

Control de posición de un balancín mediante un microprocesador Arduino y Simulink.

Eduardo J. Moya, F. Javier García y A. Poncela, *Miembros de ITAP - Universidad de Valladolid, Paseo del Cauce 59, 47011 Valladolid, España (e-mail: {edumoy, javgar, poncela}@eis.uva.es).*

Abstract— En este trabajo se muestran detalles del equipo materializado por el Departamento de Ingeniería de Sistemas y Automática (ISA) de la Escuela de Ingenierías Industriales (EII) de la Universidad de Valladolid (UVa) para la realización de prácticas de laboratorio de la asignatura Fundamentos de Automática presente en los Grados de Ingeniería (conforme a las directrices Bolonia y adaptada al EEES), de las titulaciones impartidas en la EII de la UVA para cientos de alumnos.

Se presenta en este artículo el desarrollo del control de posición de una barra giratoria con un grado de libertad utilizando un motor con una hélice como fuerza impulsora solidario en un extremo de dicha barra. El objetivo principal es que sirva como equipo docente de prácticas a los alumnos.

El control se llevará a cabo utilizando Simulink y la tarjeta Arduino UNO, siendo el mismo un controlador PID al que se le puede añadir compensación de efecto Windup, con objeto de obtener un sistema rápido sin oscilaciones y error estacionario nulo. El alumno debe ser capaz de obtener el modelo del sistema, a partir de él diseñar los parámetros de controlador y realizar las pruebas oportunas con el equipo. Se cuenta con un total de 36 equipos para los alumnos y cada uno presenta un funcionamiento diferente debido básicamente a que tienen diferentes rozamientos.

Index Terms— Balancín, Simulink, Arduino, controlador PID, efecto Windup, oscilaciones, error estacionario.

I. INTRODUCCIÓN

El estudio la automatización y los sistemas de control es una de las ramas con más importancia en la ingeniería. El estudio de esta disciplina tiene una gran relevancia en el mundo de la docencia electrónica y automática.

Mediante el siguiente artículo, se pretende explicar la implementación de un control de posición en una barra que presenta un grado de libertad. La barra puede girar respecto a un eje, el movimiento de giro es provocado por una fuerza de empuje producida por una hélice y un motor de corriente directa, de manera que controlando la velocidad de giro del motor se podrá regular la fuerza de empuje que actúa sobre la barra y con ello la posición de la misma. La posición angular de la barra se determina mediante un potenciómetro solidario al eje de giro de la barra.

Para ello se ha utilizado la automatización de un sistema mecánico, en el cual se pretende controlar la posición mediante el uso de la herramienta Simulink de Matlab. El sistema consta de dos barras ancladas a modo de balancín, en

uno de los extremos de la barra se encuentra un motor con una hélice que será el encargado de producir la fuerza de empuje proporcional al giro de su hélice que será la responsable de los cambios de posición de nuestro sistema.

Para controlar el motor y visualizar un circuito utilizaremos un montaje que incluirá una tarjeta de Arduino Uno, aprovechando la moda que actualmente se está siguiendo por parte de nuestros estudiantes de programar y trabajar con este tipo de equipos.

Como objetivo principal planteado es que el alumno debe ser capaz de regular la posición angular de la barra, lograr que el sistema alcance la referencia con el menor sobrepico posible y con error estacionario nulo.

Además debe ofrecer una buena respuesta ante perturbaciones, el sistema deberá ser capaz de corregir su posición ante la acción de fuerzas externas que puedan provocar una desviación de la posición de la barra respecto a la referencia. Para conseguir estos objetivos el alumno deberá aplicar los métodos de identificación y las tablas de Ziegler-Nichols.

Por otra parte y como objetivos secundarios se pretende que el alumno alcance las siguientes competencias:

- Conocer el funcionamiento de las tarjetas Arduino y su entorno.
- Conocer las posibilidades de manejo de nuestro sistema mediante la herramienta Simulink.
- Conocer los pasos a seguir para la instalación del software necesario para el funcionamiento del sistema en cualquier equipo.
- Obtención del modelo matemático del sistema a través de las leyes físicas y con datos extraídos de ensayos.
- Elaboración de un sistema en lazo abierto que nos permita darle diferentes entradas al sistema.
- Elaboración de un sistema en lazo cerrado con un controlador PID, que controle de una forma correcta la posición de la barra ante posibles perturbaciones ajenas al sistema.

Tras esta breve introducción, el artículo se estructura de la siguiente manera: en la segunda sección se realiza una breve descripción de las características principales de la planta, es decir del sistema mecánico, mientras que en la sección cuarta se presentan las características fundamentales del hardware utilizado para el desarrollo de la aplicación. En la sección cinco se describe la obtención de los parámetros del controlador mediante los métodos de Ziegler-Nichols. Por

último en las secciones seis y siete se muestran las conclusiones obtenidas y las líneas futuras de trabajo. Finalizando con las referencias utilizadas.

II. CARACTERÍSTICAS DE LA PLANTA

La planta está formada por un soporte metálico de 34 cm y el balancín o barra móvil cuya longitud es de 50 cm. El balancín va unido al soporte por medio de dos cojinetes o cajas de bolas que permiten el movimiento de giro. En uno de los extremos del balancín hay un soporte donde va colocado el motor y la hélice, el motor es del fabricante Mabuchi con una potencia nominal de 1866 vatios y un eje de 2.5 mm de diámetro, la hélice tiene un diámetro de 13.5 cm. Solidario al eje de giro mediante unos pequeños tornillos está fijado el cursor de un potenciómetro que actúa como sensor de posición. El sistema también cuenta con dos toques que limitan el movimiento de giro del sistema entre 44° y 136° medidos a partir de la vertical. (Figura 1).



Fig. 1. Planta real a controlar

Dentro de una caja se encuentra el circuito electrónico formado por una tarjeta de Arduino UNO, un transistor NPN (BD 139) con un radiador solapado para que disipe el calor y un transformador de corriente que nos permite conectarlo a la red eléctrica y obtener 24 voltios de cc a través de sus terminales. Este circuito (Figura 2) es el encargado de transmitir las órdenes que enviamos a través de Simulink y de recibir la posición de la barra para poder visualizar su estado con el ordenador.

El funcionamiento básico de la planta consiste en que la barra móvil, en la que uno de sus extremos hemos colocado un motor que empezará a girar cuando por el circule intensidad, ejercerá una fuerza, que según sea mayor o menor que la

anterior podrá desplazar ésta en un sentido u otro. La intensidad que circula por el motor estará controlada por un circuito eléctrico en el que tendremos control directo desde nuestra CPU gracias a su conexión USB con la placa Arduino. Desde el ordenador podremos controlar la diferencia de potencial que se le aplica al motor, como es el caso del circuito en lazo abierto, o la posición que queremos que ocupe nuestra barra móvil cuando estemos trabajando en lazo cerrado.

La posición se conocerá gracias a un potenciómetro alimentado desde la placa Arduino y a los datos que entregue a una de las entradas analógicas. Interpretando estos datos correctamente podremos conocer la posición de la barra móvil a través la pantalla del ordenador.

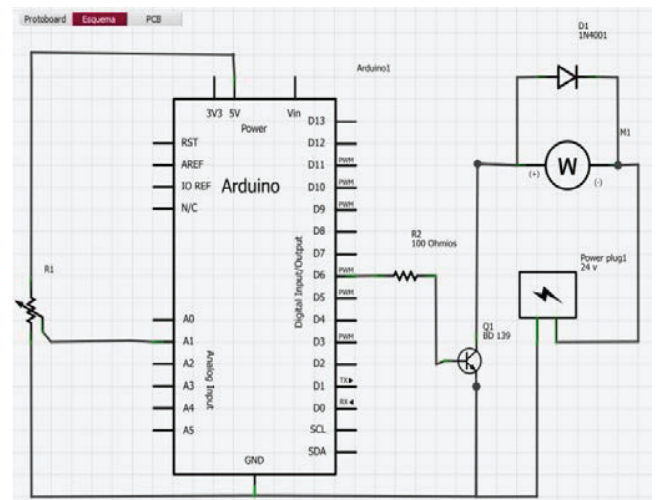


Fig. 2. Esquema eléctrico de conexión

III. HARDWARE UTILIZADO

A. Arduino UNO

Arduino [1] es una placa electrónica que contiene un microcontrolador y su objetivo se basa en pequeños proyectos, para aficionados y amantes de la electrónica en general. Su principal característica es la facilidad con la que se programa, a diferencia de las demás placas con microcontroladores del mercado que su programación es más laboriosa, además es una empresa basada en software y hardware libre con la ventaja que se puede utilizar en cualquier ambiente y proyecto. No hay que olvidar tampoco su otra característica que es la que la hace realmente competitiva es su relación precio-prestaciones.

Las placas de Arduino se pueden utilizar de diferentes maneras, ya sean alimentadas a través de USB por medio del ordenador o con una pequeña batería sin necesidad de conectarse con el ordenador. Se programa a través de un programa gratis que se descarga a través de la página web de Arduino, y a través de este se transfiere el programa que se escriba desde el ordenador a la placa, estos programas utilizan un lenguaje de programación propio de Arduino basado en Wiring. Pero también se puede utilizar con otros programas,

como por ejemplo, Simulink de Matlab (tiene librerías para utilizar Arduino), pero siempre cargando un programa a la placa que interacciona correctamente con Simulink, suelen ser programas básicos que vienen con la librería de Matlab, o con la librería del programa que quieres utilizar.

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede interaccionar con aquello que le rodea controlando luces, motores y otros actuadores. Dispone de diferentes placas dependiendo de la necesidad que tenga el proyecto, en concreto en este se ha utilizado Arduino Uno.

Arduino UNO es una placa que contiene 14 pines de entradas/salidas digitales (de las cuales 6 se pueden utilizar como salidas PWM), 6 pines de entradas analógicas, un botón de reset, una conexión para USB, una conexión para alimentarle de forma externa, todo ello basado en el microcontrolador ATmega328. Se puede alimentar de dos formas a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

La fuente externa (no USB) puede venir a través de un convertidor CA/CC o con una batería. El convertidor se conecta a través de la clavija Jack de alimentación de la placa. Si dispones de la batería esta se alimenta a través de los pines Gnd y Vin.

Los pines de alimentación son los siguientes:

- VIN: Es la tensión de entrada a la placa Arduino cuando se utiliza una fuente de alimentación externa.
- 5V: Este pin da una salida de 5V regulada por el regulador de la placa.
- 3.3V: Este pin da una salida de 3.3 voltios regulados por la placa, con una corriente máxima de 50 mA.
- GND: pin de tierra.

IOREF: Este pin de la placa Arduino proporciona la tensión de referencia con el que el microcontrolador opera. Un escudo correctamente configurado puede leer el voltaje del pin IOREF y seleccionar la fuente de alimentación adecuada.

Cada uno de los 14 pines digitales de la placa de Arduino uno se puede usar como una entrada o salida, utilizando las funciones `pinMode()`, `digitalWrite()`, y `digitalRead()`. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia interna de pull-up (desconectada por defecto) de 20-50 k. Además, algunos pines tienen funciones especializadas:

- Serie 0 (RX) y 1 (TX): Se utiliza para recibir (RX) y transmitir (TX) datos serie TTL. Estos pines están conectados a los pines correspondientes del ATmega8U2 USB-to-Serial TTL chips.
- Las interrupciones externas (2 y 3): Estos pines pueden ser configurados para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor.
- PWM(3, 5, 6, 9, 10, y 11): Proporcionar 8-bit de salida PWM con la función `analogWrite()`
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines son de apoyo para la comunicación SPI.

- TWI: A4 o A5 y SDA pin o pines SCL. Apoyo para la comunicación TWI.
- AREF. Tensión de referencia para las entradas analógicas. Se utiliza con la función `analogReference()`.
- Reset. Pon este pin a cero para reiniciar el microcontrolador. Normalmente se utiliza para agregar un botón de reinicio a los escudos que bloquean la placa.

El Arduino uno tiene 6 entradas analógicas, con la etiqueta A0 a A5, cada una de los cuales proporcionan 10 bits de resolución (es decir, 1024 valores diferentes). Por defecto se mide desde cero a 5 voltios.

B. PWM

Para mover el motor se utiliza una salida PWM [2] del controlador Arduino. La Modulación por Ancho de Pulso (PWM = Pulse Width Modulation) es una técnica para simular una salida analógica con una salida digital. El control digital se usa para crear una onda cuadrada, una señal que conmuta constantemente entre encendido y apagado. Este patrón de encendido-apagado puede simular voltajes entre 0 (siempre apagado) y 5 voltios (siempre encendido) simplemente variando la proporción de tiempo entre encendido y apagado.

A la duración del tiempo de encendido (ON) se le llama Ancho de Pulso (pulse Width). Para variar el valor analógico cambiamos, o modulamos, ese ancho de pulso. Si repetimos este patrón de encendido-apagado lo suficientemente rápido por ejemplo con un LED el resultado es como si la señal variara entre 0 y 5 voltios controlando el brillo del LED.

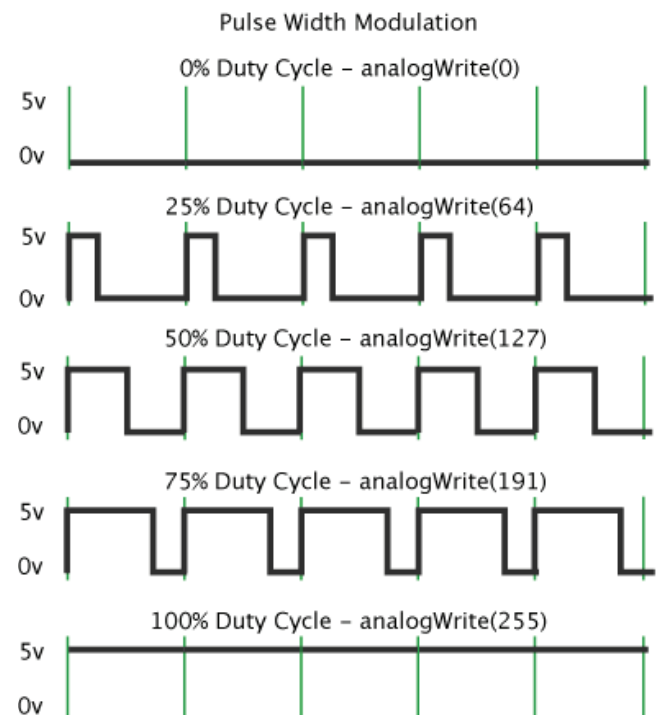


Fig. 3. Funcionamiento de la modulación PWM.

En la Figura 3 las líneas rectas representan un periodo regular. Esta duración o periodo es la inversa de la frecuencia del PWM. En otras palabras, con la Arduino la frecuencia

PWM es bastante próxima a 500Hz lo que equivale a periodos de 2 milisegundos cada uno. La llamada a la función `analogWrite()` debe ser en la escala desde 0 a 255, siendo 255 el 100% de ciclo (siempre encendido), el valor 127 será el 50% del ciclo (la mitad del tiempo encendido), etc.

C. Arduino-Simulink-Matlab

Como sabemos Simulink es una herramienta de Matlab que funciona mediante un entorno de programación visual, las funciones están representadas por bloques, lo que hace muy sencillo su utilización sin necesidad de emplear lenguajes complejos de programación.

Tiene una conexión directa con Matlab, pudiendo exportar los resultados obtenidos en Simulink para hacer análisis más exhaustivos y poder obtener nuevos resultados. También nos dan la posibilidad de incorporar algoritmos propios de Matlab.

Además ofrece la posibilidad de conectar el modelo con el hardware para comprobar en tiempo real y de una manera física el funcionamiento de este. Es uno de los diferentes tipos de software que puede establecer comunicación con Arduino, ya que tiene una biblioteca para trabajar (ArduinoIO). Para ello lo primero que habrá que hacer es cargar un programa predefinido que nos facilita Matlab, que configurara sus pines para que pueda establecer conexión con los diferentes bloques de la biblioteca ArduinoIO. La conexión de Arduino con el ordenador será USB.

IV. MODELO DEL SISTEMA.

El movimiento que realiza la barra móvil es el de rotación de un cuerpo rígido alrededor de un eje fijo. En estos movimientos lo que acelera al cuerpo angularmente es el torque y no solo la fuerza que se le aplica.

Para obtener el modelo del sistema se calcula la suma de momentos aplicados sobre el sistema respecto a su eje de giro. Se tiene por un lado el peso del conjunto motor-hélice, por otro lado la fuerza de empuje de la hélice al girar y la fuerza de rozamiento en el eje de giro, debida fundamentalmente al potenciómetro.

La figura 4 representa las fuerzas que actúan sobre el mecanismo. F_e es la fuerza que ejerce la hélice al girar, Mg es el peso del motor con el soporte y la hélice y por último M_r es el momento debido al rozamiento en el eje de giro. Se considera que el eje de giro está en el centro de gravedad de la barra. La segunda ley de Newton aplicada al giro da como resultado:

$$\sum M = I\ddot{\theta} \quad (1)$$

$$F_e \frac{L}{2} - Mg \sin(\theta) \frac{L}{2} - M_r = I\ddot{\theta} \quad (2)$$

Donde L es la longitud de la barra. El rozamiento M_r es proporcional a la velocidad de giro $B\dot{\theta}$. Todos los parámetros son fácilmente calculables excepto B , que además varía de un equipo a otro por las características de construcción. La ecuación resultante es:

$$0.245F_e = 0.049\sin\theta + B\dot{\theta} + 0.01443\ddot{\theta} \quad (3)$$

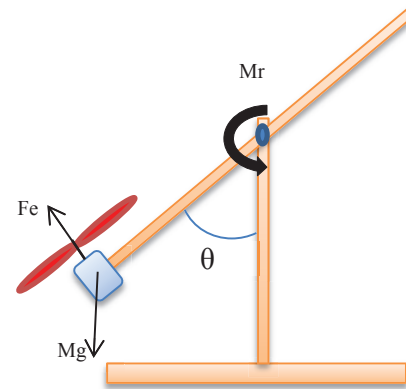


Fig. 4. Fuerzas que actúan sobre el sistema.

Si se linealiza dicha ecuación en torno a un punto de equilibrio $\theta_0 = 45^\circ$ se obtiene:

$$0.245\Delta F_e = 0.0346\Delta\theta + B\dot{\theta} + 0.01443\ddot{\theta} \quad (4)$$

Lo que permitiría obtener la función de transferencia [3] en caso de conocerse B :

$$\frac{\Delta\theta}{\Delta F_e}(s) = \frac{0.245}{0.01443s^2 + Bs + 0.0346} \quad (4)$$

V. OBTENCIÓN DE LOS PARÁMETROS DEL CONTROLADOR MEDIANTE LOS MÉTODOS DE ZIEGLER-NICHOLS.

En esta sección se utilizará el primer método de Ziegler-Nichols [3], [4] para obtener un modelo del sistema y a partir de este modelo definir un controlador PI.

En este primer método se trabaja con el sistema en lazo abierto utilizando Simulink. El esquema de funcionamiento es el que se muestra en la Figura 5. El alumno debe previamente obtener la relación entre la señal aplicada a la entrada (que es un valor entre 0 y 255) y el ángulo de la barra giratoria.

El funcionamiento en lazo abierto será muy básico. Mediante nuestro ordenador seleccionaremos la diferencia de potencial que queremos tener entre los bornes de nuestro motor y conoceremos la posición de la barra mediante el potenciómetro como hemos explicado. El circuito en Simulink quedará de la siguiente forma:

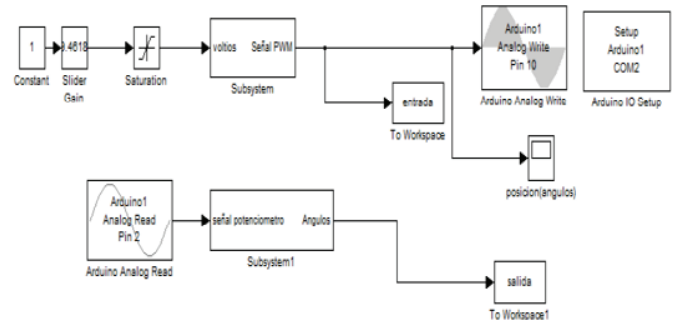


Fig. 5. Diagrama de bloques en Simulink para trabajar en lazo abierto.

Se utilizan unos bloques especiales para comunicar Simulink con Arduino [5] que son de libre distribución y pueden descargarse directamente de la página web de Mathworks.

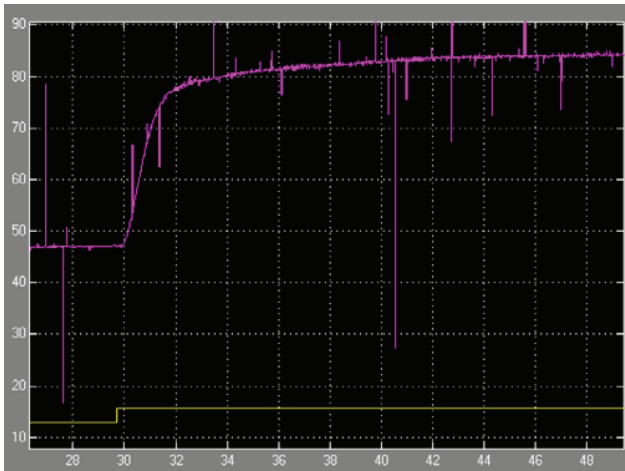


Fig. 6. Respuesta del Sistema en lazo abierto ante una entrada escalón.

Utilizando este diagrama en Simulink se obtienen las gráficas de la figura 6. Trabajando sobre estas gráficas de forma manual se puede llegar a un modelo de 1er orden con un retardo, como puede apreciarse en la figura 7, si bien como se ha visto en el modelado del sistema, este sistema es un sistema de 2º orden, sobreamortiguado como se aprecia en la forma de su respuesta.

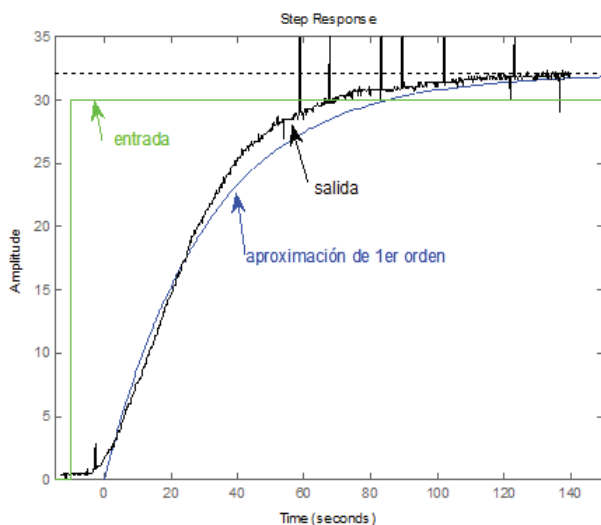


Fig. 7. Aproximación de la respuesta a la de un sistema de primer orden con retardo

A partir de este modelo el alumno usando la tabla I, puede determinar un primer controlador PID para después ajustarlo por tanteo.

$$\frac{Y(s)}{U(s)} = \frac{K e^{-ds}}{Ts+1} \quad (5)$$

Los valores que se obtuvieron fueron $K=1.1$, $d=0.4$ sg y $T=1.5$ sg, y de acuerdo con estos valores los valores de K_p y T_i en el controlador PI serían: $K_p=3.18$ y $T_i=1.33$. Probamos estos valores en nuestro sistema partiendo de una referencia angular de 70° a otra de 85° .

TABLA I

Valores de sintonización de un PID por el método 1.

Tipo de controlador	K_p	T_i	T_d
P	$\frac{T}{dK}$		
PI	$\frac{0.9T}{dK}$	$3.33d$	
PID	$\frac{1.2T}{dK}$	$2d$	$\frac{d}{2}$

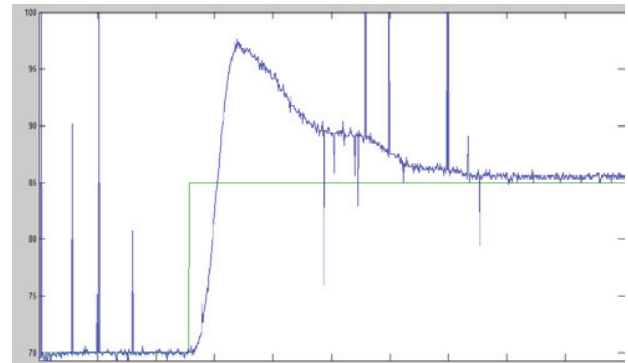


Fig. 8. Respuesta do el Sistema en lazo cerrado con el PI obtenido para un cambio en la referencia de 70° a 85° .

Como se observa en la figura 8 el sistema pasa de una situación estable de 70° y le damos un salto de 15° para llegar hasta los 85° , se aprecia un sobreimpulso que llega más allá de los 95° para después ir reduciendo hasta los 85° .

En la siguiente prueba se partió de un ángulo más elevado y se dio un escalón mayor de 80° a 110° .

En esta figura 9 pasamos de un situación estable de 80° a un nivel de lo llamado grandes de 110° , se observa que la gráfica crece de una forma rápida hasta que se choca con la limitación física (136°) la cual la hace rebotar hasta los 110° y efectuar una oscilación que hace que el sistema sea un poco más lento. Se aprecia que cuando se trabaja con valores angulares altos el sistema roza la inestabilidad, ya que se separa mucho del modelo lineal que se obtuvo en una posición angular entre 50° y 70° .

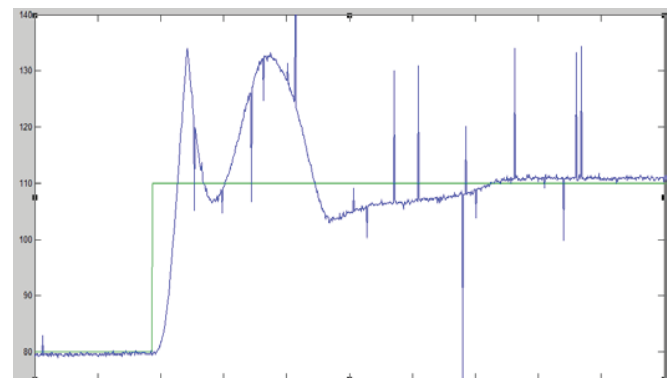


Fig. 9. Respuesta do el Sistema en lazo cerrado con el PI obtenido para un cambio en la referencia de 80° a 110° .

También se han hecho pruebas con saltos negativos como en la figura 10. Se aprecia que se produce un sobreimpulso negativo al igual que en los saltos positivos. También se apreció que debido al rozamiento el balancín presenta cierta inercia, y que a pesar de que la acción de control aumenta, en ocasiones la posición del balancín no cambia hasta vencer ese rozamiento.

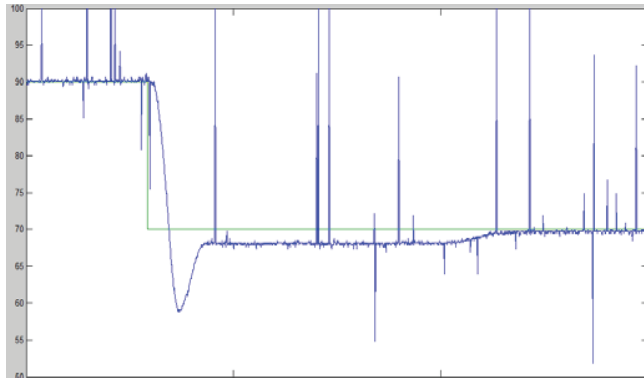


Fig. 10. Respuesta do el Sistema en lazo cerrado con el PI obtenido para un cambio en la referencia de 90° a 70°.

Tras varias pruebas de ensayo en las que se han ido modificando tanto el parámetro K_p como el parámetro T_i se ha llegado a la conclusión de que los valores más acertados para nuestro sistema son: $K_p=2,3$ y $T_i=0.91$.

VI. CONCLUSIONES

Los resultados conseguidos con este trabajo han sido satisfactorios. Se han cumplido los objetivos marcados al inicio de éste, obteniendo un correcto funcionamiento y control así como un completo conocimiento de todo el sistema, tanto de la parte física como de la programación.

Se ha podido comprobar la versatilidad de Arduino, aunque no se haya hecho un amplio uso de posibilidades nos ha permitido ser conscientes de todo su potencial a la hora de hacer cualquier tipo de proyecto relacionado con la electrónica. Una de sus grandes ventajas es su versatilidad ya que se puede trabajar con él desde diferentes plataformas software, en este caso se ha utilizado Arduino via Matlab-Simulink, este software nos ha permitido una poder trabajar con datos de una manera sencilla e intuitiva ya que se ha tenido un control directo del sistema desde la CPU.

Este trabajo tiene un amplio uso didáctico, ya que su sencilla implementación permite a los alumnos de las disciplinas de Automática de los distintos Grados de Ingeniería trabajar con conceptos como lazo abierto, lazo cerrado, función de transferencia, controlador PID, referencia, señal de control, error estacionario... viendo reflejado sus resultados en un medio físico. Lo cual es algo que nos permite afianzar conocimientos.

En concreto durante los cursos 2012-2013 y 2013-2014 ha permitido la realización de 2 prácticas de laboratorio, un para identificar los parámetros de la planta y otra al final del curso para diseñar un controlador PI, cerrar el lazo y controlar la

posición del balancín.

Por último destacar que un vez realizado el diseño, montaje y validación del prototipo inicial y en base al conocimiento adquirido, problemas surgidos y solventados, se ha procedido a la fabricación, en las propias dependencias del departamento, de más de treinta equipos.

VII. FUTURAS LÍNEAS DE TRABAJO

Como futuras líneas de trabajo se tiene previsto la inclusión de filtros para obtener una señal con menos ruido, también se pretende trabajar con el segundo método de diseño de Ziegler-Nichols. Otra futura línea de trabajo sería realizar una plataforma similar para diseñar controladores borrosos. Por último se está trabajando ahora en un controlador PID por eventos, de forma que se pueda controlar de forma remota modificando la señal de control únicamente si se produce algún cambio en el funcionamiento.

REFERENCIAS

- [1] T. Trash, "Getting stating with Arduino," Available: <http://arduino.cc/en/Guide/HomePage>.
- [2] T. Hirzel, "PWM", Available: <http://arduino.cc/en/pmwiki.php?n=Tutorial/PWM>
- [3] K. Ogata, *Ingeniería de Control Moderna*. PRENTICE-HALL, 2013.
- [4] J.G. Ziegler and N.B. Nichols, "Optimum settings for automatic controllers", Transactions of the ASME 64, pp. 759–768, 1942.
- [5] J. Garcia Tiscar, "Arduino+MATLAB/Simulink: controlador PID" Available: <http://wechoosethemoon.es/2011/07/21/arduino-matlab-simulink-controlador-pid/>, 2011.
- [6] MathWorks, "Arduino Support for Simulink", Available: <http://es.mathworks.com/hardware-support/arduino-simulink.html>
- [7] TR4NSDUC7OR, Tutorial de Arduino y MPU 6050, <http://robologs.net/2014/10/15/tutorial-de-arduino-y-mpu-6050/>
- [8] J.L. Mírez Tarrillo, "Control PID", http://jmirezcontrol.files.wordpress.com/2012/01/control_pid_jorge_mirez_2012.pdf, 2012, pp. 123–135.
- [9] B. Beauregard "Improving the Beginner's PID – Derivative Kick", <http://brettbeauregard.com/blog/2011/04/improving-the-beginner%E2%80%99s-pid-derivative-kick/>, 2012.
- [10] F. Morilla, "Controladores PID. Ajuste Empírico," http://www.dia.uned.es/~fmorilla/MaterialDidactico/ajuste_empirico.pdf 2006.
- [11] K. Astrom and T. Hagglund, "PID Controllers: Theory, design and tuning." Instrument Society of America (ISA), 2nd Edition 1994.
- [12] <http://es.mathworks.com/hardware-support/arduino-simulink.html>