

Diseño, construcción y estabilización de un sistema Bola-Balancín

B. A. López, N. C. Castillo, L. A. Cabrera^a, L. Vite^a, R. Villafuerte^{a,*}

^aÁrea Académica de Computación y Electrónica, Universidad Autónoma del Estado de Hidalgo, Mineral de la Reforma, Hgo., México

Resumen

En este trabajo se presenta el diseño y la construcción de una plataforma experimental Bola-Balancín. Para el diseño mecánico se emplea la herramienta CAD SolidWorks®, mientras que para la construcción física se utiliza una impresora 3D. Además, se aborda el problema de estabilización de dicha plataforma considerando desconocida la dinámica del sistema. La estrategia de control aplicada es la clásica ley de control PID, cuya sintonización se realiza empleando la segunda metodología propuesta por Ziegler-Nichols.

Palabras Clave: Bola-Balancín, Ley de control PID, Ziegler-Nichols.

1. Introducción

La plataforma experimental Bola-Balancín, comúnmente conocida como Ball & Beam por su nombre en inglés, es uno de los sistemas no lineales clásicos empleados para el diseño, sintetización e implementación de leyes de control. Tal como su nombre lo indica, la plataforma consiste en una bola localizada sobre un balancín, cuyo objetivo es posicionar la bola en un punto o trayectoria en particular mediante la acción de un motor acoplado al sistema. En las últimas décadas, diversas estrategias de control han sido aplicadas sobre la plataforma experimental, por ejemplo en (Hauser et al., 1992) se presenta un enfoque por realimentación linealizante para la estabilización y seguimiento de trayectorias. En (Wang and Mendel, 1992) se aplica una ley de control difusa para el sistema no lineal Bola-Balancín, en este caso los resultados son probados vía simulación. Por otra parte, en (Laukonen and Yurkovich, 1993) se realiza la identificación y control basada en la lógica difusa, reportando resultados vía experimentación. Otras estrategias de control tal como el control robusto se han aplicado a la plataforma considerando el problema de seguimiento de trayectorias, ver (Huang and Lin, 1995). Las redes neuronales también han sido aplicadas para la identificación del sistema no lineal, ver (Morioka et al., 1995), en ella se realiza una comparación con el control por modos deslizantes. Algunas investigaciones recientes se presentan en (Chen et al., 2012)-(Maalini et al., 2016), en estos trabajos se obtiene el modelo matemático del sistema aplicando el formalismo Euler-Lagrange, de la misma manera en (Vincenty

and Beauchamp, 2014) y (Valluru et al., 2016), la dinámica del sistema es obtenida mediante la segunda ley de Newton.

Tal como se ha descrito anteriormente, diversos estudios se basan en el modelado del sistema. Sin embargo, esto puede incrementar la dificultad en el diseño y sintonización de los controladores. A diferencia de los trabajos descritos con anterioridad, en esta investigación se opta por aplicar el segundo método propuesto por Ziegler-Nichols (Ziegler and Nichols, 1942). Este método tiene la finalidad de sintonizar las ganancias de un controlador Proporcional Integral Derivativo (PID) para estabilizar sistemas con ciertas características (Åström and Hägglund, 1995). Por lo cual, para este caso en particular se empleará para estabilizar la plataforma Bola-Balancín en lazo cerrado con el controlador PID, sin la necesidad de considerar el modelo matemático del sistema.

Otra particularidad de la presente propuesta, es que los prototipos Bola-Balancín existentes en el mercado tienen un costo muy elevado, motivo por el cual se propone el diseño y construcción de dicha plataforma experimental a un bajo costo, siendo así una sugerente alternativa para la enseñanza de la teoría de control en escuelas con recursos restringidos. En este caso, el diseño se realiza empleando el software SolidWorks, mientras que la impresión de las piezas se ha realizado en la máquina Anet-A8.

El presente trabajo se organiza de manera siguiente, en la Sección 2 se detalla la metodología para la sintonización del control PID empleando el segundo método de Ziegler-Nichols. En la Sección 3 se describe la plataforma experimental considerando la estructura mecánica y electrónica, además se explica la adquisición de datos mediante el programa Matlab-Simulink. La sintonización del controlador PID para la estabilización de

* Autor en correspondencia.

Correo electrónico: villafuerte@uaeh.edu.mx (R. Villafuerte)

sistema Bola-Balancín se presenta en la Sección 4, mientras que los resultados experimentales del sistema en lazo cerrado se muestran en la Sección 5. Finalmente el trabajo termina con las conclusiones en la Sección 6.

2. Preliminares

En esta sección se describe la estrategia de control PID, de la misma manera se presenta la sintonización del controlador mediante el segundo método propuesto por Ziegler-Nichols.

2.1. Control PID

Una estrategia de control clásica en el marco referencial de la teoría de control es el PID. Esta estrategia es ampliamente aplicada para la estabilización de sistemas con fines académicos e industriales debido a la efectividad y simplicidad en sus términos. Tal controlador calcula el error $e(t)$ existente entre la salida $y(t)$ de un sistema y un valor deseado $y_d(t)$, es decir

$$e(t) = y_d(t) - y(t),$$

con lo cual se genera una señal de control formada por una acción proporcional (P), una acción integral (I) y una acción derivativa (D). Aquí, la acción proporcional, tal como su nombre lo indica, depende proporcionalmente del error; la parte integral provee robustez ante perturbaciones constantes, eliminando el error en estado estacionario; mientras que la acción derivativa garantiza amortiguamiento para el sistema en lazo cerrado.

Una estructura estándar del controlador PID es expresada en la ecuación siguiente

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right), \quad (1)$$

donde $u(t)$ es la señal de control PID, K_p es la ganancia proporcional, T_i es la ganancia de tiempo integral y T_d es la ganancia de tiempo derivativa.

2.2. Método de Ziegler-Nichols

A continuación se describe la sintonización para la ley de control (1) mediante el segundo método propuesto por Ziegler-Nichols (Ziegler and Nichols, 1942).

1. En primer lugar se establecen los parámetros $T_i = \infty$ y $T_d = 0$.
2. Aplicando únicamente la ganancia K_p , se incrementa este valor desde cero hasta un valor crítico K_{cr} , es decir, cuando la salida del sistema presenta oscilaciones.
3. Posteriormente se determina el periodo crítico P_{cr} correspondiente a las oscilaciones obtenidas en el paso anterior, a partir de la cual se calculan las ganancias del controlador PID, de la misma manera es posible calcular las ganancias para el control proporcional (P) y el control proporcional integral (PI), tal como se muestra en la Tabla 1.

Tabla 1: Sintonización de controladores P, PI y PID por el segundo método de Ziegler-Nichols.

Tipo de controlador	K_p	T_i	T_d
P	$0.5 K_{cr}$		
PI	$0.45 K_{cr}$	$0.83 P_{cr}$	
PID	$0.6 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

3. Resultados principales

En la presente sección se describe el diseño y la construcción del prototipo Bola-Balancín considerando la estructura mecánica y electrónica. Además, se detalla la adquisición de señal mediante la tarjeta Atmega 328 junto con el programa Matlab-Simulink.

3.1. Diseño y construcción de la plataforma

En la Figura 1 se observa la plataforma experimental desarmada, a partir de la cual se describe cada una de las partes que la conforman.

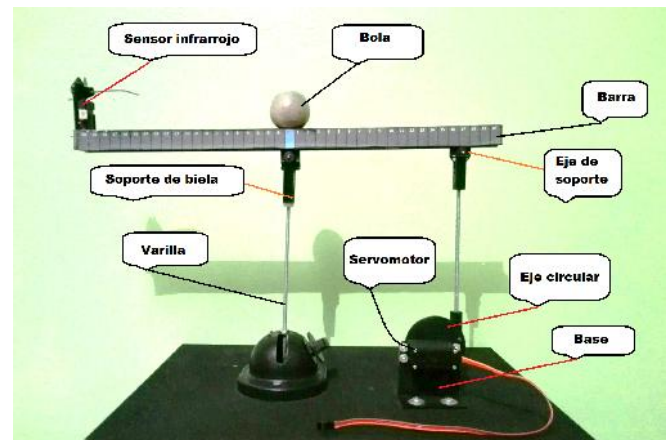


Figura 1: Plataforma experimental Bola-Balancín.

Se tiene una base rectangular sobre la cual se montan los componentes del sistema, en ella se colocan dos elementos en las que se insertan dos varillas metálicas, mismas que permiten sostener una barra horizontal. Esta barra metálica posee una ranura en la cual se inserta la bola permitiendo su movimiento con la acción de un servomotor acoplado. En un extremo de la barra horizontal se ubica el sensor infrarrojo.

A continuación se ilustran las piezas mecánicas empleadas en la construcción de la plataforma experimental. Las bases para el servomotor y el sensor son mostradas en la Figura 2. Mientras que en la Figura 3 se observa el eje circular colocado en el rotor del servomotor. En la misma figura también se tiene la biela para sostener la barra horizontal. El diseño de estas piezas fue realizado en la herramienta CAD SolidWorks.

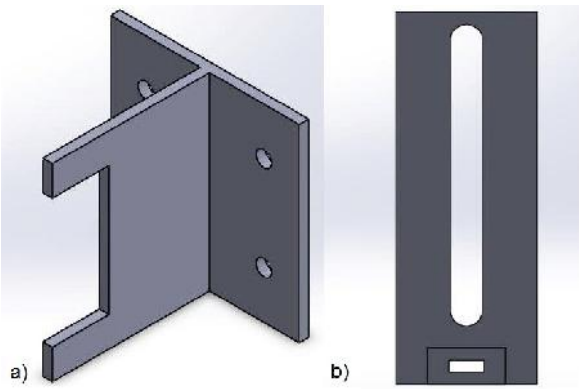


Figura 2: Bases para el servomotor a) y el sensor b).

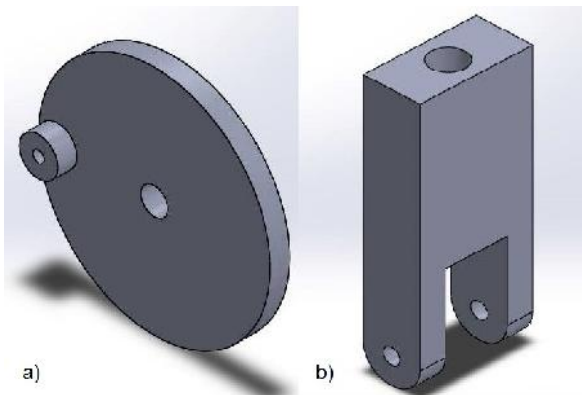


Figura 3: Extensión para el eje del servomotor a) y biela b).

La impresión 3D de las piezas descritas anteriormente fue realizada en la máquina Anet-A8 empleando ácido poliláctico.

Referente a la parte electrónica se tiene un sensor infrarrojo SHARP GP2Y0A41SK0F, una tarjeta Atmega 328 y un servomotor TowerPro MG996R. Cada uno de los elementos representa una parte esencial en el funcionamiento de la planta, por lo cual son detallados en los párrafos subsecuentes.

Sensor infrarrojo SHARP GP2Y0A41SK0F: Dispositivo encargado de enviar una señal de voltaje en función de la distancia en la que se encuentra cierto objeto. Este elemento es el encargado de medir la salida del sistema, cuyo rango de operación es de 4 a 30 cm.

Tarjeta de adquisición de datos: Se emplea la tarjeta Atmega 328 para adquirir la señal emitida por el sensor infrarrojo. La función de este elemento, es tomar una lectura constante del sensor a través del convertidor analógico digital (Analog Digital Converter, ADC). En este proceso se convierte la señal del sensor a distancia en milímetros, posteriormente la señal es enviada a la herramienta de programación Matlab-Simulink donde se encuentra el bloque de control PID. Subsecuentemente, la señal de control generada por el PID es enviada a la tarjeta de adquisición de datos a través de comunicación serial para aplicarla al actuador de la planta.

Servomotor TowerPro MG996R: Es el actuador del sistema el cual genera el movimiento angular de la barra metálica. Su operación esta subyugada por la tarjeta Atmega 328 mediante la

modulación de ancho de pulso (Pulse Wide Modulation, PWM) generada por el controlador PID.

3.2. Adquisición de señal

La señal analógica proveniente del sensor infrarrojo es recibida y procesada por el microcontrolador mediante un convertidor ADC de 10 bits. Dado que el sensor posee un comportamiento no lineal tal como se ilustra en la Figura 4, se realiza una interpolación lineal, la cual permite manipular la señal adquirida de manera adecuada.

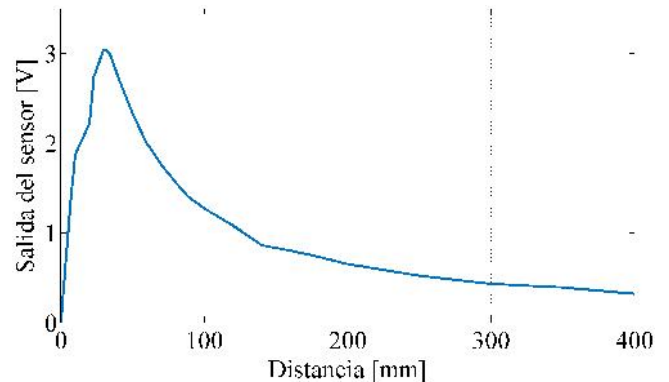


Figura 4: Comportamiento no lineal del sensor infrarrojo.

Para realizar la interpolación se emplea la función MAP incorporada en el lenguaje de programación C++, esto mediante dos vectores, uno para la señal de entrada, es decir, para los datos adquiridos por el sensor; y otro vector para la salida, el cual contiene la posición de la bola. Tal como se observa en el código fuente, es evidente que el segundo vector corresponde a la posición de la bola en milímetros, en este caso el origen se considera a 160 mm frente al sensor. Para esa distancia se tiene una lectura analógica de 0.942V, lo cual corresponde un valor de 193 con base en la escala del ADC de 10 bits.

Por otra parte, los datos son enviados al programa Matlab-Simulink mediante comunicación serial. Dado que en la comunicación serial sólo es posible transferir un byte a la vez, es necesario dividir el paquete de datos en dos partes, cada una de 8 bits. Sin embargo, antes de enviar la señal es necesario crear dos identificadores, uno para el encabezado y otro para finalizar la transmisión de cada muestra realizada.

El paquete de datos recibido en la herramienta Simulink es separado en 4 bytes, para lo cual se utiliza un bloque demultiplexor de 4 salidas, ver Figura 5.

Posteriormente se comprueba que la recepción de datos haya sido realizada de manera correcta mediante el empleo de bloques condicionales, de tal manera, una vez que se reconocen los identificadores de encabezado y de finalización, se procede con la manipulación de los datos intermedios, los cuales son los bytes que contienen la posición de la bola (1 Byte para el signo y otro Byte para la distancia).

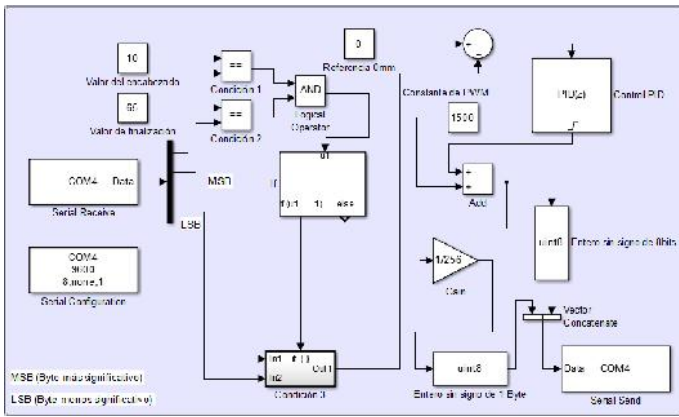


Figura 5: Diagrama a bloques para la recepción de datos en Simulink.

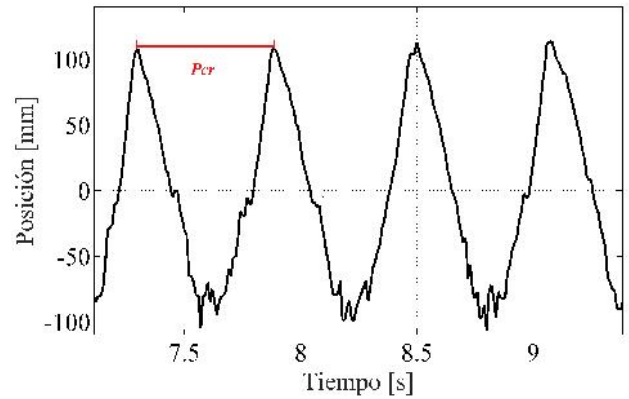


Figura 6: Respuesta marginalmente estable de la plataforma Bola-Balancín.

Para determinar el signo de la señal se emplea la siguiente condición: si el primer Byte es igual a una constante con valor decimal de 45, el segundo Byte que contiene la distancia es multiplicado por -1 , en dado caso que el primer Byte sea distinto de 45 se considera que dicho dato es positivo. Una vez procesada la señal recibida, se genera un error $e(t)$ entre la referencia $y_d(t)$ y la salida $y(t)$ del sistema. El error generado es empleado por el controlador PID implementado en el entorno Matlab-Simulink. Consecuentemente, la señal de control PID es sumada con una constante con valor decimal de 1500, el cual es el valor en micro segundos del PWM donde la barra se posiciona en forma horizontal. Para enviar la señal de control PID a la tarjeta Atmega, es necesario separarla en dos Bytes dado que supera los 8 bits y por lo tanto no sería posible enviarla correctamente mediante comunicación serial, por lo que primero se envía el Byte más significativo y en segunda instancia se envía el Byte menos significativo.

En la siguiente sección se presenta la sintonización del controlador PID aplicada en la estabilización la plataforma experimental Bola-Balancín.

4. Sintonización del controlador PID

Aplicando la metodología propuesta por Ziegler-Nichols, la plataforma experimental desarrollada es marginalmente estable con una ganancia crítica $K_{cr} = 3$, mientras que el periodo crítico es $P_{cr} = 0,59$, ver Figura 6.

El control elegido para la estabilización de la plataforma experimental es el PID, esto con la finalidad de eliminar el error en estado estacionario y además reducir las oscilaciones del sistema. Las ganancias obtenidas con base en la Tabla 1 son las siguientes

$$\begin{aligned} K_p &= 1,8, \\ T_i &= 0,295, \\ T_d &= 0,073. \end{aligned}$$

La implementación del controlador PID se ha realizado mediante la herramienta Matlab-Simulink junto con la tarjeta Atmega 328, los resultados obtenidos se muestran en la sección subsecuente.

5. Resultados experimentales

La respuesta del sistema en lazo cerrado con el controlador PID para tareas de estabilización es ilustrada en la Figura 7.

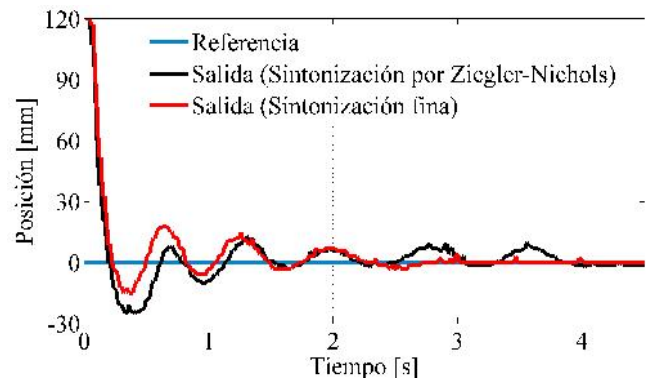


Figura 7: Salida del sistema en lazo cerrado con la ley de control PID.

Tal como se observa en la figura anterior, la prueba experimental consiste en la estabilización de la bola considerando una condición inicial $y(0) = 120 \text{ mm}$, dicha tarea es lograda en un tiempo inferior a 3 segundos. Cabe mencionar que las ganancias aplicadas al controlador han sido ligeramente ajustadas en base a la experiencia del usuario (sintonización fina), esto considerando el propósito que tiene cada una de las ganancias K_p , T_i y T_d . En este caso, los valores ajustados son

$$K_p = 1,8, \quad T_i = 0,295 \quad \text{y} \quad T_d = 0,12,$$

los cuales no varían considerablemente con respecto a los valores obtenidos en la sintonización por Ziegler-Nichols, sin embargo, la salida de sistema con las ganancias ajustadas presenta una mejora con respecto a las primeras ganancias.

Por otra parte, la señal de control es mostrada en la Figura 8, en este caso la señal corresponde a los micro segundos en estado alto necesarios para generar la modulación de ancho de pulso. Cabe mencionar que con un valor de 1500 micro segundos el eje del servomotor se encuentra en una posición horizontal, es decir a cero grados. Con un valor de 1000 micro segundos el

eje se coloca en -90 grados, mientras que con un valor de 2000 micro segundos el eje del servomotor se coloca en 90 grados.

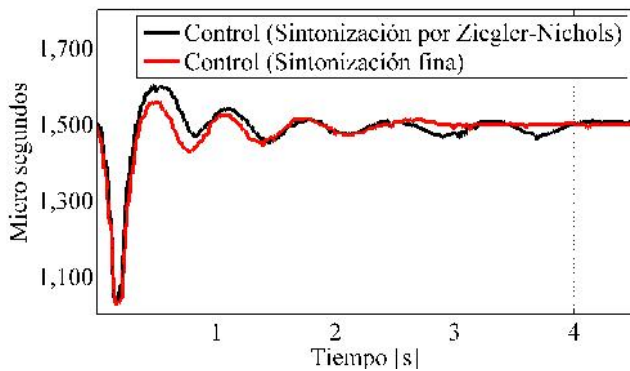


Figura 8: Señal en estado alto para la modulación de ancho de pulso.

6. Conclusiones

Se ha diseñado y construido una plataforma experimental Bola-Balancín de bajo costo. Mediante la implementación de la ley de control PID en la plataforma desarrollada se han obtenido resultados experimentales aceptables. Para la sintonización del controlador se ha empleado el segundo método propuesto por Ziegler-Nichols, con lo cual se ha logrado estabilizar el sistema subactuado Bola-Balancín sin la necesidad de recurrir al modelo matemático del sistema.

Aplicando la metodología propuesta por Ziegler-Nichols es relativamente sencillo sintonizar un controlador PID, lo cual reduce considerablemente los costos y tiempos en ciertas aplicaciones académicas e industriales. De igual manera, se puede observar que éste método cuenta con algunas desventajas, una de ellas es que no todos los sistemas pueden tener una respuesta marginalmente estable, por lo tanto el método de Ziegler-Nichols puede emplearse únicamente en ciertas plataformas.

Con respecto al diseño mecánico, cabe destacar que las piezas de la plataforma experimental Bola-Balancín son de suma importancia, por lo tanto es necesario que dichas piezas sean de buena calidad y de cortes finos. Además, en el ensamble de la estructura se debe reducir la fricción entre los componentes, por ello la mayoría de las piezas de dicho sistema son impresas en 3D empleando ácido poliláctico, con lo cual se reducen los costos, se obtienen piezas de mejor calidad, durabilidad y amigables al medio ambiente.

English Summary

Design, construction and stabilization of a Ball-Beam system.

Abstract

In this work the design and construction of an experimental platform Ball & Beam are presented. For the mechanical design the CAD SolidWorks® tool is applied, whereas for the physical

construction a 3D printer is employed. In addition, the stabilization problem of this platform is approached considering the dynamics of the system unknown. The control strategy applied is the classical PID control law, whose tuning is performed using the second methodology proposed by Ziegler-Nichols.

Keywords:

Ball-Beam, PID, Ziegler-Nichols.

Referencias

- Åström, J., Hägglund, T., 1995. *Pid controllers: theory, design, and tuning*. Isa Research Triangle Park.
- Chen, W., Sui, X., Xing, Y., 2012. Modeling and modulation of nonlinear ball-beam system controller based on matlab. In *Fuzzy Systems and Knowledge Discovery (FSKD)*, 2388–2391.
- Hauser, J., Sastry, S., Kokotovic, P., 1992. Nonlinear control via approximate input-output linearization: The ball and beam example. *IEEE transactions on automatic control* 37, 392–398.
- Huang, J., Lin, C., 1995. Robust nonlinear control of the ball and beam system. In *American Control Conference, Proceedings of the 1995* 1, 306–310.
- Laukonen, E., Yurkovich, S., 1993. A ball and beam testbed for fuzzy identification and control design. In *American Control Conference*, 665–669.
- Maalini, P., Prabhakar, G., Selvaperumal, S., 2016. Modelling and control of ball and beam system using pid controller. In *Advanced Communication Control and Computing Technologies (ICACCCT)*, 322–326.
- Morioka, H., Wada, K., Sabanovic, A., 1995. Neural network based chattering free sliding mode control. *Proceedings of the 34th SICE Annual Conference*, 1303–1308.
- Valluru, S., Singh, M., Singh, S., 2016. Prototype design and analysis of controllers for one dimensional ball and beam system. In *Power Electronics, Intelligent Control and Energy Systems (ICPEICES)* 1, 1–6.
- Vincenty, C., Beauchamp, G., 2014. Modelling the ball-and-beam system from newtonian mechanics and from lagrange methods. In *Twelfth Latin American and Caribbean Conference for Engineering and Technology*.
- Wang, L., Mendel, J., 1992. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE transactions on Neural Networks* 3, 807–814.
- Ziegler, J., Nichols, N., 1942. Optimum settings for automatic controllers. *Trans. ASME*.

Apéndice A. Código fuente

Se anexa el código fuente para la transmisión y recepción de datos entre la tarjeta Atmega 328 y el software Matlab-Simulink. El lenguaje de programación está basado en C++ bajo el entorno de desarrollo Arduino.

```
#include <Servo.h>
int sensorPin=0; //Pin analógico para el sensor infrarrojo
int medida=0; //Variable que guarda el mapeo entre el voltaje adquirido y la referencia
int distancia=0; //Variable para guardar la distancia de la bola
int PWM_us=0; //Variable que guarda el tiempo para el ciclo de trabajo en el PWM
int puntos_distancia[]={120, 90, 60, 40, 0, -40, -60, -90, -120}, valores_sensor[]={70, 90, 119, 138, 193, 286, 332, 450, 778};
unsigned long cont1=0, cont2=0; //Contadores para el tiempo de realimentación
byte a=0; //Variable que lee los caracteres
Servo myservo; //Se crea el objeto 'myservo'
void setup()
{
```

