

# Real-time control prototype and its use in Automatic Control courses.

J. Maldonado, R. Garrido, G. Castro

*Departamento de Control Automático, CINVESTAV-IPN, Ciudad de México, México (e-mail: jmaldonado,garrido,zavala@ctrl.cinvestav.mx).*

---

**Abstract:** In this article, a prototype for executing experiments in real-time and its use as a tool to improve Automatic Control courses is presented. The main components of the platform are an STM32F4 Discovery development kit made by STMicroelectronics, a LEGO servo motor, and an in/out board created at CINVESTAV. The code for the STM32F4 is programmed in a laptop using Matlab/Simulink or C++ software. The performance of the platform is measured by using the algorithm PID implemented in discrete and continuous time.

*Keywords:* Educational prototypes, Real-Time Control, PID control, Discrete PID.

---

## 1. INTRODUCTION

En un estudio de 2016 sobre perspectivas económicas para América Latina [OCDE, 2016], realizado por la Organización para la Cooperación y el Desarrollo Económico (OCDE) se observa un incremento salarial considerable entre los graduados de las carreras CTIM (Ciencia, Tecnología, Ingeniería y Matemáticas), o STEM [Sanders, 2008] en sus siglas en inglés. Lo anterior se debe a la creciente demanda de nuevas tecnologías y el aumento de la automatización en trabajos repetitivos [Daron, 2017]. En México, desde hace algunos años, el gobierno trabaja en conjunto con varias universidades para mejorar las habilidades de los alumnos, para esto se ha propuesto incluir metodologías de enseñanza no tradicionales como la Educación STEM [Bybee, 2010],[Villareal, 2017], [SEP, 2018]. Esta forma de enseñanza plantea que las áreas que la componen, la Ciencia, Tecnología, Ingeniería y Matemáticas, son dependientes entre ellas por lo que deben enseñarse en forma conjunta. Propone que los conocimientos teóricos sean puestos en práctica usando por ejemplo nuevas tecnologías, aprendizaje activo [Freeman et al., 2014], entre otros.

Tomando en cuenta lo anterior, en carreras como Robótica, Mecatrónica, y Control Automático se incluyen materias que hacen uso de prácticas de laboratorio para combinar conocimiento teórico y práctico [IPN, 2018], estas prácticas pueden realizarse usando simulaciones numéricas (SN) y/o Plataformas Experimentales (PE).

Las simulaciones numéricas se ejecutan mediante *software* especializado [Gilat, 2004],[Jeffrey, 2006] o utilizando una red de servidores. Las PE se pueden clasificar como plataformas *in situ* y remotas. Las plataformas remotas son prototipos físicos a los cuales se accede a través de internet [Hahn and Spong, 2000],[Parkin et al., 2002],[Kaluz

et al., 2015]. En este caso el estudiante no tiene interacción directa con el prototipo, y usualmente suelen ser equipos con costos elevados y/o con pocas unidades. Por otro lado, las plataformas experimentales *in situ* son prototipos físicos con los que el estudiante tiene interacción directa. Estas pueden ser comerciales como las mostradas en [Apkarian and Astrom, 2004], [Bernstein, 2005]. Normalmente, el costo elevado es una de las mayores desventajas para las plataformas comerciales, debido a que las universidades generalmente requieren varias. Por esta razón se han creado alternativas de bajo costo como las señaladas en [Araújo et al., 2015],[Binugroho et al., 2015],[Sobota et al., 2013],[Tsuruta and Sawada, 2016]. Considerando lo anterior, el objetivo de este trabajo es presentar un prototipo basado en una tarjeta de desarrollo embebida, un servomotor marca LEGO y una tarjeta de entrada-salida, cuyo propósito es el de apoyar la enseñanza del Control Automático.

El artículo se divide de la siguiente manera, la Sección 2 describe el prototipo y sus componentes, la Sección 3 muestra los algoritmos de control implementados en el prototipo y la Sección 4 reporta los experimentos realizados en tiempo real con una versión del controlador Proporcional Integral Derivativo (PID) descrito en tiempo continuo y en tiempo discreto, y la configuración experimental usada. Al final se expresan observaciones y conclusiones.

## 2. DESCRIPCIÓN DEL PROTOTIPO

Los componentes del prototipo, denominado en lo sucesivo Kit DV1.0 (ver Fig. 1), son los siguientes:

- Tarjeta Discovery STM32F04.
- Tarjeta de entrada-salida H1D1A-V1 de diseño propio.

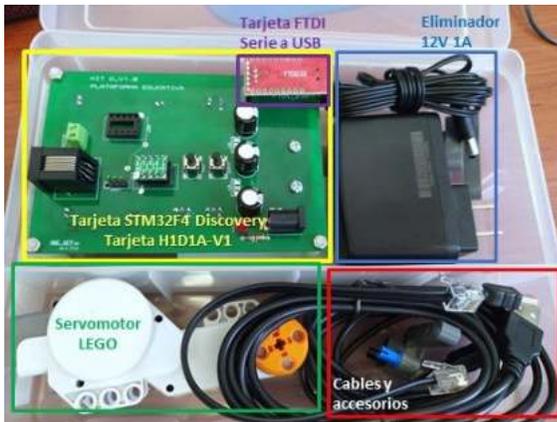


Figura 1. Plataforma Educativa.



Figura 2. Tarjeta STM32F4 Discovery

- Servomotor LEGO.
- Tarjeta FTDI de conversión de protocolo serie a protocolo USB.
- Eliminador 12 V, 1 A.
- Cables, accesorios y caja.

Los programas requeridos:

- MatLab 2015A.
- STM32 ST-LINK utility.
- FTDI USB driver.

### 2.1 Tarjeta STM32F4 Discovery

La tarjeta STM32F04 Discovery, mostrada en la Figura 2, tiene un microcontrolador STM32F407VGT6 y un procesador ARM Cortex M4F [STMicroelectronics, 2017]. Posee un reloj interno que es capaz de realizar interrupciones y un procesador que realiza operaciones de punto flotante lo que permite la ejecución de algoritmos en tiempo real. Algunas de sus características son:

- 3 puertos CAD (Convertidor Analógico-Digital) de 12 bits.
- 2 puertos CDA (Convertidor Digital-Analógico) de 12 bits.

Ventajas	Desventajas
Bajo costo (menor a los 500 pesos mexicanos)	Memoria FLASH limitada
Bajo consumo energético	Falta de información en comparación con tarjetas más populares (Arduino, Raspberry Pi)
Implementación de algoritmos usando Simulink	Curva de aprendizaje lenta si se usa programación en C++
Programación basada en C++	Las librerías que se usan en Simulink se descargan e instalan por separado

TABLA 1. Ventajas y desventajas de la tarjeta STM32F407 Discovery.

- 12 Temporizadores, los más importantes son:
  - 2 temporizadores de modulación de ancho de pulso PWM (*pulse-width modulation*) para control de motores.
  - 2 temporizadores de 32 bits.
- Interfaces de comunicación:
  - 4 USARTS (*Universal Synchronous / Asynchronous Receiver Transmitter*) o Transmisor-Receptor Síncrono/Asíncrono Universal.
  - 2 UARTS (*Universal Asynchronous Receiver Transmitter*) o Transmisor-Receptor Asíncrono Universal.

La Tabla 1 resume las ventajas y desventajas de la tarjeta.

### 2.2 Tarjeta H1D1A-V1

La tarjeta de entrada-salida H1D1A-V1 mostrada en la Figura 3 ha sido creada en el Departamento de Control Automático del CINVESTAV. La tarjeta STM32F4 Discovery se monta mecánicamente sobre la tarjeta H1D1A-V1 y de esta manera se realizan las conexiones de señales entre ambas tarjetas. Este diseño modular permite reemplazar esta tarjeta por otra con un amplificador de ancho de pulso diferente y/o de más potencia, o para el control simultáneo de dos servomotores. La H1D1A-V1 contiene los siguientes elementos:

- Puente H: El circuito usado para puente H es el DRV8838 de Texas Instruments [Instruments, 2016] el cual permite frecuencias de conmutación de 250 Khz de una señal PWM. Este circuito convierte la señal de PWM que proviene del algoritmo de control ejecutado en la tarjeta STM32F4 en una señal de mayor voltaje y corriente para controlar la velocidad y dirección del servomotor LEGO.
- Fuente de alimentación: La fuente de alimentación utiliza un adaptador de corriente de 12V 1A, capacitores y un regulador de voltaje LM7809.
- Comunicación serie: La comunicación entre la tarjeta STM32F4 Discovery y la computadora que ejecuta MatLab/Simulink se realiza a través de una interfaz

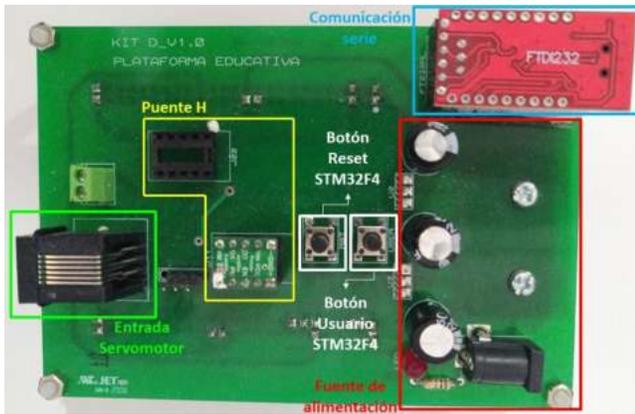


Figura 3. Tarjeta entrada-salida H1D1A-V1.

Velocidad	160-170 RPM
Rango de Movimiento	360° Grados
Voltaje	9V
Resolución del codificador óptico	180 ppr
Peso	0.04 Kg

TABLA 2. Características técnicas del servomotor LEGO EV3

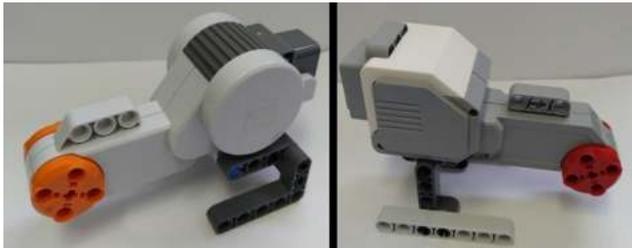


Figura 4. Servomotor LEGO NXT (izquierda) y EV3 (derecha)

serie-USB FTD1. Esta permite observar la posición angular del servomotor en el entorno de Simulink.

- Conexión con el servomotor LEGO. Permite la conexión entre el circuito DRV8838 y el servomotor, y entre la señal del codificador óptico que proviene del servomotor LEGO y la tarjeta STM32F4 Discovery.

### 2.3 Servomotor LEGO

Se han probado dos modelos de servomotores: LEGO EV3 Grande y LEGO NXT (ver Fig. 4). Sus características se muestran en la Tabla 2. Los servomotores usan como sensor de posición angular la señal enviada por un codificador óptico que está acoplado al eje del motor mediante un engrane plástico.

### 2.4 Costo del prototipo

Como se observa en la Tabla 3, el Kit DV1.0 tiene un precio del orden de los \$2000 pesos mexicanos el cual

Descripción	Cantidad	Precio (MXN)
STM32F407G-DISC1	1	500
Hardware on Top (HAT) H1D1A-V1	1	450
Tarjeta FTDI Serial a USB	1	60
Servomotor LEGO EV3	1	550
Eliminador 12 V 1 A	1	70
Cable Mini-USB	2	130
Cable Motor	1	35
Caja de Plástico	1	120
<b>TOTAL</b>		<b>1915</b>

TABLA 3. Costo de los componentes del Kit Educativo

puede cambiar de acuerdo a la paridad peso mexicano-dólar estadounidense.

## 3. DISCRETIZACIÓN DEL ALGORITMO DE CONTROL PID

### 3.1 Ley de control en tiempo continuo

El modelo del prototipo sin considerar la dinámica eléctrica del servomotor es [Ogata, 2010]:

$$\ddot{y} = -a\dot{y} + bu \quad (1)$$

donde  $y$  es la posición angular del motor y  $u$  la señal de control. El parámetro  $a$  está asociado a la fricción viscosa, a los efectos de la fuerza contra-electromotriz del servomotor y el término  $b$  es función de la ganancia del amplificador de potencia, de la constante de par y de la resistencia de armadura. Ambos términos son función inversa de la inercia del servomotor. Transformando (1) a una función de transferencia se obtiene

$$\frac{Y(s)}{U(s)} = \frac{b}{s^2 + as} \quad (2)$$

Se definen  $Y(s) = \mathcal{L}\{y\}$  y  $U(s) = \mathcal{L}\{u\}$ , siendo  $\mathcal{L}\{\cdot\}$  el operador de Laplace.

El algoritmo Proporcional Integral con realimentación de velocidad está descrito por [Ang et al., 2005],[Astrom and Hagglund, 1995]

$$U(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau - K_d \dot{y} \quad (3)$$

siendo los términos  $K_p$ ,  $K_i$  y  $K_d$ , las ganancias proporcional, integral y derivativa respectivamente. El error de posición se define como  $e(t) = r(t) - y(t)$  donde  $r(t)$  es la posición deseada. La función de transferencia de la ley de control (3) es:

$$U(s) = K_p E(s) + \frac{1}{s} K_i E(s) - K_d s Y(s) \quad (4)$$

donde  $E(s) = R(s) - Y(s)$ .

### 3.2 Discretización de la ley de control usando el método de Euler

Para implementar leyes de control en sistemas embebidos es necesario obtener sus versiones en tiempo discreto. El método de Euler se usa para este fin. La ley de control PID (3) se modifica de la siguiente manera:

$$U(s) = K_p E(s) + \frac{K_i}{s} E(s) - \frac{\alpha s}{s + \alpha} K_d Y(s) - K_{pr} Y(s) \quad (5)$$

Se agrega el término  $K_{pr} Y(s)$  para modificar los polos en lazo cerrado independientemente del cero que también se genera en lazo cerrado. Además, el filtro pasa altas  $\alpha s / (s + \alpha)$  produce estimados de la derivada de la posición angular  $y$  del motor.

El método de discretización de Euler requiere la sustitución en el controlador PID (5) de la variable compleja  $s$  por la siguiente expresión:

$$s = \frac{1 - q^{-1}}{h} \quad (6)$$

Donde  $h$  es el periodo de integración y  $q^{-1}$  es el operador retardo. Realizando la sustitución se puede demostrar que la discretización queda de la manera siguiente:

$$\begin{aligned} u(k) &= K_p e(k) + K_i a(k) - K_d b(k) - K_{pr} y(k) \\ e(k) &= r(k) - y(k) \\ a(k) &= a(k-1) + h e(k) \\ b(k) &= c(k) + \alpha y(k) \\ c(k) &= \frac{1}{1 + \alpha h} [c(k-1) - \alpha^2 h y(k)] \end{aligned} \quad (7)$$

### 3.3 Discretización usando el método Trapezoidal

En este caso se utiliza la sustitución:

$$s = \frac{2}{h} \left( \frac{1 - q^{-1}}{1 + q^{-1}} \right) \quad (8)$$

la cual produce la siguiente discretización del controlador (5):

$$\begin{aligned} u(k) &= K_p e(k) + K_i a(k) - K_d b(k) - K_{pr} y(k) \\ e(k) &= r(k) - y(k) \\ a(k) &= a(k-1) + \frac{h}{2} (e(k) + e(k-1)) \\ b(k) &= c(k) + \alpha y(k) \\ c(k) &= \frac{2 - \alpha h}{2 + \alpha h} c(k-1) - \frac{\alpha^2 h}{2 + \alpha h} (y(k) + y(k-1)) \end{aligned} \quad (9)$$

## 4. EXPERIMENTOS EN TIEMPO REAL

### 4.1 Configuración experimental

La configuración experimental para utilizar el prototipo se muestra en la Figura 5. Para implementar un algoritmo de control se procede de la siguiente manera:

- Se codifica el algoritmo de control deseado en Simulink usando la librería de “Waijung” [Co, 2017].

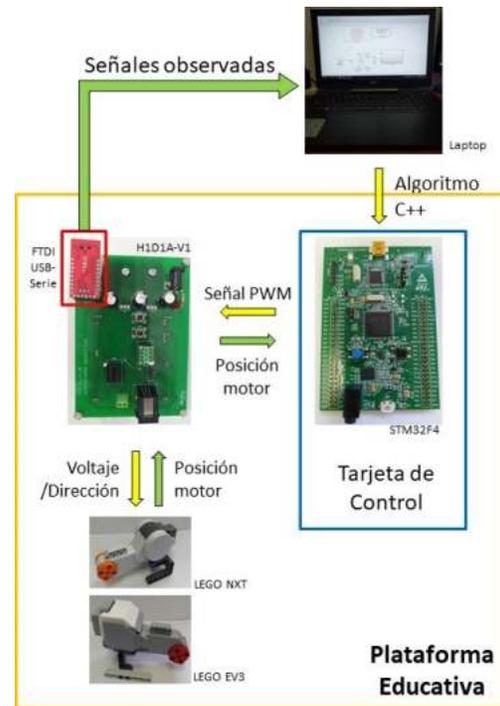


Figura 5. Configuración experimental.

- Se conecta la tarjeta STM32F4 Discovery vía un primer puerto USB a la computadora que ejecuta Simulink.
- Se presiona el botón “Build Model” en Simulink para descargar e instalar el algoritmo en la tarjeta STM32F4 Discovery.
- Una vez energizado el prototipo, se visualizan las señales de posición angular y de control mediante la conexión serie-USB utilizando la tarjeta FTDI conectada a un segundo puerto USB.

El periodo de muestreo es de 1 ms y el algoritmo de integración cuando la discretización la realiza automáticamente Simulink corresponde al método Euler01. Se utiliza el siguiente filtro pasa altas para obtener un estimado  $\dot{y}_e$  de  $\dot{y}$  usando la posición angular del codificador óptico.

$$sY_e(s) = \mathcal{L}\{\dot{y}_e\} = \frac{150s}{s + 150} \mathcal{L}\{y\} \quad (10)$$

Para medir el desempeño de los controles discretizados se emplean la Integral del Error Cuadrático (IEC), la Integral del Valor Absoluto de la Señal de Control (IVABSC) y la Integral del Valor Absoluto de la Variación de la Señal de Control (IVAVSC).

Periodo de muestreo	Índice	Método de discretización		
		Euler01 MatLab	Euler	Trapezoidal
$h = 0.001$	IEC	312.9962	312.8742	311.9981
	IVABSC	499.6103	502.8109	499.8488
	IVAVSC	89.4514	85.6993	86.195
$h = 0.01$	IEC	356.1206	309.9015	317.6116
	IVABSC	503.5326	503.9695	500.6677
	IVAVSC	100	50.9462	50.4282

TABLA 4. Tabla de índices de desempeño

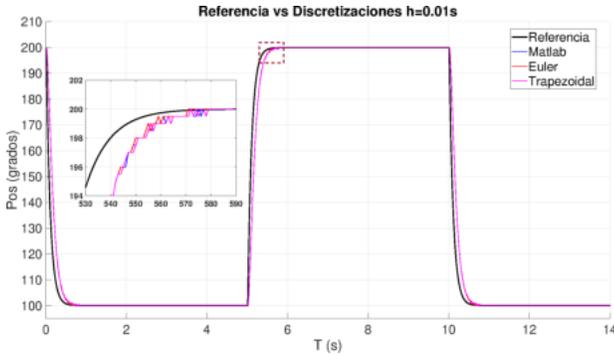


Figura 6. Posición vs Referencia usando el tiempo de muestreo  $h = 0.001$

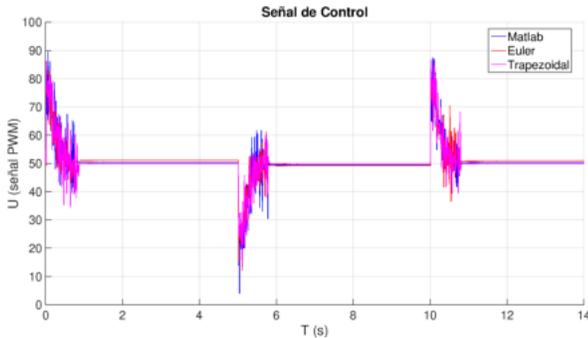


Figura 7. Señal de Control usando el tiempo de muestreo  $h = 0.001$

$$\begin{aligned}
 IEC &= \int_{T_1}^{T_2} e^2 dt \\
 IVABSC &= \int_{T_1}^{T_2} |u(t)| dt \\
 IVAVSC &= \int_{T_1}^{T_2} \left| \frac{d[u(t)]}{dt} \right| dt
 \end{aligned} \quad (11)$$

#### 4.2 Resultados experimentales

Los valores de los índices de desempeño para los controladores implementados (5), (7) y (9) se encuentran en la Tabla 4. Se observa que cuando el período de muestreo es  $h = 0.001$  los valores de los índices son muy parecidos y

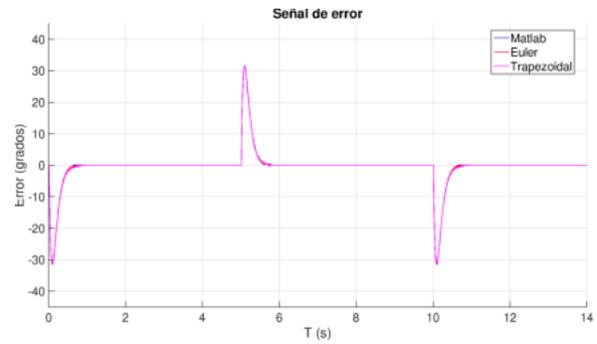


Figura 8. Señal de error usando el tiempo de muestreo  $h = 0.001$

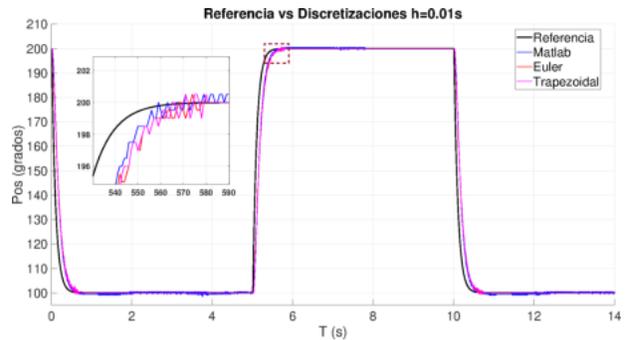


Figura 9. Posición vs Referencia usando el tiempo de muestreo  $h = 0.01$

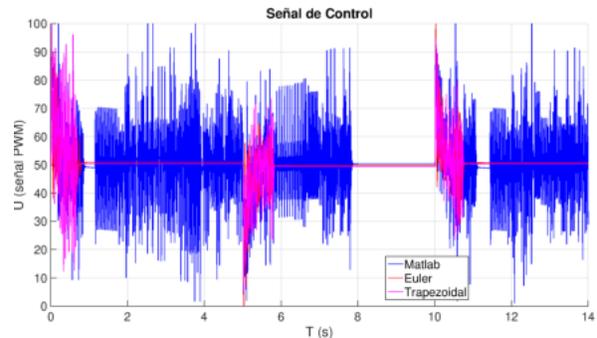


Figura 10. Señal de Control usando el tiempo de muestreo  $h = 0.01$

no existen entre ellos cambios significativos. Se observa en las Fig. 6, 8 y 7 el desempeño de los tres controladores. Se observan cambios significativos en los índices IEC e IVAVSC cuando el periodo de muestreo se incrementa a  $h = 0.01$ . Esto es visible en las señales de error mostradas en Figura 11 y en las señales de control mostradas en la Figura 10. Cabe mencionar que la señal que se indica como “Matlab” corresponde a la discretización que realiza automáticamente Matlab/Simulink.

#### 5. CONCLUSIONES

Los resultados mostrados permiten concluir que el prototipo propuesto puede servir como una herramienta de

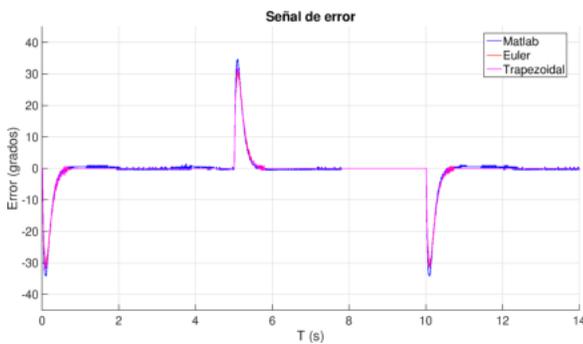


Figura 11. Señal de error usando el tiempo de muestro  $h = 0.01$

apoyo para la enseñanza del Control Automático en los niveles de licenciatura y posgrado. Como trabajo futuro se contempla la implementación de controladores no lineales y la evaluación del prototipo mediante prácticas dirigidas a estudiantes de posgrado.

#### ACKNOWLEDGEMENTS

Los autores agradecen la ayuda de Jesús Meza Serrano en la puesta en funcionamiento del prototipo. Se agradece el apoyo al primer autor por medio de una beca doctoral CONACyT-MEXICO.

#### REFERENCIAS

- Ang, K.H., Chong, G., and Li, Y. (2005). Pid control system analysis, design and technology. *IEEE Transactions on control systems technology*, 559–576.
- Apkarian, J. and Astrom, K. (2004). A laptop servo for control education. *IEEE control systems*, 24, 70–73.
- Araújo, A., Portugal, D., Couceiro, M.S., and Rocha, R.P. (2015). Integrating arduino-based educational mobile robots in ros. *Journal of Intelligent & Robotic Systems*, 77(2), 281–298.
- Astrom, K. and Hagglund, T. (1995). *PID Controllers*. Instrument Society of America.
- Bernstein, D.S. (2005). The quanser dc motor control trainer. *IEEE Control Systems Magazine*, 3, 90–93.
- Binugroho, E.H., Pratama, D., Syahputra, A.Z.R., and Pramadihanto, D. (2015). Control for balancing line follower robot using discrete cascaded pid algorithm on adroit v1 education robot. In *2015 International Electronics Symposium (IES)*, 245–250. IEEE.
- Bybee, R.W. (2010). Advancing stem education: A 2020 vision. *Technology and engineering teacher*, 70(1), 30.
- Co, A. (2017). Waijung blockset. Accessed at <http://waijung.aimagin.com/index.htm> Enero-2020.
- Daron, A. (2017). La automatización y el futuro del empleo. Accessed at <https://focoeconomico.org/2017/11/01/la-automatizacion-y-el-futuro-del-empleo/>.
- Freeman, S., Eddy, S.L., McDonough, M., Smith, M.K., Okoroafor, N., Jordt, H., and Wenderoth, M.P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23), 8410–8415.
- Gilat, A. (2004). *MATLAB: An Introduction with Applications 2nd Edition*. John Wiley & Sons.
- Hahn, H.H. and Spong, M.W. (2000). Remote laboratories for control education. *Proceeding of the 39th IEEE: Conference on Decision and control*, 39, 895–900.
- Instruments, T. (2016). *DRV883x Low-Voltage H-Bridge Driver*.
- IPN (2018). Mapa curricular de la carrera de ingeniería en comunicaciones y electrónica. Accessed at <https://www.esimez.ipn.mx/assets/files/esimez/docs/inicio/Slider/2020/Enero/Plandeestudios.pdf>.
- Jeffrey, T. (2006). *LabVIEW for everyone : graphical programming made easy and fun*. Prentice Hall.
- Kaluz, M., Garcia-Zubia, J., et al. (2015). A flexible and configurable architecture for automatic control remote laboratories. *IEEE Transactions on Learning Technologies*, 8, 299–310.
- OCDE, CEPAL, C. (2016). *Perspectivas económicas de América Latina 2017: Juventud, competencias y emprendimiento*. OECD Publishing.
- Ogata, K. (2010). *Modern Control Engineering*. Prentice Hall, 5 edition.
- Parkin, R. et al. (2002). A pid servo control system experiment conducted remotely via internet. *Mechatronics*, 12, 833–843.
- Sanders, M.E. (2008). Stem, stem education, stemmania. SEP (2018). ¿qué es niñas stem pueden? Accessed at <https://www.gob.mx/sep/articulos/ques-ninas-stem-pueden>.
- Sobota, J., Pišl, R., Balda, P., and Schlegel, M. (2013). Raspberry pi and arduino boards in control education. *IFAC Proceedings Volumes*, 46(17), 7–12.
- STMicroelectronics (2017). *UM1472 User manual*.
- Tsuruta, K. and Sawada, S. (2016). Development of servo controller using arm32. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 1–6. IEEE.
- Villareal, E. (2017). Fortalecer enseñanza stem factor crítico en México. Accessed at <https://www.consumotico.mx/steam/fortalecer-ensenanza-stem-factor-critico-en-mexico/>.