

ANÁLISIS DE COMPORTAMIENTO DE HARDWARE OPEN SOURCE PARA ADQUISICIÓN DE DATOS EN MATLAB-SIMULINK®

Romero Cruz, Lucero Monserrat; Rosas González, Alicia; Herrera López, Mily Margarita;
Romero Galván, Gerardo; Ampudia Ramírez, Federico.

Universidad Autónoma de Tamaulipas
Centro de Innovación Tecnológica en Electricidad y Electrónica
Departamento de Posgrado. Edificio H2. Unidad Académica Multidisciplinaria Reynosa-Rodhe
Carretera Reynosa - San Fernando, cruce con Canal Lateral, Col. Arcoíris, C.P. 88779. Reynosa, Tamaulipas.
(899) 921 33 00 ext. 8116
a2163728012@alumnos.uat.edu.mx / federico.ampudia@gmail.com

1. RESUMEN

Este análisis documenta los resultados del comportamiento de un sistema de control diseñado en el *software* MATLAB-SIMULINK®, el cual consiste en computar señales de entrada de variables físicas externas para después procesarlas de acuerdo a las técnicas de control seleccionadas, graficando los resultados obtenidos y estableciendo una comparación de valores de entrada con respuesta esperada y valores de salida. En este caso se ha configurado un control PID.

Para lograr la adquisición de datos de variables externas se utiliza el *hardware* comercial de acceso libre y bajo costo ARDUINO® modelo DUE, el cual reúne las características técnicas necesarias para acoplarse con MATLAB-SIMULINK®.

El objetivo es analizar el comportamiento de todo el sistema, evaluando la respuesta de procesamiento en tiempo real del *hardware* ARDUINO®, estableciendo el grado de pertinencia para su aplicación en proyectos de control que así lo requieran y delimitando sus alcances como plataforma de adquisición de datos de bajo costo.

Palabras Clave: MATLAB-SIMULINK®, ARDUINO®, Teoría de Control, Adquisición de datos, Bajo costo.

ABSTRACT.

This analysis reports the results of the behavior of a control system designed in the MATLAB-SIMULINK® software, which consists of computing input signals of external physical variables and then processing them according to the selected control techniques, graphing the results obtained and establishing a comparison of input values, expected response and output values. In this case, a PID control is configured. To obtain the data acquisition of external variables, the commercial hardware of free access and low cost ARDUINO® model DUE is used, which has the necessary technical characteristics to be coupled with MATLAB-SIMULINK®. The objective is to analyze the behavior of the entire system, evaluating the response in real time of ARDUINO® hardware processing, establishing the degree of relevance for its application in control projects that require it and delimiting its scope as a data acquisition platform for low cost.

Keywords: MATLAB-SIMULINK®, ARDUINO®, Control Theory, Data Acquisition, Low Cost.

2. INTRODUCCIÓN

Con el avance que la tecnología ha experimentado en el siglo XXI, las fronteras entre lo real y lo onírico comienzan a desvanecerse poco a poco. Las propuestas que los fecundos escritores del decimonónico formularan, casi a manera de disparate, se hacen realidad en los renglones de las comunicaciones, los medios de transporte, la calidad y esperanza de vida, la tecnología médica e inclusive la conquista del espacio exterior.

En el segmento de lo cotidiano, en la actualidad no se conciben algunas tareas rutinarias sin el uso de elementos como la computadora, el teléfono celular inteligente o la televisión [13]. Al mismo tiempo, la dinámica académica ha transmutado con el advenimiento de la modernidad y ha convertido al internet en uno de los pilares de apoyo con recursos de última generación soportando a las bibliotecas virtuales, las redes sociales, las plataformas educativas y el almacenamiento de audio y video, de manera colectiva, para utilizarlos como herramientas en el Proceso Enseñanza Aprendizaje [8].

Aunado a todo lo anterior, el campo de la investigación científica no está exento y ha visto un cambio vertiginoso en la forma de llevar a cabo las tareas propias de la búsqueda de información y generación de nuevo conocimiento.

Puntualmente, en el marco de la presente investigación, el área de Ingeniería Electrónica ha experimentado un cambio en el desarrollo de prototipos y pruebas con el advenimiento de *software* especializado en simulaciones. Estos permiten aproximar, de manera muy apegada a la realidad, las suposiciones técnicas con datos propuestos, obteniendo un resultado calculado a través de una realidad no tangible y simulada en ambientes virtuales [10].

En este caso, tal alcance se obtiene a través del *software* de simulación matemática y mecatrónica llamado MATLAB-SIMULINK®, mismo con el que se pueden diseñar distintas configuraciones para controlar sistemas proyectados y cuyas herramientas son capaces de generar datos y gráficas que permitan la interpretación del comportamiento del sistema ante perturbaciones determinadas [11].

MATLAB-SIMULINK® posee la capacidad de procesar variables físicas reales externas como señales de entrada. Estas son de naturaleza diversa y para lograrlo es necesario contar con un *hardware* que pueda sensor, procesar y enviar las variables recibidas al software a través de una tarjeta para adquisición de datos [6]. En este caso se propone utilizar el *hardware* de acceso libre y bajo costo llamado ARDUINO® con el objetivo de evaluar su desempeño y con ello establecer el grado de pertinencia para utilizar la mencionada plataforma en pruebas, abatiendo costos económicos sin comprometer la calidad técnica del proyecto de investigación que se esté llevando a cabo [15].

3. METODOLOGÍA

La dinámica general se basa en diseñar un programa de control en el *software* MATLAB-SIMULINK® en cuya terminal de entrada al sistema se contemple una variable de origen externo obtenida a través de la tarjeta de adquisición de datos ARDUINO® previa diligencia para comunicar ambas plataformas. El programa procesará las señales de entrada y ejercerá las funciones determinadas dentro del *software* acorde a lo establecida. El resultado es una señal de salida que puede ser visualizada en forma de gráfica. La configuración del programa permite apreciar todos los datos a través de un arreglo de opciones con los cuales se pueden mostrar los valores graficados en tiempo real de las señales permitiendo establecer el análisis y las comparaciones pertinentes.

3.1. MATLAB-SIMULINK®

Para un proyecto que significa adquirir datos a través de un *hardware* y mostrarlos en una interfaz gráfica en tiempo real es necesario un equipo de cómputo que posea características de procesamiento de gama media a alta. Esto se debe a que los requerimientos técnicos para el correcto funcionamiento del sistema demandan características específicas [1,3].

MATLAB® es un *software* de simulación matemática ampliamente utilizado en investigación y desarrollo. Posee una serie de herramientas que facilitan la dinámica de procesar operaciones y cálculos a través de algoritmos y programación. Para este artículo se ha utilizado la versión MATLAB® R2018b – 64 bits.

Complementariamente, se encuentra disponible una extensión de las funciones que este *software* ofrece, se trata del entorno de simulación virtual SIMULINK®, mismo que permite construcción de modelos y sistemas a través bloques con funciones específicas. Aunado a ello, también ofrece todo tipo de instrumentos de medición y monitoreo, además de otras utilidades. La versión de la interfaz de simulación electrónica utilizada es SIMULINK® Versión 9.2 (R2018b) [12].

Este conjunto es especialmente utilizado para investigaciones relativas al espectro de la Teoría del Control debido a sus características técnicas, su capacidad de procesar señales de variables externas y también la ventaja de poder trabajar con variables en tiempo real [15].

3.2. ARDUINO®

Como ya fue abordado, el *hardware* que se utiliza para adquisición de datos es el ARDUINO®, perteneciente a una compañía que trabaja con el sistema *open source* que diseñan y manufacturan placas de desarrollo de *hardware* para implementar dispositivos digitales de control enfocados en facilitar el uso de la electrónica y programación de sistemas embebidos para proyectos de índole multidisciplinaria basándose en el concepto DIY (*Do It Yourself*).

Estos dispositivos usan el microcontrolador Atmel AVR® bajo la configuración conocida como sistema mínimo realizada sobre una placa de circuito impreso a la cual es posible conectar módulos de expansión a través de los puertos de entrada y salida presentes en el modelo que se esté utilizando. Generalmente los dispositivos ARDUINO® pueden ser, tanto energizados como comunicados, bien por un puerto USB o bien por alimentación externa.

Para programar funciones se hace uso del *Bootloader* propio del AVR a través de una interfaz de programación instalada en computadora que funciona usando comunicación serie. El lenguaje de programación está basado en C++.

El modelo de tarjeta de adquisición de datos utilizado para esta investigación es el DUE, mismo que posee un microcontrolador de 32 bit CortexM3 ARM el cual puede ser programado mediante el IDE de ARDUINO® y cuya característica es que se puede incrementar la velocidad de procesamiento. Posee 54 pines digitales de entrada y salida, de los cuales 12 pueden ser usados como salidas PWM, además de 12 entradas análogas, 2 salidas análogas, 4 UART (puertas seriales por *hardware*), un cristal oscilador de 84 MHz, una conexión compatible con USB-OTG, 2 TWI, *jack* de poder, conexión JTAG, botón reset y un botón para borrar [15].

3.3. PROTOCOLO DE COMUNICACIÓN

Para establecer la comunicación entre el *software* SIMULINK® y el *hardware* ARDUINO®, se realiza el siguiente procedimiento: Una vez ejecutado el *software* MATLAB®, en la pantalla principal hacer click en el ícono *Add-Ons* en la barra principal superior en la sección *Home*, seleccionando *Get Hardware Support Packages* (Fig. 1).

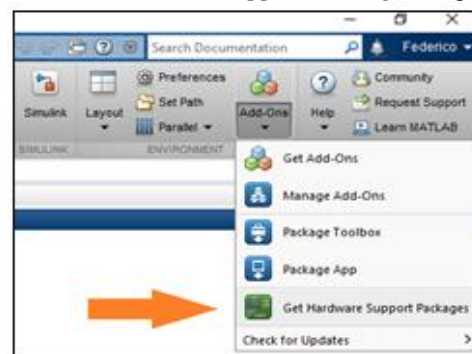


Figura 1: Configuración de *hardware*.
Fuente: Elaboración propia en MATLAB®.

En la ventana emergente que se despliega motivo de la selección de la Figura 1, deben de ser seleccionados los complementos llamados *MATLAB Support Package for Arduino Hardware*, así como el complemento llamado *Legacy MATLAB and Simulink Support for Arduino* y por último se deberá agregar el complemento llamado *Simulink Support Package for Arduino Hardware* (Fig. 2).

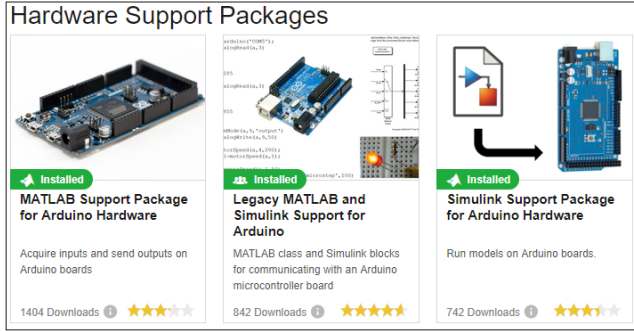


Figura 2: Paquetería de soporte MATLAB®-ARDUINO®.
 Fuente: Elaboración propia en MATLAB®.

El objetivo de estas librerías es que el *software* reconozca el dispositivo para adquisición de datos para que se pueda establecer la comunicación entre ellos en configuración *plug and play*. De esta manera, cada vez que un dispositivo ARDUINO® sea conectado al puerto USB de la computadora, esta automáticamente lo reconocerá y las condiciones para su operación estarán disponibles para utilizarse.

Una vez finalizadas las instalaciones de las paqueterías-librerías que permiten la comunicación entre SIMULINK® y ARDUINO®, se procede a verificar que los complementos hayan sido instalados. Esto se efectúa en la ventana principal de SIMULINK®, en la barra de herramientas, en *View* → *Library Browser*. En la lista desplegada, aparecerá la sección *Simulink Support Package for Arduino Hardware*.

En la Figura 3 se describe que una vez que se tiene listo el circuito en SIMULINK® se debe de dar click en *Tools* → *Run on target hardware* → *Prepare to run*.

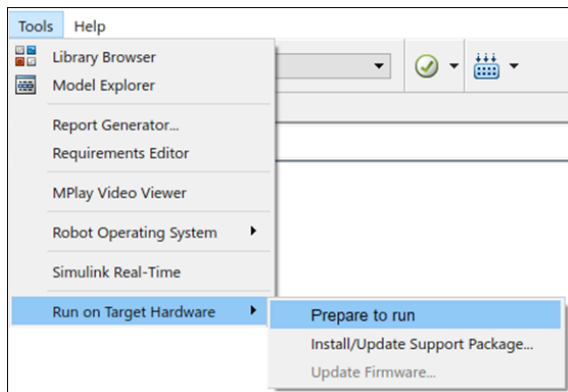


Figura 3: Configuración de adquisición de datos.
 Fuente: Elaboración propia en SIMULINK®.

En la ventana emergente llamada *Configuration Parameters* en la sección llamada *Hardware Implementation* localizada del lado izquierdo de la ventana se deberá elegir el *hardware* con el cual se va a trabajar, que para este caso es el ARDUINO® DUE (Fig. 4). Después de la selección se deberá hacer click en el botón *Apply*.

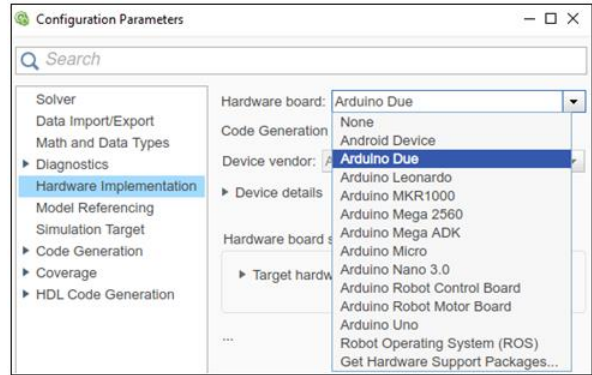


Figura 4: Selección de ARDUINO DUE® como *hardware*.
 Fuente: Elaboración propia en SIMULINK®.

Después de haber seleccionado el *hardware* y habiendo hecho click en *Apply*, dentro de la misma ventana llamada *Configuration Parameters* se elegirá la sección llamada *Solver* localizada del lado izquierdo de la ventana en la parte superior de esta (Fig. 5).

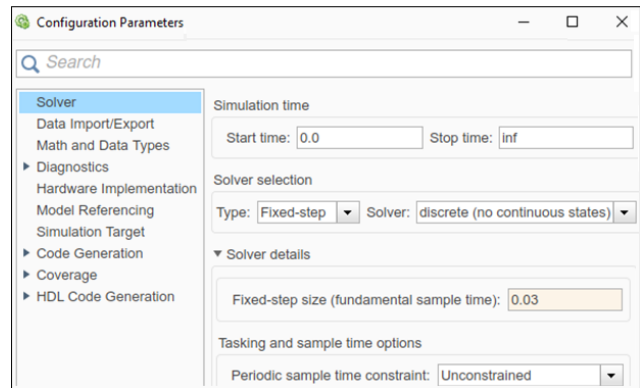


Figura 5: *Configuration parameters* → *Solver*.
 Fuente: Elaboración propia en SIMULINK®.

En la referida ventana *Configuración parameters* se selecciona el tiempo de ejecución de la acción generada por todo el sistema, tanto *software* como *hardware*. Para este caso se han determinado que el tiempo sea infinito, tecleando “inf” en la casilla *Stop time*. Por otro lado, en la sección *solver selection* debe de ser elegido para la opción *Type: Fixed-step* mientras que en la opción *Solver* se debe de elegir la opción *discrete (no continuous states)*. En la casilla *Fixed-step size (fundamental sample time)* el valor asignado es de 0.03 y corresponde al tiempo de muestreo para la adquisición de las señales y el barrido de actualización del proceso.

Estos últimos dos parámetros de selección corresponden a las características del procesamiento de las señales del circuito implementado en SIMULINK®.

En el caso de la opción *Solver*, esta hace referencia directa al método de resolución de ecuaciones diferenciales ordinarias. Para esto se presentan varios tipos de algoritmos que pueden representar una solución acorde al tipo de ecuaciones que se tengan por resolver.

En general, se sabe que todos los *solvers* de ODE de MATLAB® pueden resolver sistemas de ecuaciones con la forma $y' = f(t, y)$ [7].

Por otro lado se citan ejemplos como el del *solver* ode23s mismo que solo puede resolver problemas con una matriz de masas si la matriz de masas es constante.

De este modo ode15s y ode23t pueden resolver problemas con una matriz de masas singular, conocidas como ecuaciones diferenciales algebraicas (DAE) [4,5].

En cuanto a modelos con estados continuos, ode45 es un versátil *solver* de ecuaciones diferenciales ordinarias y es el primero que se debe de proponer para su uso para la mayoría de los problemas. No obstante, si el problema es rígido o requiere una alta precisión, entonces existen otros *solvers* de ODE que podrían estar mejor adaptados al problema.

Una vez que se ha seleccionado todo lo concerniente a la ventana *Configuration Parameters*, se procede a hacer click en el botón *OK*.

3.4. PROGRAMA E INTERCONEXIÓN FÍSICA

El circuito que se debe construir en SIMULINK® (Fig. 6) es a través del cual, una vez configurados los parámetros pertinentes, se adquirirán los datos de variables externas y por medio de cuya interface se podrán observar y documentar las gráficas de comportamiento de todo el sistema.

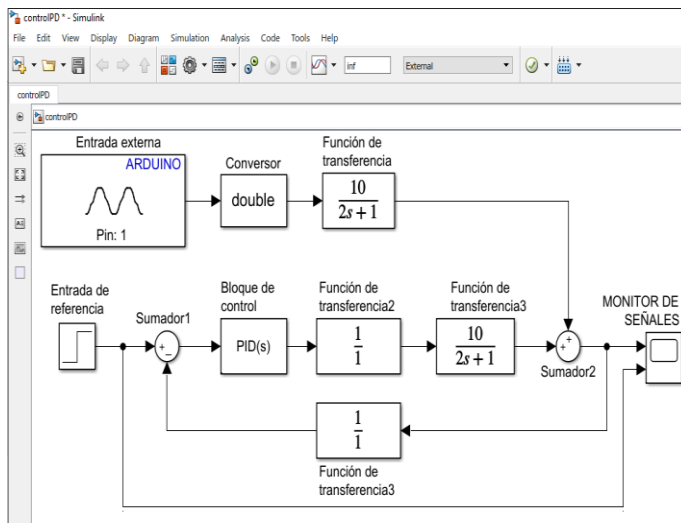


Figura 6: Circuito de control en SIMULINK® propuesto para evaluar desempeño de tarjeta de adquisición de datos.

Fuente: Elaboración propia en SIMULINK®.

Cabe destacar que el circuito de la Figura 6 es un arreglo clásico de control PID con dos señales de entrada, una es la ENTRADA DE REFERENCIA, mientras que la otra es la ENTRADA EXTERNA en la cual se envía la perturbación desde la tarjeta de adquisición de datos [2, 6, 9]. La propuesta que se presenta en este trabajo es que la perturbación sea una variable externa que el usuario pueda manipular a voluntad y cuyos efectos desestabilicen al sistema, mismo cuyo esquema de control deberá corregir la señal hasta el SET POINT indicado en la mencionada ENTRADA DE REFERENCIA. El objetivo es evaluar el desempeño de una señal externa para comparar la respuesta que tiene el sistema y con base en ello determinar el alcance que posee la plataforma ARDUINO® referente a la velocidad de adquisición y procesamiento de datos [11, 14]. El PID tiene valores P=2, I=1 y D=1, SET POINT se ubica en el valor de 3 en el tiempo T = 5.

En el circuito presentado en SIMULINK® destaca un elemento identificado como *Analog Input*, el cual es un bloque para recepción de señales externas de tipo analógicas provenientes del ARDUINO®. El bloque recibe la tensión como valor digital (0-1023 de mínimo a máximo). La tensión máxima está determinada por el voltaje de referencia de la entrada analógica que se tenga en el circuito electrónico físico, que en este caso se ha determinado como 5VCD..

La interconexión física se puede apreciar en la Figura 7, misma que es muy sencilla y se basa en la variación de un voltaje utilizando una resistencia variable. Las terminales de los extremos de esta se conectan a 5VCD y a GND de la misma tarjeta de adquisición de datos, sin una polaridad específica, mientras que la terminal central, la de control, será ingresada a la terminal de entradas analógicas A1 de la plataforma ARDUINO®. Esta será conectada a través del puerto USB a la computadora donde resida el programa de control diseñado en MATLAB-SIMULINK® [15].

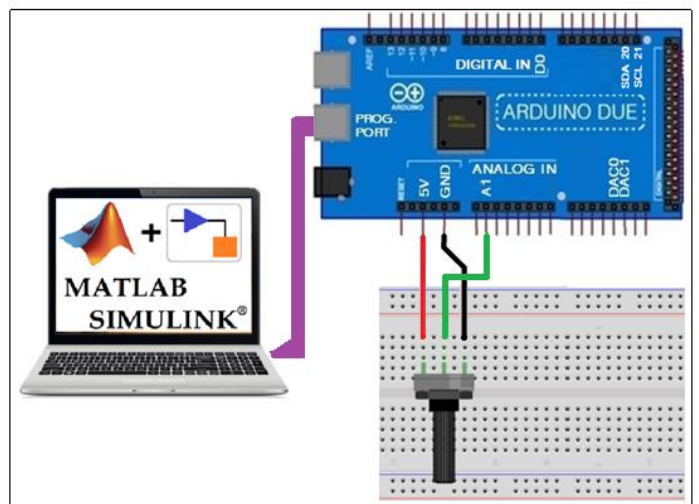


Figura 7: Circuito de interconexión física entre ARDUINO® y MATLAB-SIMULINK®

Fuente: Elaboración propia.

4. RESULTADOS

La interconexión del programa con la tarjeta de adquisición de datos de variables físicas externas se logra configurando los parámetros establecidos para este propósito. Siendo así se obtiene como resultado el trabajo de un sistema que computa datos en tiempo real y que responde a las perturbaciones externas también en tiempo real. Este comportamiento se puede apreciar a través del monitor de señales, interface que posee el mismo *software* y en donde se puede dar lectura a los resultados obtenidos.

La Figura 8 muestra el funcionamiento del sistema de control en configuración PID con valores $P = 2$, $I = 1$ y $D = 1$ con *SET POINT* en el valor de 3 indicada en el tiempo $T = 5$. Se puede observar como el proceso se lleva a cabo de manera normal no existiendo manipulación de variables a través de los datos adquiridos externamente. El valor de la señal se vuelve a estacionar en el *SET POINT* en $T = 12.1$. Este comportamiento es el convencional para un sistema de control.

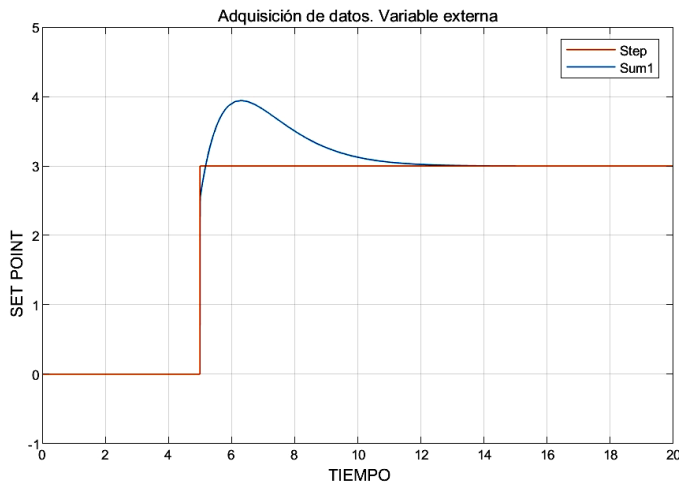


Figura 8: Comportamiento del sistema de control PID sin perturbación externa.
 Fuente: Elaboración propia.

En la Figura 9 se observa el comportamiento del sistema bajo las mismas condiciones de *SET POINT* y T descritas en la Figura 8. En esta prueba se aplica una perturbación establecida por el usuario a través del sistema de adquisición de datos externos utilizando ARDUINO®, manipulando el potenciómetro con el cual produce la señal externa que se refleja en el sistema en tiempo real.

Dicha alteración se realiza cuando $T = 6$ y el sistema está en pleno procesamiento para rectificar el valor de la señal retronándolo al valor establecido originalmente en $T = 12.9$. En este caso se observa una alteración que se describe como una fuerza contraria a la que el sistema intenta corregir. Posteriormente, el sistema retoma sus funciones normales y retorna a la señal al valor establecido.

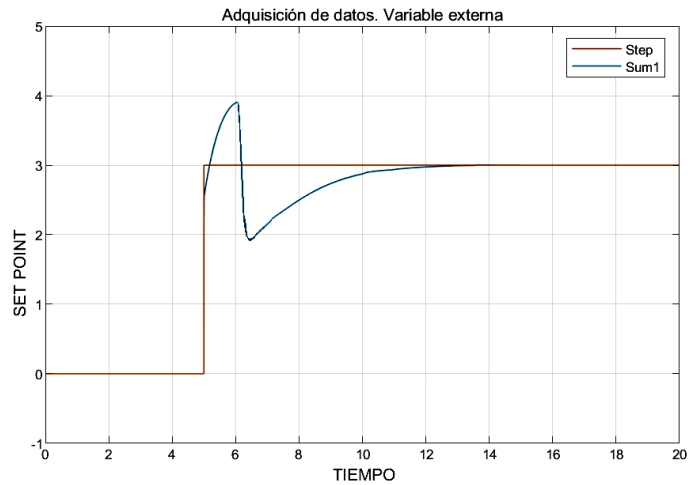


Figura 9: Comportamiento del sistema de control PID con perturbación externa en $T=6$.
 Fuente: Elaboración propia.

En la Figura 10 se aprecian una vez más una perturbación realizada por el usuario, en esta ocasión se realizan a partir del valor $T = 6$ y $T = 7$. El comportamiento observado es la tendencia a regresar al valor establecido, en esta ocasión alcanzándolo en $T = 14.8$.

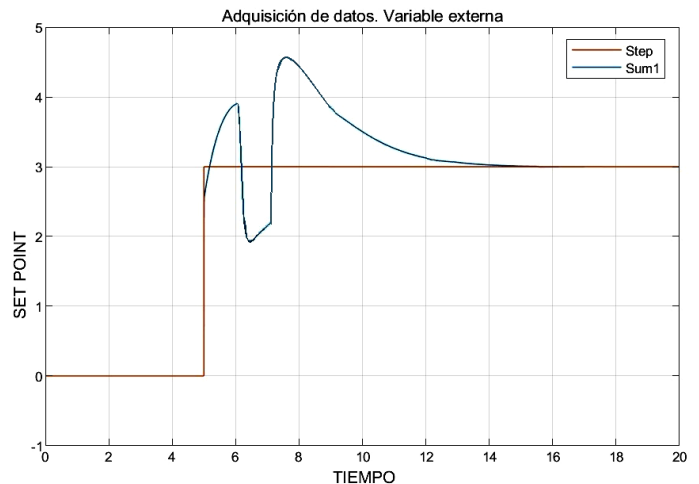


Figura 10: Comportamiento del sistema de control PID con perturbación externa en $T=6$ y $T=7$.
 Fuente: Elaboración propia.

5. CONCLUSIONES

Los datos de las variables externas que conforman las señales de entrada al sistema de control diseñado en MATLAB-SIMULINK® han sido adquiridos a través del hardware de acceso libre y bajo costo ARDUINO®. Los resultados

obtenidos después de realizar las pruebas de comportamiento presentadas en las Figuras 8 a 11 revelan que la dinámica de la obtención y procesamiento de señales se realiza en tiempo real con una respuesta inmediata y sin retrasos perceptibles significativos o nocivos en el sistema de control propuesto. Tomando esto en consideración, se puede aseverar que no existe obstáculo que impida que la mencionada plataforma pueda ser considerada para utilizarse en simulaciones con condiciones controladas y cuyo alcance se circunscriba a los efectos del ejercicio del control tipo PID, como el mostrado en el presente trabajo de documentación teórico-práctica.

Se debe de puntualizar que el desempeño satisfactorio del ARDUINO® para este caso, supone contar con la opción de adquirir datos de variables externas de manera confiable y a bajo costo. No se descarta que un sistema más complejo diseñado en MATLAB-SMULINK® requiera de alcances mayúsculos que no pueda proveer el *hardware* propuesto utilizado en este análisis. Sin embargo, como conclusión general para esta propuesta el resultado fue satisfactorio.

6. TRABAJOS FUTUROS

Una vez analizado que el comportamiento del hardware es pertinente para los sistemas propuestos, se establece que este trabajo de investigación es el punto de partida para continuar utilizando la plataforma *low cost* en diseños de sistemas embebidos o *hardware* intercomunicado, con aplicaciones tanto al amplio espectro del tema de la Teoría de Control como al de Telecomunicaciones.

7. REFERENCIAS

- [1] Al-Hammouri, A., Nordstrom, L., Chenine, M., Vanfretti, L., Honeth, N. y Leelaruji, R. (2012). Virtualization of synchronized phasor measurement units within real-time simulators for smart grid applications. *Power and Energy Society General Meeting, 2012 IEEE*: 1–7. doi:10.1109/PESGM.2012.6344949.
- [2] Åström K.J. y Hägglund, T. (1995) New tuning methods for PID controllers. *Proceedings of the 3rd European Control Conference*, p.2456–62.
- [3] Bin, Lu, Xin, Wu, Figueroa, H. y Monti, A. A Low-Cost Real-Time Hardware-in-the-Loop Testing Approach of Power Electronics Controls. *IEEE Transactions on industrial electronics*, vol. 54, no. 2, april 2007. 919 a 931.
- [4] Cale, J., Johnson, B., Dall'Anese M., Young, P, Duggan, G., Bedge P., Zimmerle, D. y Holton, L. Mitigating. Communication Delays in Remotely Connected Hardware-in-the-Loop Experiments. *IEEE transactions on industrial electronics*, vol. 65, no. 12, december 2018. 9739 a 9748.
- [5] Cebi, A., Guvenc, L., Demirci, M., Karadeniz, C., Kanar, K y Guraslan, E. "A low cost, portable engine electronic control unit hardware-in-the-loop test system", *Proceedings of the IEEE International Symposium on Industrial Electronics, 2005*.
- [6] Dorf R. y Bishop, R. (2005) *Sistemas de control moderno*. 10ma Edición. México: Pearson.
- [7] Hdez.-Gaviño, R. (2010) *Introducción a los sistemas de control*. Madrid: Pearson.
- [8] Jensen, R. (2000) *Metodología de la investigación*. México: Vanity SelFish.
- [9] Kuo, B. (1996) *Sistemas de control automático* 7ma. Ed. México: Prentice-Hall.
- [10] Levine, W. (2000) *Control System Applications*. New York: CRC Press.
- [11] Ogata, K. (2002) *Ingeniería de Control Moderna* 5ta. Ed. México: Pearson.
- [12] Pérez, C. (2002) *MATLAB y sus aplicaciones en la ingeniería*. México: Prentice-Hall
- [13] Reyes, F., Cid, J. y Vargas, E.(2015) *Mecatrónica, Control y Automatización*. México: Alfaomega.
- [14] Roca-Cusidó, A.(1999) *Control de procesos*. México: Alfaomega.
- [15] Torrente, J. (2013) *Arduino: Curso práctico de formación*. México: Alfaomega