición angular (*LJ Technical Systems*) (Figura 1), cada uno de ellos conectado a un PC equipado con tarjetas AD/DA. La supervisión y el control se realiza mediante la librería de tiempo real para *MATLAB* (*Real-time Widows target*. La interfase ha sido desarrollada en el departamento de ESAII (sección ETSEIB) y permite al usuario especificar el periodo de muestreo, los parámetros de los diferentes controladores, y el tipo de entrada que se genera.

El coste del sistema de las tarjetas de entrada salida es elevado y se trata de dispositivos que utilizan el bus PCI. Este es cada vez menos común cosa que complica la actualización de los equipos. Este es uno de los principales motivos por los que se inicia el proyecto descrito en este proyecto. Como objetivo principal se pretende analizar la viabilidad de substituir los sistemas basados en tarjetas AD/DA por sistemas basado en dispositivos Arduino.

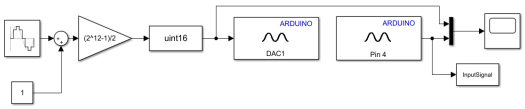

Figura 2: Diagrama de bloques de Simulink para el análisis de la variabilidad en el período de muestreo.

3. Variabilidad en el período de muestreo

Como paso previo a estudiar la planta o a diseñar el controlador, hay que analizar las posibles limitaciones que pueden aparecer en la transferencia de datos entre la placa de control Arduino y la hoja de simulación en Simulink. Este aspecto es relevante puesto que se pretenden utilizar los bloques Scope de Simulink como elemento de graficación de las señales relevantes (referencia, salida y señal de control) del sistema en lazo cerrado real. A tal efecto, se distinguen dos periodos de muestreo diferentes: el periodo de muestreo de graficación T_g, de valor pequeño, para que las representaciones gráficas en los Scopes sean lo más fiel posibles a las señales reales de tiempo continuo y el periodo de nuestreo del sistema de control digital T_s que será un múltiplo entero de T_g a fin de simplificar la codificación.

A fin de analizar la limitación en la transferencia de datos se selecciona una señal sinusoidal de 10 Hz (frecuencia fundamental mayor que las que pueden aparecer en el sistema real) y se verifica el comportamiento con distintos períodos de graficación T_g.

Para ello, se empieza con un T_g de 0.001 s y se verifica si hay pérdida de muestras en los Scopes. Si las hay, se va incrementado el período de graficación hasta que no se produce pérdida de información. En la Fig. 2 se muestra el esquema de bloques Simulink para llevar a cabo el experimento: la señal sinusoidal se saca por el conversor D/A y la salida física de este conversor se conecta con la entrada física del conversor A/D, comparándose, de forma gráfica, la lectura digital con la señal original.

Como se ha mencionado anteriormente, se utiliza una placa Arduino Due que ofrece una resolución de 12 bits en el conversor A/D. No obstante, el bloque *analog input* de la librería Simulink solamente ofrece una resolución de 10 bits, con lo que la señal de entrada y salida tendrían distinta cuantificación y distintos valores para una misma señal original.

Para que esto no ocurra, se ha cambiado la codificación interna del bloque *analog input* para poder aprovechar la resolución de 12 bits que ofrece el hardware. La modificación efectuada consiste en llamar, en el código asociado al bloque, a la función *analogReadResolution*, fijando 12 bits, antes de usar *analogRead* si la placa en uso es un Arduino Due. De esta forma, los conversores A/D y D/A operan a 12 bits y la cuantificación de las señales de entrada y salida es la misma.

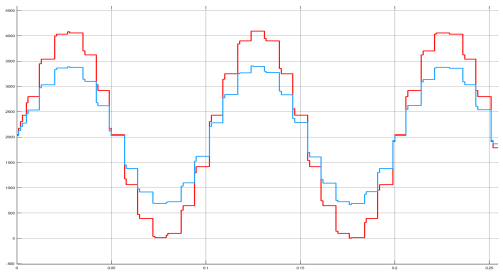

Figura 3: Señales obtenidas con $T_g = 1$ ms.

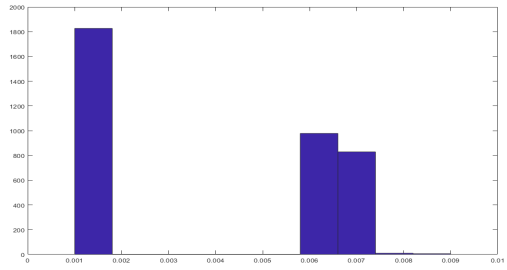

Figura 4: Variación del período de muestreo para $T_g = 1$ ms.

La Fig. 3 muestra las señales obtenidas (en el Scope de la Fig. 2) para el caso de $T_g = 1$ ms. Como se puede observar se producen pérdidas de muestras no asegurándose un período de graficación constante. Por otra parte, como se puede observar, los rangos de las señales de entrada y salida no son iguales. Esto ocurre porque el rango de tensiones del conversor D/A en el uC de la placa Arduino Due no es de 0 V a 3.3 V sino de 0.5 V a 2.82V. Esta característica se corregirá en el diseño del sistema en lazo cerrado.

Para poder hacer un análisis cuantitativo de la variabilidad del período T_g, se ha utilizado el código MatLab que aparece a continuación con el cual se obtiene un histograma representando la distribución de períodos ocurridos en un tiempo de experimentación finito.

```
h=InputSignal.time;
h=diff(h);
hmean=h-mean(h);
mean(h);
hist(h)
```

La Fig. 4 muestra que, aproximadamente, la mitad de los períodos corresponden a $T_g = 1$ ms pero la otra mitad se reparten entre 6 y 8 ms. En consecuencia, no se puede seleccionar el valor de $T_g = 1$ ms como período de graficación y debe incrementarse. Tras unas cuantas iteraciones se selecciona como período de graficación $T_g = 5$ ms puesto que, para este valor, la totalidad de los períodos medidos es 5 ms excepto un pequeño *jitter*, ver Fig. 5.

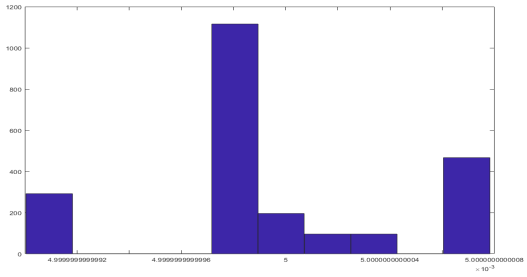

Figura 5: Variación del período de muestreo para $T_g = 5$ ms.

La Fig. 5 muestra las señales temporales obtenidas (en el Scope de la Fig. 2) para el caso de $T_g = 5$ ms. En ellas puede observarse que el período de graficación se mantiene constante no perdiéndose muestras por problemas de comunicación.

4. Diseño del acondicionador de señales

La Fig. 7 muestra una fotografía del prototipo de tarjeta de acondicionamiento de señales que se ha diseñado y construido para efectuar las adaptaciones de rango (ganancia y *offset*), y si conviene un filtrado pasa-bajos, entre la placa de control Arduino Due y la planta.

El rango de las señales de E/S de la placa controladora Arduino Due, tanto para señales analógicas como digitales, es 0 V a 3.3 V. No obstante, su conversor D/A no aprovecha todo este rango de tensiones. Para determinar su rango efectivo se

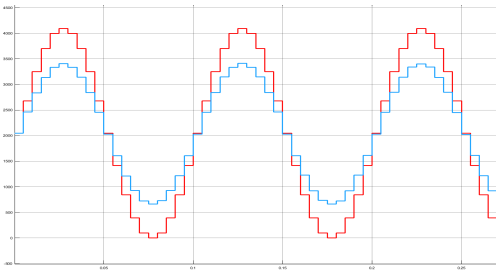

Figura 6: Señales obtenidas con $T_g = 5$ ms.

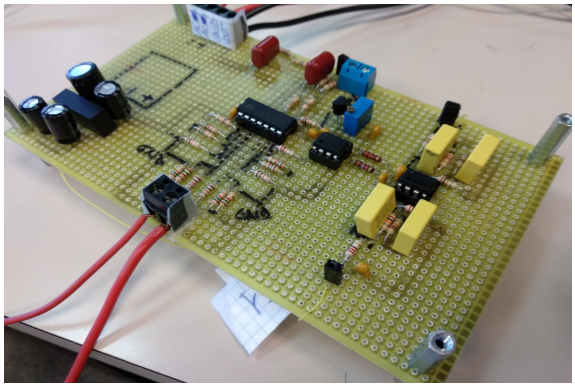

Figura 7: Prototipo de la tarjeta de acondicionamiento de señales de entrada y salida.

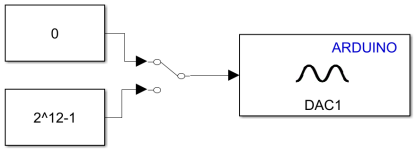

Figura 8: Diagrama de bloques de Simulink para el análisis del rango de tesiones de salida del conversor D/A.

utiliza el diagrama de bloques que se muestra en la Fig. 8 que permite obtener el mínimo y el máximo valor de salida del conversor D/A.

Como se se observa en la Fig. 9, la máxima tensión es 2.83 V y la mínima tensión es 0.58 V. Estos valores son los que se tienen en cuenta en el diseño del adaptador de niveles que llevará las señales al rango -5 V a 5 V, o -10 V a 10 V, en función de las características de la planta.

Para efectuar este cambio de rango, las señales deben experimentar un cambio de ganancia y una compensación de valor medio (*offset*) tanto en los canales de medida (conversión A/D) como en el canal de la señal de control (conversión D/A). Estas dos acciones se realizan de forma separada (en cascada) para reducir el acoplamiento entre las dos acciones y facilitar el análisis por parte de los alumnos.

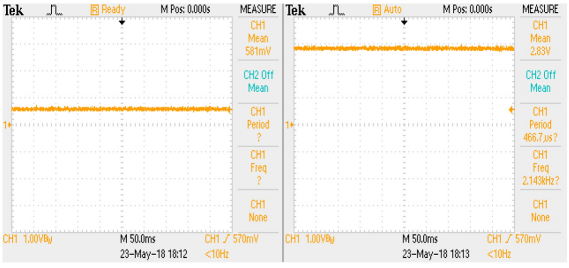

Figura 9: Mínima y máxima tensión obtenida mediante el DAC

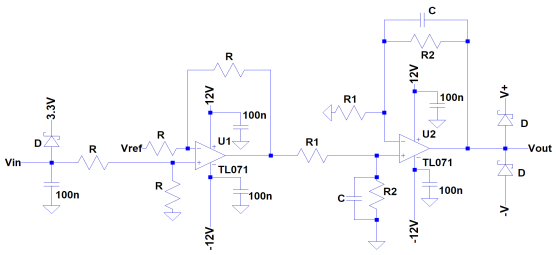

Figura 10: Circuito de acondicionamiento de señales para la salida del conversor D/A.

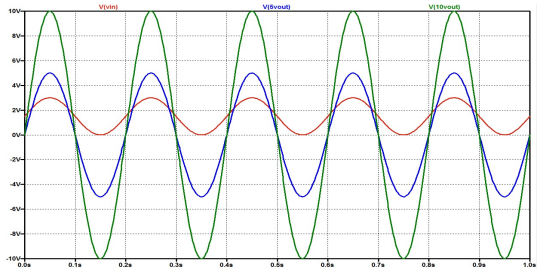

Figura 11: Simulación temporal del circuito para salidas en el rango 5 V y 10 V.

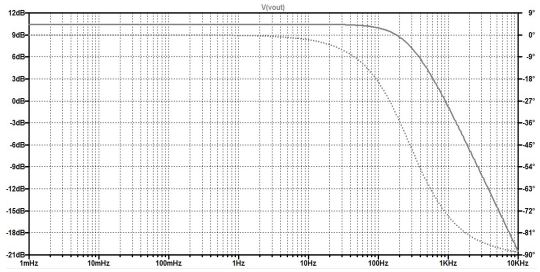

Figura 12: Diagrama de bode de la respuesta frecuencial del adaptador de señales de salida.

4.1. Canales de salida

Las señales de entrada de las plantas son bipolares, mientras que la señal del conversor D/A del Arduino Due es unipolar. Por tanto, a fin de acondicionarla, primero se le resta el *offset* y luego se amplifica su amplitud. Tal como muestra la Fig. 10, para desplazar la señal, mediante un amplificador operacional, se usa un circuito de ganancia unitaria con una tensión de 1.5 V en laentrada inversora. La amplitud de la señal se cambia con la segunda etapa del circuito que tiene una ganancia $G_2 = \frac{R2}{R1}$.

En concreto, puesto que la planta[1] presenta un rango de entrada de ±5 V se han escogido los valores $R_2 = 10$ kΩ y $R_1 = 3$ kΩ.

La relación entre la tensión de salida y la diferencia de las entradas es

$$V_{\text{out}} = \frac{R_2}{R_1(1 + R_2 C s)}(V_{\text{in}} - V_{\text{ref}}) \qquad (1)$$

Como se observa en la Fig. 10, en la segunda etapa hay dos condensadores que permiten filtrar pasa-bajos. La frecuencia de corte (en Hz), fijada en el diseño en 300 Hz, de este filtro es

$$f_c = \frac{1}{2\pi R_2 C}. \qquad (2)$$

Las Figs. 11 y 12 muestran, respectivamente y para el circuito de la Fig. 10, las respuestas temporales (para salida 5 V, señal azul, y salida 10 V, señal verde) para una señal sinusoidal en la salida del conversor D/A (señal roja), y la respuesta frecuencial en el diagrama de Bode.

4.2. Canales de entrada

En el caso de las señales de realimentación (entrada a la placa de control Arduino Due), ver Fig. 13,la primera etapa efectúa la reducción de ganancia y la segunda la corrección del *offset*. Como la salida de las señales de la planta está en el mismo rango que las señales de entrada (±5 V) la ganancia de la primera etapa es la inversa del caso de los canales de salida de la placa de control. Por tanto, se mantendrán los mismos valores de resistencias pero cambiando su orden. En el caso de la segunda etapa, se sumará una tensión continua de 1.5 V para hacer que la señal en la entrada del conversor A/D sea unipolar centrada en 1.5 V.

Como se observa en la Fig. 13, en la salida de la segunda etapa del circuito acondicionador se ha añadido un filtro pasa-bajos con una frecuencia de corte muy alta. El condensador de este filtro tiene como función ayudar a que la operación del conversor A/D sea mejor al eliminar los efectos del multiplexor interno. Además, el filtroo aporta un cierto comportamiento de filtro antialiasing. También se ha añadido un diodo Schottky conectado a

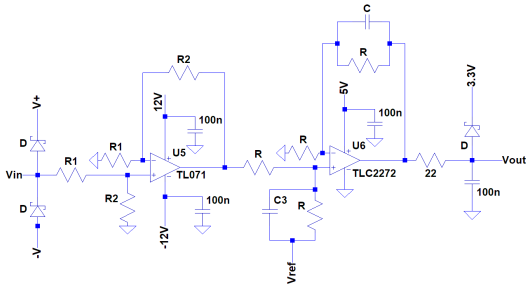

Figura 13: Circuito de acondicionamiento de señales para la entrada del conversor A/D.

[1]En el caso de un rango de entrada de ±10 V podrían escogerse los valores $R_2 = 20$ kΩ y $R_1 = 3$ kΩ.

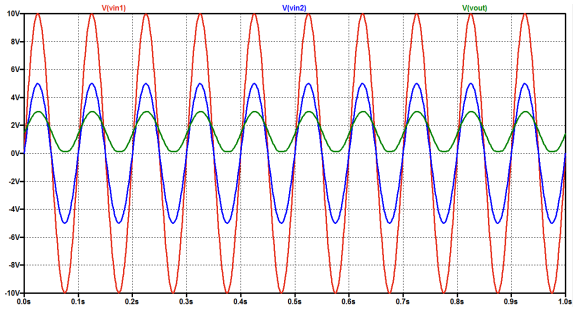

Figura 14: Simulación temporal del circuito para salida en el rango 3,3 V y entradas en los rangos de 5 V y de 10 V.

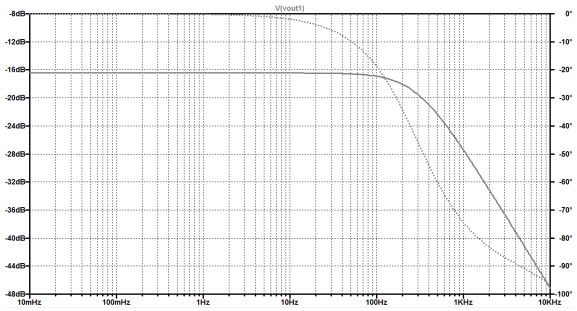

Figura 15: Diagrama de bode de la respuesta frecuencial del adaptador de señales de entrada.

3.3 V para evitar tensiones mayores a ese valor en el pin de entrada del conversor A/D del uC y así evitar dañar la placa en el caso de circunstancias operativas imprevistas.

La relación entre la tensión de salida del circuito y la tensión de entrada es

$$V_{\text{out}} = \frac{R_2}{R_1(1+RCs)(1+22\cdot 100e-9\cdot s)}V_{\text{in}} + \frac{1}{1+22\cdot 100e-9\cdot s}V_{\text{ref}} \quad (3)$$

Las Figs. 14 y 15 muestran, respectivamente y para el circuito de la Fig. 13, las respuestas temporales en la entrada del conversor A/D (señal verde) para señales sinusoidales en la salida de la planta (para salida 5 V, señal azul, y salida 10 V, señal roja), y la respuesta frecuencial en el diagrama de Bode.

4.3. Generación de la referencia de tensión

La tensión de referencia que se emplea en las etapas de corrección del *offset* se obtiene, ver Fig. 13, a partir de la referencia de voltaje LM385Z y un seguidor de voltaje para minimizar los efectos de carga sobre el dispositivo de referencia.

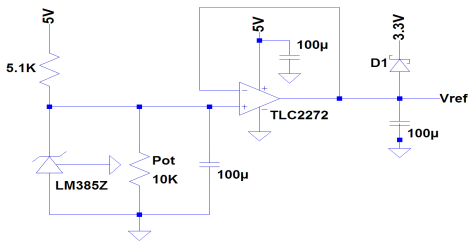

Figura 16: Circuito para obtener la referencia de tensión.

5. Descripción de las actividades prácticas

En este entorno descrito en la sección 2 se realizan 4 sesiones prácticas, en las 3 primeras se realizan diferentes experiencias mientras que en la cuarta se realiza una evaluación de los conocimientos adquiridos por el estudiante. Las experiencias realizadas en las 3 primeras prácticas son las siguientes:

- Modelado y Respuesta Temporal. Se analiza la respuesta temporal del sistema bajo estudio; este estudio se realiza en lazo abierto y lazo cerrado, para salida velocidad y posición.
- Respuesta frecuencial y estabilidad. Se construye un diagrama de Nyquist experimental comparándolo con la diagrama teórico.
- Diseño PID mediante asignación de polos. Se diseña la respuesta temporal deseada y posteriormente se diseña un controlador PID con el fin que el sistema de lazo cerrado se comporte acorde con dicha respuesta temporal.

Para guiar los pasos de los estudiantes se han desarrollado dos manuales, uno primero en el que se explica el entorno experimental y plantean los pasos a seguir[4] y uno segundo que corresponde a una introducción a MATLAB como herramienta de análisis de los sistemas dinámicos de tiempo discreto[3].

6. Resultados experimentales preliminares

Esta sección muestra los resultados preliminares obtenidos en la experimentación de lazo abierto de la planta y de lazo cerrado, con controlador PI, de control de velocidad.

6.1. Experimentación de lazo abierto

La Fig. 17 muestra el diagrama de bloques Simulink para aplicar a la planta una señal de control

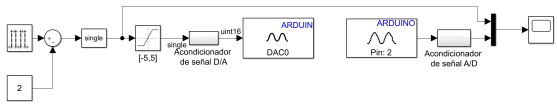

Figura 17: Diagrama de bloques Simulink para el experimento de lazo abierto.

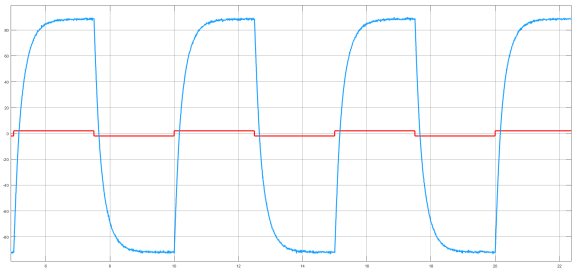

Figura 18: Respuesta temporal experimental en lazo abierto.

cuadrada (simétrica y de amplitud 2 V) y poder observar la respuesta en velocidad. Tal como se aprecia en la Fig. 18 la respuesta temporal obtenida corresponde, de una manera bastante fiel, a la típica respuesta de un sistema de primer orden y grado relativo uno. Efectuando mediciones sobre la gráfica se han obtenido los valores de constante de tiempo, $\tau = 0{,}265$ s, y ganancia estática, $k = 42{,}5$ rpm/V.

6.2. Experimentación con el control de velocidad

Una vez obtenido el modelo lineal aproximado de la planta con entrada tensión (V) y salida velocidad (rpm) se ha diseñado un controlador PI, véase Fig. 19, que da lugar a una respuesta a referencia escalón en lazo cerrado con error estacionario nulo y tiempo de establecimiento (banda de $\pm 5\,\%$) de 1 s. El controlador diseñado tiene como parámetros $k_p = 0{,}0184$ y $k_i = 3{,}5031e-4$.

Los resultados experimentales se muestran en la Fig. 20. En la gráfica superior, aparecen la referencia de velocidad y la velocidad real de la planta y, en la gráfica inferior, la correspondiente señal de control. Las pruebas recogidas son: primero, un

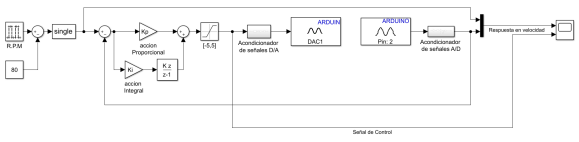

Figura 19: Diagrama de bloques Simulink para el lazo de control PI de velocidad.

cambio en la referencia del lazo cerrado en forma de escalón de -80 rmp a +80 rpm y viceversa y, segundo, la aplicación (en la velocidad de -80 rpm) del freno magnético y viceversa. De esta forma, se evalúa el comportamiento del sistema frente a cambios de referencia y a perturbación. Cabe destacar que, dadas las especificaciones del sistema en lazo cerrado, la señal de control se mantiene en régimen lineal no viéndose afectada por la saturación de las señal analógica de control. Asimismo, dada la presencia de la parte integral en el controlador, tanto el cambio de referencia escalón como la introducción del freno magnético (también una perturbación escalón) dan lugar después del correspondiente transitorio a un error estacionario nulo.

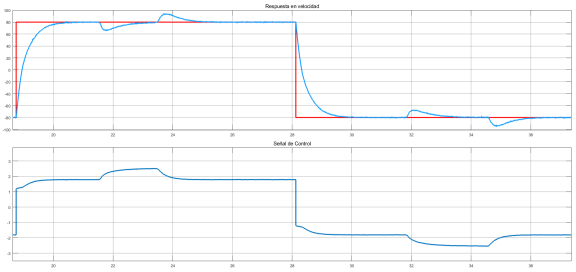

Figura 20: Respuesta temporal experimental en lazo cerrado con controlador PI.

7. Conclusiones

El trabajo ha presentado una plataforma de bajo coste, compuesta por un Arduino y MATLAB/Simulink para la realización de prácticas de control digital en el ámbito de la ingeniería industrial. Se han descrito los elementos de adaptación necesarios, las prestaciones de tiempo real ofrecidas y se han mostrado unos resultados preliminares.

Los resultados obtenidos son plenamente satisfactorios y todo apunto que será una excelente plataforma para la realización de prácticas de control automático en el ámbito de la ingeniería industrial.

Agradecimientos

This work has been partially funded by the Spanish national projects MICAPEM (ref. DPI2015-69286-C3-2-R, MINECO/FEDER), DPI2017-85404-P and the donation of Mathworks UPC-I-01523 .

English summary

CONTROLLER IMPLEMENTATION IN ARDUINO USING SIMULINK

Abstract

In this work, a MATLAB and Arduino based platform designed to develop digital control hands on laboratories is introduced. The paper describes the hardware components, MATLAB/Simulink elements and shows some of the preliminary results that are being obtained.

Keywords: Arduino, Simulink, Matlab, signal conditioning, DC motor control.

Referencias

[1] R. Barber, M. Horra, and J. Crespo. Practices using simulink with arduino as low cost hardware. *IFAC Proceedings Volumes*, 46(17):250 – 255, 2013.

[2] F.A. Candelas, G.J. García, S. Puente, J. Pomares, C.A. Jara, J. Pérez, D. Mira, and F. Torres. Experiences on using arduino for laboratory experiments of automatic control and robotics. *IFAC-PapersOnLine*, 48(29):105 – 110, 2015.

[3] Oriol Causí Casamor, Miquel Angel Mañanas Villanueva, Ramon Costa Castelló, and Luis Basañez Villaluenga. *Control amb Computador. Simulación en entorno MATLAB.* CPDA, Barcelona, Octubre 1999. ISBN 84-95355-04-3.

[4] Profesores de la asignatura. Control por computador: Pràcticas. Technical report, Escuela Técnica Superior de Ingeniería Industrial de Barcelona (ESTEIB), Barcelona, 1998.

[5] Masato ISHIKAWA and Ichiro MARUTA. Rapid prototyping for control education using arduino and open-source technologies. *IFAC Proceedings Volumes*, 42(24):317 – 321, 2010.

[6] P. Reguera, D. García, M. Domínguez, M.A. Prada, and S. Alonso. A low-cost open source hardware in control education. case study: Arduino-feedback ms-150. *IFAC-PapersOnLine*, 48(29):117 – 122, 2015.

[7] Jaroslav Sobota, Roman PiSl, Pavel Balda, and MiloS Schlegel. Raspberry pi and arduino boards in control education. *IFAC Proceedings Volumes*, 46(17):7 – 12, 2013.