

Prácticas experimentales de control digital con elementos de bajo coste

Adolf Izquierdo Borràs

Escola Politècnica Superior(EPS), Universitat de Lleida (UdL)

adolf.izquierdo@udl.cat

Ramon Costa-Castelló

Universitat Politècnica de Catalunya (UPC).

ramon.costa@upc.edu

Resumen

En este trabajo se presenta un entorno de bajo coste, compuesto por una placa Arduino y un circuito analógico, para su uso en prácticas de control digital. Se describe el entorno completo (tanto software como hardware), unas prácticas propuestos y se muestran algunos resultados experimentales.

Palabras clave: Prácticas de laboratorio, Control Digital, Arduino

1. Introducción

La realización de prácticas experimentales en estudios de ingeniería es una actividad de gran relevancia pedagógica, por ello es importante incorporar este tipo de actividades en la docencia. Desafortunadamente, el alto coste de la mayoría de plataformas experimentales hace que no siempre sea viable la realización de este tipo de actividades.

La aparición de plataformas hardware de bajo coste tipo Arduino y Raspberry 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 [1, 2, 3, 4, 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[3, 1]. 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.

El otro elemento que suele incrementar el coste de las prácticas es el dispositivo a controlar, este suele ser en la mayoría de casos el elemento más costoso. En este contexto el uso de circuitos electrónicos analógicos con plantas a controlar está tomando especial relevancia [6, 7] por su facilidad de montaje, bajo coste y interés pedagógico.

En este trabajo se presenta un entorno de prácticas para asignaturas de control digital [8, 9, 10], el entorno incorpora un circuito electrónico como planta a controlar y una placa Arduino MEGA, conectada al puerto USB de un computador personal

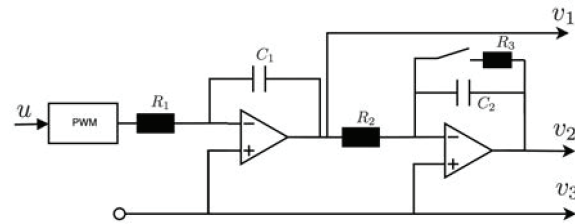


Figura 1: Esquema del circuito electrónico a controlar.

como elemento de adquisición de datos, procesado y actuación. El entorno permite prácticas introductorias de modelado de sistemas lineales, diseño e implementación de controladores digitales.

El trabajo está estructurado de la siguiente manera, la sección 2 describe el circuito analógico utilizado como planta a controlar y su montaje práctico; la sección 3 muestra los modelos utilizados para describir el comportamiento del sistema y las implementaciones realizadas en MATLAB/Simulink; la sección 4 describe un conjunto de experiencias prácticas propuestas sobre el sistema desarrollado y muestra el resultado de algunas de ellas, la sección 5 describe el entorno utilizado para implementar y validar los controladores; finalmente la sección 6 presenta algunas conclusiones y trabajos futuros.

2. Descripción del circuito analógico

Circuito	Arduino
u	Pin 5
v_1	Pin 4
v_2	Pin 7
v_3	Pin 5

Cuadro 1: Enlace entre el circuito y el Arduino.

En esta sección se describe el circuito que se utiliza como dispositivo a controlar. El esquema del circuito electrónico se muestra en la figura 1. Con la finalidad de simplificar al máximo el disposi-

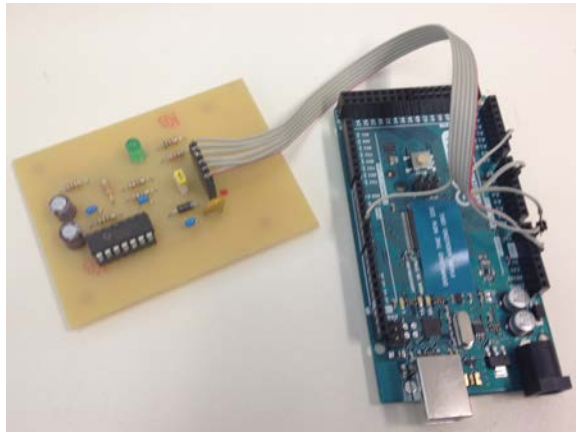


Figura 2: Fotografía de la implementación del circuito, la placa Arduino y su interconexión.

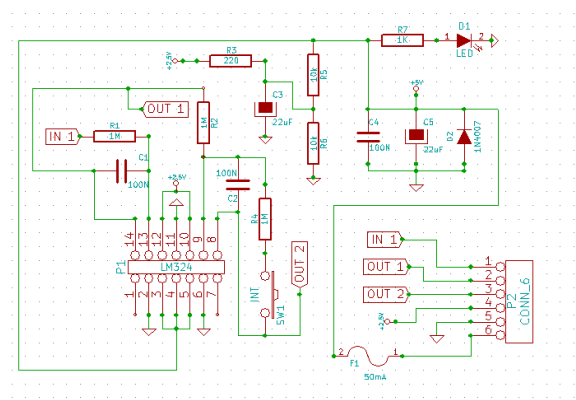


Figura 3: Esquemático de la placa.

tivo y su implementación, dado que el consumo del circuito es muy pequeño y que se ha pensado para que sea utilizado con una placa Arduino la alimentación se tomará directamente del puerto USB. Por tanto los amplificadores operaciones se alimentaran de forma asimétrica, es decir en el rango $[0, V_{cc}]$, donde V_{cc} corresponderá a la tensión que genere el puerto USB, unos 5 V nominalmente.

Dado que el Arduino Mega no dispone de conversores DA, se utilizará una salida digital con modulación en anchura de pulso (PWM) como acción de control. Así, el sistema presenta una entrada (u), correspondiente al *duty cycle* del modelador PWM (los niveles son entre 0 V y V_{cc}).

A efectos del diseño del sistema de control y con el fin de poder realizar diferentes tipos de experiencias se han introducido tres puntos de medida (v_1, v_2 y v_3) correspondientes a los niveles de tensión del tres puntos del circuito (todos ellos en el rango $[0, v_{cc}]$). Aunque v_1 y v_2 son salidas propiamente dichas, v_3 juega el papel de tensión de refe-

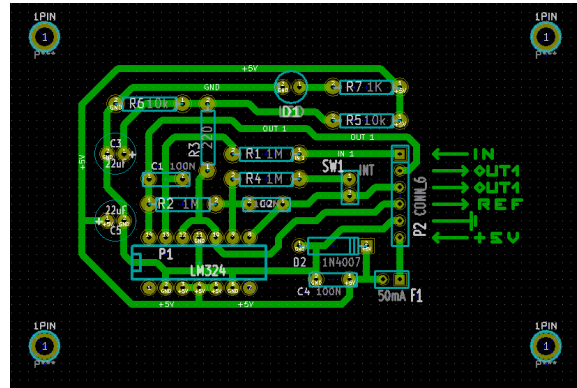


Figura 4: PCB de la placa.

rencia (masa), el valor de v_3 es aproximadamente el valor medio del intervalo $[0, v_{cc}]$ y se genera en el propio circuito mediante un divisor de tensión.

Aunque se puede modificar la conexión entre el Arduino Mega y el circuito se ha realizado de la forma indicada en la tabla 1. La figura 2 muestra una fotografía del circuito, la placa Arduino y el interconexión entre ambos. La figura 3 muestra el esquemático completo de la placa desarrollada. Como se puede comprobar se han incorporado algunos elementos de seguridad, estabilización de la alimentación y de extracción del valor de referencia v_3 . Finalmente la figura 4 muestra el PCB utilizado.

3. Modelado y Simulación del sistema

El modelo del circuito puede obtenerse aplicando conocimiento básico de electrónica analógica y amplificadores operacionales [11]. Así, asumiendo que a la salida del bloque de PWM hay un valor igual $u \cdot V_{cc}$ ¹ la transformada de Laplace de la tensión v_1 puede obtenerse como :

$$V_1(s) = V_3(s) - \frac{1}{R_1 \cdot C_1 \cdot s} (U(s)V_{cc} - V_3(s)). \tag{1}$$

De forma similar es posible obtener la transformada de Laplace de v_2 :

$$\begin{aligned} V_2(s) &= V_3(s) - \frac{1}{R_2 \cdot C_2 \cdot s} (V_1(s) - V_3(s)) \\ &= V_3(s) + \frac{1}{R_2 \cdot C_2 \cdot s} \frac{1}{R_1 \cdot C_1 \cdot s} (U(s)V_{cc} - V_3(s)). \end{aligned}$$

¹A la salida del bloque de PWM hay una señal de frecuencia 490 Hz que toma únicamente los valores 0 y V_{cc} pero que toma valor medio $u \cdot V_{cc}$. Dado que los circuitos son pasabajos pueden menospreciarse el efecto de las componentes armónicas y por tanto únicamente se tiene en cuenta la componente continua.

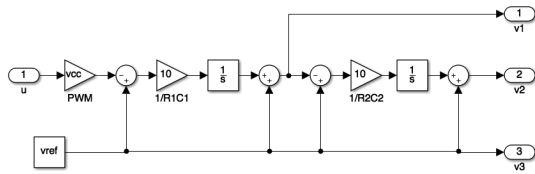


Figura 5: Modelo Simulink del sistema sin restricciones físicas.

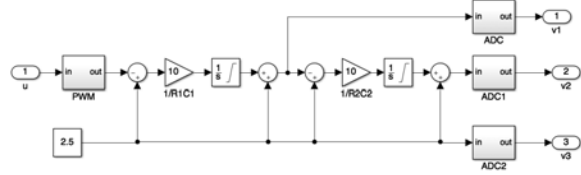


Figura 6: Modelo Simulink del sistema con restricciones físicas.

Para simular el comportamiento del sistema se ha desarrollado un modelo computacional utilizando Simulink (Figura 5), este modelo es una idealización del sistema real y será utilizado inicialmente para el diseño de los controladores. El sistema real está sometido a diferentes restricciones no recogidas en los modelos obtenidos anteriormente, las principales son :

- $u \in [0, 1]$. u corresponde al *duty cycle* del modular PWM, por lo tanto en la realidad no puede tomar valores por debajo del 0% ni por encima del 100%. En el modelo se ha incorporado una saturación de la acción de control para recoger esta restricción.
- $v_1, v_2, v_3 \in [0, v_{cc}]$ corresponden a las tensiones de salida de los amplificadores operacionales, por ello su valor no puede estar por encima de la tensión de alimentación (v_{cc}) ni por debajo de la mínima (0 V). Por ello los integradores se han limitado al rango $[-v_3, v_{cc} - v_3]$
- El modulador PWM presenta una cuantificación de 256 niveles (8 bits). Se ha incorporado esta cuantificación en el bloque del PWM.
- Los conversores AD presentan una cuantificación de 1024 niveles (10 bits). Se ha incorporado esta cuantificación en el bloque del ADXC

Estas limitaciones se han recogido en un modelo Simulink más complejo (Figura 6) que presenta un comportamiento prácticamente igual al del dispositivo real. Este modelo está pensado para poder

validar el comportamiento de los controladores antes de llevarlos al sistema real.

Todos los modelos desarrollados están a disposición bajo demanda a los autores.

4. Prácticas propuestas

4.1. Introducción

Esta sección presenta un conjunto de ejercicios propuestos a modo de prácticas; primero se propone realizarlos con los modelos de simulación y posteriormente con el sistema real. Claramente se puede prescindir de los ejercicios con los modelos pues se dispone de la prototipo real, pero bajo nuestro modo de ver el uso combinado de experimentación real y modelos define un marco de trabajo similar al que normalmente se usa al trabajar con procesos complejos.

Los ejercicios propuestos presentan complejidad incremental y realizan un recorrido por los principales conceptos vistos en una asignatura de control digital[8, 9, 10].

4.2. Modelado

El primer ejercicio planteado es obtener un modelo del sistema propuesto. A partir del análisis del circuito (Figura 1) se debe obtener un modelo de comportamiento del sistema, posteriormente el valor de los parámetros se deben obtener mediante ensayos sobre el modelo. Se propone obtener el modelo en dos pasos :

- Obtener un modelo que relacione v_1 , u y v_3 .
 - A partir del circuito obtener las ecuaciones que relacionan las diferentes variables.
 - Definir un experimento para obtener los parámetros del modelo obtenido.
 - Aplicar el experimento sobre el sistema sin restricciones (Figura 5) y calibrar el modelo.
 - Aplicar el experimento sobre el sistema con restricciones (Figura 6) y analizar el efecto de las mismas.
 - Aplicar el experimento sobre el sistema real (Figura 2) y calibrar el modelo.
 - Comparar los diferentes modelos
- Obtener un modelo que relacione v_2 , v_1 , u y v_3 .
 - A partir del circuito obtener las ecuaciones que relacionan las diferentes variables.
 - Definir un experimento para obtener los parámetros del modelo obtenido.

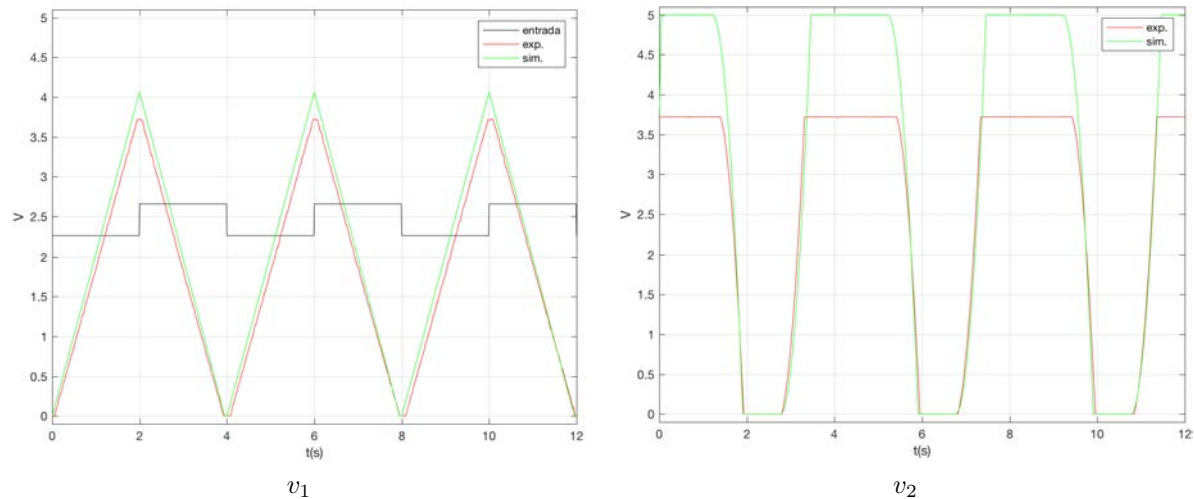


Figura 7: Ensayos para caracterizar el dispositivo: respuestas de las tensiones v_1 y v_2 , del modelo con restricciones y el sistema real, cuando la entrada es un tren de pulsos de amplitud 0,2V y frecuencia 0,25Hz.

- Aplicar el experimento sobre el sistema con restricciones (Figura 5) y analizar el efecto de las mismas.
- Aplicar el experimento sobre el sistema sin restricciones (Figura 6) y calibrar el modelo.
- Aplicar el experimento sobre el sistema real (Figura 2) y calibrar el modelo.
- Comparar los diferentes modelos
- Proponer un período de muestro T_s (por las limitaciones del sistema de control no debe ser inferior a 0,001s).
- Diseñar un sistema de control que cumpla la especificaciones. Para el diseño utilizar los modelos obtenidos en el apartado anterior.
- Validar el experimento sobre el modelo linealizado (Figura 5).
- Validar el experimento sobre el modelo con restricciones (Figura 6). Rediseñar el controlador en caso de ser necesario.
- Aplicar el experimento sobre el sistema real (Figura 2). Rediseñar el controlador en caso de ser necesario.

La figura 7 muestra el resultado de introducir en el sistema un voltaje de valor medio v_3 y amplitud 0,2V en el sistema, se muestra la respuesta de modelo completo (Figura 6) y el sistema real (Figura 2). Como puede observarse la respuesta de ambos elementos es prácticamente la misma, por lo que podemos concluir que el modelo describe perfectamente el comportamiento del sistema. Existen discrepancias en la parte superior de las gráficas pero ello es debido a que el modelo no contempla las caídas de tensión propias de los amplificadores operacionales y asume que la salida de los amplificadores puede tomar valores hasta v_{cc} .

4.3. Control de v_1

Se desea que v_1 siga un tren de escalones de amplitud 0,5V y período 2s (y valor medio v_3). Para ello se pide diseñar un controlador con error estacionario nulo en el seguimiento de escalones, con una respuesta transitoria con un sobrepico del 16% y un tiempo de estabilización de 0,32s. El controlador puede medir v_1 y v_3 .

Para alcanzar el objetivo se propone seguir los siguientes pasos

La figura 8 muestra el comportamiento de lazo cerrado de un controlador que cumple la especificaciones indicadas y mantiene la acción de control dentro del rango $[0, v_{cc}]$. Ello se puede conseguir con un controlador de primer orden de grado relativo 1.

4.4. Control de v_2 (lazo jerárquico)

Assumiendo que se dispone del controlador diseñado en el punto anterior, y utilizando un esquema de control jerárquico (Figura 9), diseñar un controlador que sea capaz de regular v_2 con un tiempo de establecimiento inferior a los 0,5s. Para validar el comportamiento utilizar una señal de referencia en forma tren de escalones de amplitud 0,5V y período 2s (y valor medio v_3). El sistema de control puede medir v_1, v_2 y v_3 .

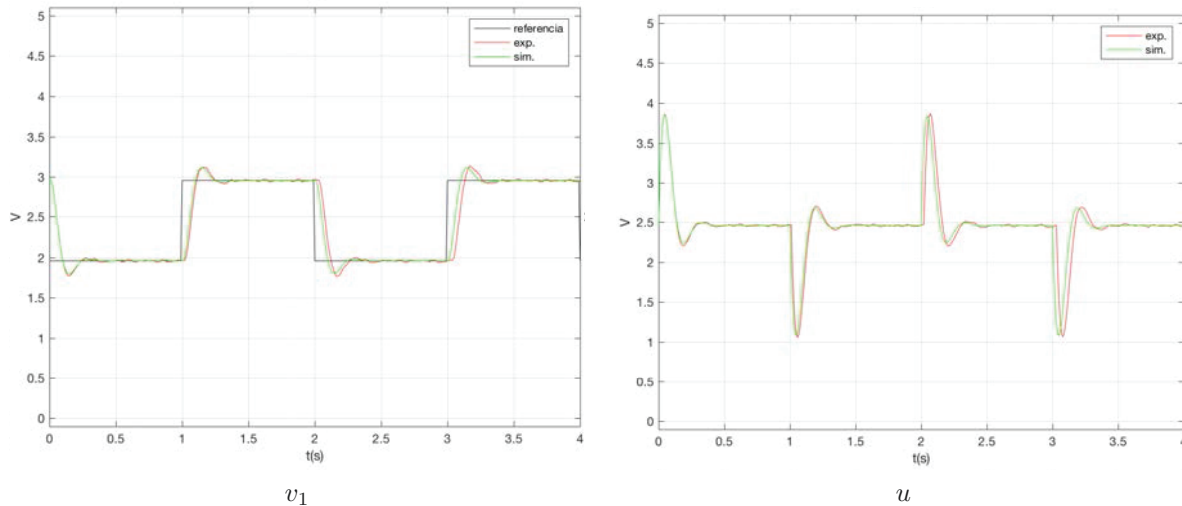


Figura 8: Control de v_1 : comportamiento de lazo cerrado. Evolución de v_1 y u en el modelo completo y el dispositivo real.

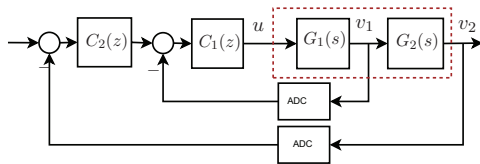


Figura 9: Esquema jerárquico de control.

Posteriormente se propone seguir los pasos siguientes:

- Diseñar un control que cumpla la especificaciones. Para el diseño utilizar los modelos obtenidos en el apartado anterior y el controlador diseñado en el apartado anteriores.
- Validar el experimento sobre el sistema sin restricciones (Figura 5).
- Validar el experimento el sistema con restricciones (Figura 6). Rediseñar el controlador en caso de ser necesario.
- Aplicar el experimento sobre el sistema real (Figura 2). Rediseñar el controlador en caso de ser necesario.

La figura 10 muestra la respuesta de v_2 utilizando un controlador jerárquico basado en el controlador de v_1 diseñado en el apartado anterior, $C_1(z)$, y un controlador proporcional, $C_2(z)$. Como se puede observar el sistema cumple las especificaciones indicadas.

4.5. Control de v_2 (lazo en cascada)

Utilizando un esquema basado en control en cascada (Figura 11), diseñar un controlador que sea

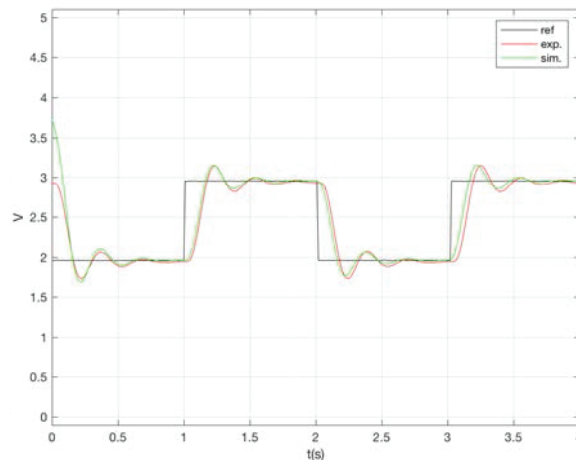


Figura 10: Control de v_2 con lazo jerárquico. Evolución del controlador diseñado sobre el modelo con restricciones y sobre el sistema real.

capaz de regular v_2 con un tiempo de establecimiento inferior a los 2,63s con sobrepico inferior al 63%. Para validar el comportamiento utilizar una señal de referencia en forma tren de escalones de amplitud 0,5V y período 6s (y valor medio v_3). El controlador puede medir únicamente v_2 y v_3 .

Se propone seguir los pasos siguientes:

- A partir del modelo que relaciona v_2 y u , diseñar un control que cumpla la especificaciones.
- Validar el experimento sobre el sistema sin restricciones (Figura 5).
- Validar el experimento sobre el sistema con restricciones (Figura 6). Rediseñar el controlador en caso de ser necesario.

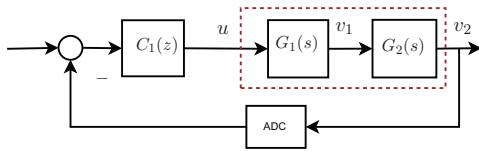


Figura 11: Esquema de control en cascada.

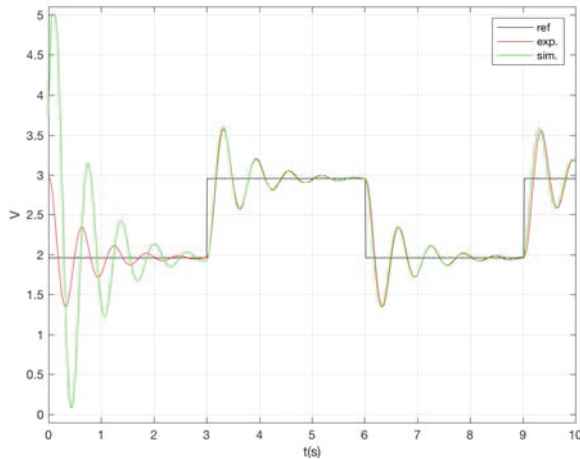


Figura 12: Control de v_2 con enfoque directo. Evolución del controlador diseñado sobre el modelo con restricciones y sobre el sistema real.

- Aplicar el experimento sobre el sistema real (Figura 2). Rediseñar el controlador en caso de ser necesario.

La figura 12 muestra la respuesta de v_2 utilizando un controlador de control directo (Figura 11) que cumple las especificaciones realizadas.

Dado que el sistema real (Figura 2) incorpora un conmutador, que conecta una resistencia en paralelo al segundo integrador, es posible activarlo en tiempo de ejecución, ello permite observar el cambio en el comportamiento del sistema de lazo cerrado. En este esquema de control, el cambio, no implica la pérdida de la capacidad de seguimiento de escalones pues conserva un integrador (se pasa de tipo 2 a tipo 1), ver Figura 13. Esto sería diferente con el esquema de control jerárquico, ver Figura 14, pues en este caso el conmutador pasa el sistema de lazo cerrado de tipo 1 a tipo 0 cosa que comporta la pérdida de la capacidad de seguimiento de escalones.

5. Arduino y entorno software

Para implementar los controladores y analizar los resultados obtenidos se utiliza únicamente el entorno de desarrollo que proporciona Arduino. Este permite desarrollar controladores en un lenguaje muy similar a C, incorporando además funciones de alto nivel para gestionar las entradas y salidas. El propio entorno provee un editor/compilador.

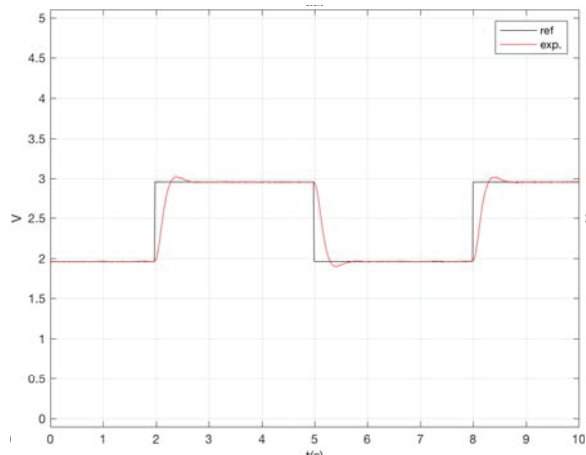


Figura 13: Control de v_2 con enfoque directo. Evolución del controlador diseñado sobre el sistema real cuando se activa el conmutador.

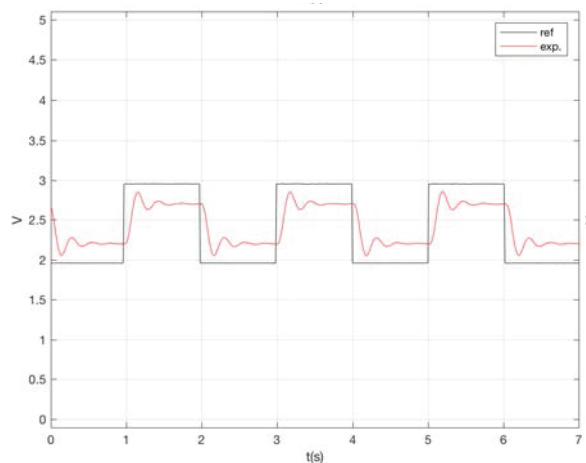


Figura 14: Control de v_2 con enfoque jerárquico. Evolución del controlador diseñado sobre el sistema real cuando se activa el conmutador.

Con el objetivo de comunicar la placa Arduino con el PC de desarrollo se ha utilizado el puerto serie del Arduino (integrado en la conexión USB). El propio entorno de desarrollo ofrece un monitor del puerto serie en modo texto o en modo gráfico (Figura 15). Ello hace que se pueda validar y visualizar fácilmente el comportamiento del sistema sin necesidad de programas adicionales ni instrumentación.

Con el fin de facilitar la implementación de los controladores se entrega a los estudiantes un código base (*template*) que implementa una tarea periódica (Figura 16). Para implementar la tarea periódica se utiliza la librería `arduino-timer-library`. A efectos de la implementación los estudiantes deben codificar la ecuaciones en diferencias del controlador en la zona indicada. El código genera además un señal de referencia para el sistema de control y envía al puerto serie los valores de la principales señales utilizadas.

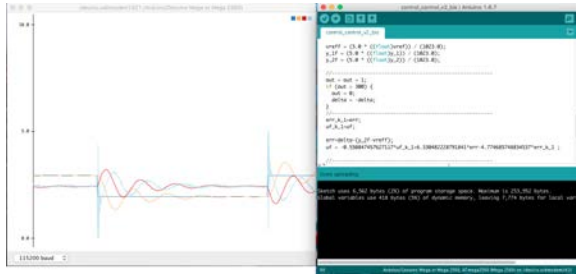


Figura 15: Vista del entorno de desarrollo y visualización utilizado.

Los estudiantes más avanzados pueden fácilmente adaptar el código a sus necesidades.

6. Conclusiones

En este trabajo se han propuesto unas prácticas experimentales de bajo coste para asignaturas de control de tiempo discreto. El entorno de prácticas está compuesto por una placa Arduino y un pequeño circuito analógico formado por dos amplificadores operacionales y diferentes componentes pasivos. Pese a la simplicidad de los elementos el entorno propuesto permite realizar experiencias prácticas de una cierta complejidad y con restricciones propias de los sistemas reales. La gestión de estas restricciones abre un nuevo campo de trabajo a la mayoría de estudiantes y permite introducir reflexiones sobre la aplicabilidad de la teoría. Además del sistema real se han desarrollado dos modelos Simulink de diferente complejidad que pueden ser usados de forma previa a la implementación del controlador en el sistema real.

El bajo coste y la simplicidad del equipo, los estudiantes sólo necesitan un PC y el equipo proporcionado. Todo ello hace que los estudiantes se lo puedan llevar a casa y trabajar con él sin restricciones en espacio y tiempo.

En este trabajo se han mostrado unas prácticas basadas en formulación entrada salida y con problemas de diferente complejidad. Se pueden plantear también experiencias en el espacio de estado, ya sea mediante realimentación de estado o mediante el uso de observadores.

En el futuro se incorporaran circuitos de mayor orden y se diseñará un mecanismo de integración más robusto entre la placa Arduino y el circuito diseñado.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el proyecto DPI2015-69286-C3-2-R (MINECO/FEDER) y el proyecto 2014 SGR 267 de la AGAUR (agencia de la Generalitat de Catalunya).

Referencias

- [1] 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.
- [2] F.A. Candelas, G.J. García, S. Puente, J. Pomaes, 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] R. Barber, M. Horra, and J. Crespo. Practices using simulink with arduino as low cost hardware. *IFAC Proceedings Volumes*, 46(17):250 – 255, 2013.
- [4] 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.
- [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] J. M. Fuertes Armengol, R. Villa Millaruelo, J. Ayza Graells, P. Marès Martí, P. Martí Colom, M. Velasco García, J. Yépez Castillo, and G. Torres Guerrero. Design of a hands-on course in networked control systems. <http://hdl.handle.net/2117/18144>.
- [7] P. Marti, M. Velasco, J. M. Fuertes, A. Camacho, and G. Buttazzo. Design of an embedded control system laboratory experiment. *IEEE Transactions on Industrial Electronics*, 57(10):3297–3307, Oct 2010.
- [8] Ramon Costa Castelló and Enric Fossas. *Sistemas de Control en Temps Discret*. Edicions UPC, 2014. ISBN: 978-84-9880-492-8.
- [9] Roland Longchamp. *Comande Numériques de Systèmes Dynamiques. Cours d'Automatique*. Presses Polytechniques et Universitaires Romandes, Laussane, 2006.
- [10] Benjamin C. Kuo. *Digital Control Systems*. ISBN : 9780195120646.
- [11] John V. Wait, Lawrence P. Huelsman, and Granino A. Korn. *Introduction to operational amplifier theory and applications*. McGraw-Hill Inc. US, 1975. ISBN : 0070677654.

```

#include "Timer.h"

Timer t;
// Periodo de muestreo 10ms
const int Ts = 10;

// Datos canal 1 (v1)
const int Pin_y_1 = 4;
int y_1 = 0;
float y_1f = 0;

// Datos canal 2 (v2)
const int Pin_y_2 = 7;
int y_2 = 0;
float y_2f = 0;

// Datos canal 3 (v3)
const int Pin_vref = 9;
int vref = 0;
float vreff = 0;

// Datos accion de control (u)
const int Pin_u = 5;
int u = 0;
float uf = 0;
float uf_ = 0;

// Datos senal de ref
int out = 0;
float delta = 0.2;

// Configuracion del sistema
void setup() {
  // Velocidad canal serie
  Serial.begin(115200);

  // Puerto PWM
  pinMode(Pin_u, OUTPUT);

  // Periodo de la tarea periodica
  t.every(Ts, control);
}

//programa principal
void loop() {
  t.update();
}

void control()
{
  // Lectura de los datos
  y_1 = analogRead(Pin_y_1);
  y_2 = analogRead(Pin_y_2);
  vref = analogRead(Pin_vref);

  // Adaptacion de los datos
  vreff = (5.0 * ((float)vref)) / (1023.0);
  y_1f = (5.0 * ((float)y_1)) / (1023.0);
  y_2f = (5.0 * ((float)y_2)) / (1023.0);

  // Generador senal de referencia
  out = out + 1;
  if (out > 200) {
    out = 0;
    delta = -delta;
  }

  //-----
  //Codigo del Controlador

  uf = delta;

  //-----

  // adaptacion senal de control
  uf_ = vreff + uf;
  if (uf_ > 5.0) {
    uf_ = 5.0;
  }
  if (uf_ < 0.0) {
    uf_ = 0.0;
  }
  u = (int)((255.0 * uf_) / 5.0);
  analogWrite(Pin_u, u);

  // Datos al puerto serie
  Serial.print(uf_, 6);
  Serial.print("\t");
  Serial.print(y_1f, 6);
  Serial.print("\t");
  Serial.print(y_2f, 6);
  Serial.print("\t");
  Serial.print(vreff, 4);
  Serial.print("\t");
  Serial.println("");
}

```

Figura 16: Código base ofrecido a los estudiantes.