

iiDAQ: An Affordable Data Acquisition System as a Tool for Teaching-Learning Process in Instrumentation and Control *

Lizbeth González-Arrieta * Roberto G. Ramírez-Chavarría *
Lizeth Torres * Areli Resendiz-Quintanar *

* Instituto de Ingeniería, Universidad Nacional Autónoma de México,
Ciudad de México 04510, México.
(e-mails: LGonzalezA@iingen.unam.mx, RRamirezC@iingen.unam.mx,
FTorresO@iingen.unam.mx, rqa1926@gmail.com)

Abstract: Teaching-learning process in undergraduate courses requires alternative methods to succeed, even more in a blended mode. Particularly, for instrumentation and control courses, practical activities are mandatory to strengthen the theoretical content. However, accessing to laboratory workbench and devices is not straightforward due to several constraints. In this work, we introduce the so-called iiDAQ, a low-cost and affordable solution as a data acquisition (DAQ) device devoted to didactic practices. The system is based on open hardware devices to acquire and generate signals from sensors and actuators. Moreover, we develop a full Matlab compliant toolbox with a graphical user interface (GUI) interaction and a low-level method scripting capabilities. We show the iiDAQ design and details on its functionality. The performance is shown in three didactic case studies. Results show that the proposal is an attractive tool to leverage teaching-learning processes for undergraduate instrumentation and control courses.

Keywords: Data acquisition, Instrumentation and Control, System Identification, Adaptive Control, Open Source

1. INTRODUCCIÓN

La adquisición de datos (DAQ: *data acquisition*), es actualmente una de las herramientas más útiles e importantes en la instrumentación y control de todo tipo de procesos (Blanco et al., 2014). Al ser una tendencia tecnológica tan significativa (González et al., 2018), muchos de los sistemas DAQ están disponibles sólo como herramientas de gran escala, principalmente enfocados a aplicaciones y soluciones industriales. Aunque existen sistemas DAQ comerciales para fines académicos, su principal limitante es el costo del producto y licenciamiento, el cual no es accesible para todas las instituciones de educación superior, ni para los estudiantes. Por ello, las actividades prácticas han sido reemplazadas por simulaciones, las cuales, permiten al estudiante tener apenas un mínimo acercamiento tangible al fenómeno real que se desea estudiar, presentando una gran problemática debido a la falta de elementos con los cuales el estudiante pueda experimentar y, al mismo tiempo, el docente pueda emplear para reforzar la enseñanza de los contenidos que inminentemente lo requieren. Se cree que las propuestas de solución para

estos sistemas deben tener como directriz herramientas de hardware y software de fácil acceso (John Park, 2003), que permitan a los alumnos, a partir de un entorno ya conocido, familiarizarse con el sistema de manera rápida e intuitiva. Para esto, existen múltiples plataformas hardware de acceso abierto, las cuales han cobrado relevancia en los últimos años para el desarrollo de proyecto y prototipos (Enériz et al., 2020; Aunkaew et al., 2021). Así entonces, se decidió realizar una primera aproximación en MATLAB lo que no limita que en un futuro se pudiese migrar a una plataforma de software libre, dadas las ventajas que estas representan. Así entonces, presentamos el sistema iiDAQ, un dispositivo de generación y adquisición de datos, accesible y de bajo costo, para ser controlado desde el del entorno de programación de Matlab™, inspirados en la problemática del proceso de enseñanza-aprendizaje de instrumentación y control, así como la adquisición de datos como herramienta fundamental del proceso (Morris and Langari, 2012). El desempeño del sistema se ilustra en tres casos de estudio, personalizados de acuerdo con una aplicación específica: i) medición de temperatura con sensor resistivo, ii) controlador proporcional-integral (PI) de voltaje en un capacitor, y iii) controlador adaptable de un motor de corriente directa (DC: *direct current*).

* Trabajo realizado con el apoyo del Programa UNAM-DGAPA-PAPIME PE101222

2. SISTEMA iIDAQ

2.1 Funcionamiento general

El sistema de adquisición de datos iIDAQ es un sistema electrónico que permite adquirir y generar señales, a partir de un microcontrolador y comunicación serial mediante una computadora personal. iIDAQ es capaz de obtener la información requerida por el usuario para la implementación de un gran número de aplicaciones, principalmente a las que están orientadas en el aprendizaje de tópicos de instrumentación y control. Así mismo, se cuenta con un *toolbox*. Éste se refiere a un conjunto de herramientas virtuales en el ambiente MATLABTM para operar el sistema iIDAQ. La Figura 1 muestra el diagrama de bloques del sistema iIDAQ y sus principales componentes.

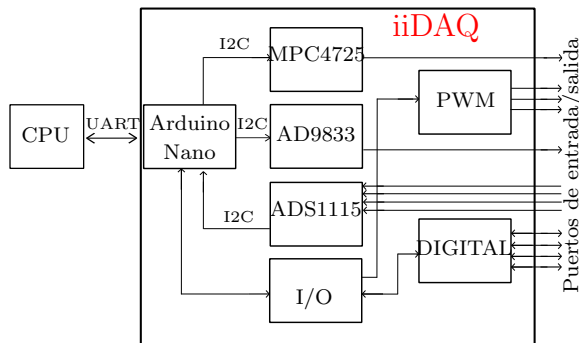


Figura 1. Diagrama a bloques del sistema iIDAQ.

En la Tabla 1, se resumen las características de los dispositivos que componen el *hardware* del sistema iIDAQ, donde es posible notar que, frente a otras alternativas de sistemas de adquisición de datos, éste es de fácil adquisición y costo reducido, logrando que estudiantes y docentes puedan reproducir el sistema para fines didácticos¹.

3. TOOLBOX

Como elemento indispensable se desarrolló un conjunto de herramientas *software* en el ambiente MATLABTM, denominado *toolbox*. Así, se cuenta con dos modalidades de funcionamiento para interactuar con el hardware del iIDAQ: i) modo gráfico y ii) modo de comandos.

3.1 Modo gráfico

El principal objetivo del modo gráfico de operación es tener un primer acercamiento con el sistema de adquisición de datos iIDAQ. Así, el usuario puede explorar el funcionamiento del sistema y hacer uso de los módulos del iIDAQ para aplicaciones de propósito general. Estas características, hacen del modo gráfico, un ambiente intuitivo y de gran apoyo para un aprendizaje rápido

¹ Los archivos para reproducir el sistema iIDAQ se encuentran disponibles en <https://github.com/rgunam/iIDAQ>

Tabla 1. Dispositivos *hardware* del sistema iIDAQ.

Dispositivo	Descripción	Costo (USD)
Arduino Nano 3.0	Microcontrolador base: Atmel ATmega328 de 8 bits @ 16 MHz.	\$10,99
ADS1115	Convertidor analógico-digital (ADC). Resolución: 16-bit con ganancia programable 2/3, 1, 2, 4, 8, 16.	\$7,49
MCP4725	Convertidor digital-analógico (DAC). Resolución: 12-bit (1.2 mV @ 5 V).	\$3,49
AD9833	Generador de señales sinusoidal, triangular y cuadrada. Frecuencia: 0.1 Hz a 10 MHz.	\$12,90
PWM	Tres terminales para generar señales PWM (pulse width modulation).	N/A
Digital I/O	Cuatro terminales digitales configurables como entrada o salida.	N/A

de conceptos relacionados con la adquisición de datos y generación de señales. La Figura 2 muestra la interfaz gráfica de usuario del sistema iIDAQ, y los bloques con los que cuenta para su operación, los cuales se describen a continuación.

1. *Bloque de conexión*. Configuración del puerto COM para la comunicación entre el sistema iIDAQ y la computadora, además de opciones para conectar y desconectar el sistema.
2. *Bloque módulo DAC*: Configuración de valor en bits, en un rango de 0 a 4095 para una salida de 0 a 5 volts. Se realizan cambios a partir de barra deslizante o *textbox* (caja de texto).
3. *Bloque módulo generador de señales*: Se permite seleccionar la forma de la señal de salida, además de su frecuencia en Hertz y grados de su fase. Disponible con dos canales.
4. *Bloque PWM*: Configuración del ciclo de trabajo (%) para los pines de salida PWM.
5. *Bloque pines digitales I/O (Input/Output)*: Configuración de pines como puertos de entrada o salida. Se obtienen o generan valores lógicos de 0 o 1, dependiendo el caso.
6. *Bloque módulo ADC*: Configuración del modo de lectura de datos (*Single-Ended* o diferencial), la ganancia operativa y los puertos que queremos utilizar para la adquisición de datos y su conversión a voltaje (ya sea en Volts o miliVolts), además de la opción de y registrador de datos (*datalogger*).

3.2 Funciones de bajo nivel

En esta modalidad, se desarrollaron funciones dedicadas para interactuar de forma separada con los módulos *hardware* del iIDAQ. El objetivo principal de este conjunto de funciones es usar al iIDAQ en aplicaciones más específicas



Figura 2. Interfaz gráfica de usuario para el funcionamiento básico del sistema iiDAQ.

y/o avanzadas. Particularmente, la intención es desarrollar aplicaciones con algoritmos para la adquisición de datos, control y sistemas de medición (Zhang, 2012). En la Tabla 2, se describen las funciones disponibles para poder utilizarlas en la programación de *scripts* en MATLAB™.

4. CASOS DE ESTUDIO

4.1 Medición de temperatura

El objetivo de este caso de estudio es construir de un termómetro con base en un sensor resistivo de temperatura RTD (*resistance temperature detector*), cuyo cambio de resistencia debido a un cambio de temperatura está dado por

$$R_x = R_0 \cdot (1 + \alpha \cdot (T - T_0)), \quad (1)$$

en donde R_0 es la resistencia nominal del Pt100, α es el coeficiente térmico del platino, T_0 es la temperatura ambiente y T la temperatura a medir. La Figura 3 muestra el circuito de acondicionamiento para la señal de un sensor modelo Pt100. Debido a las características de un RTD, la medición debe realizarse mediante un puente de Wheatstone para obtener datos con alta precisión. El objetivo es medir, directamente, la diferencia de potencial entre los nodos A y B, y, a partir de éste, obtener el valor de resistencia del Pt100. Realizando el análisis del circuito de la Figura 3 se tiene que la temperatura medida T está dada por

$$T = \frac{\frac{R_x}{R_0} - 1}{\alpha} + T_0. \quad (2)$$

Una vez definidas las operaciones que utilizaremos, y empleando el iiDAQ en modo GUI, configurado con entrada analógica de tipo diferencial, se mide y registra en una hoja de cálculo el votaje a la salida del puente de Wheatstone. Finalmente, se obtiene el valor de temperatura usando las ecuaciones (1) y (2). La Figura 3 muestra el resultado de la medición de temperatura para

una fuente de calor Q_0 . Como resultado, se se observa la evolución entre la temperatura ambiente $T_a \approx 25^\circ\text{C}$ y el valor máximo de $T_m \approx 105^\circ\text{C}$. Este primer caso estudio,

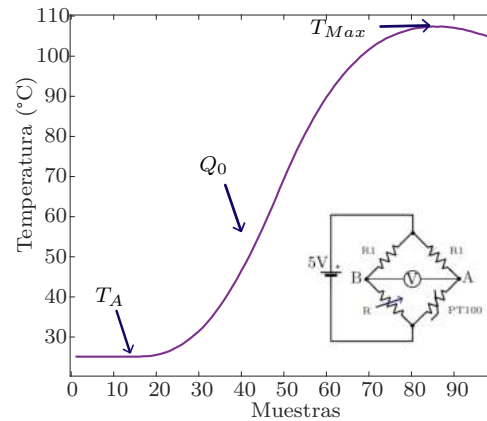


Figura 3. Medición de temperatura con sensor resistivo y el sistema iiDAQ. Se presenta también el Puente de Wheatstone conectado para medición de temperatura con RTD.

pretende ser un primer acercamiento a la instrumentación y medición de variables de proceso empleando el sistema iiDAQ. En dicho ejercicio, es posible poner en práctica tópicos relacionados con acondicionamiento de señales, análisis dinámico del sistema y adquisición de datos.

4.2 Controlador PI de un circuito RC

Para el segundo caso de estudio, considere un filtro, pasivo RC, paso bajas con $R = 10 \text{ k}\Omega$ y $C = 22 \mu\text{F}$. Dentro de los objetivos, por una parte será identificará el sistema dinámico, y, posteriormente, se diseñará un controlador proporcional-integral (PI) discreto para fines didácticos.

Tabla 2. Funciones del *toolbox* para desarrollo de aplicaciones personalizadas. La variable **s** será el objeto del puerto serie.

Función	Descripción/Parámetros [valor]
<code>connectiiDAQ()</code>	Conecta el sistema iiDAQ con el puerto serie disponible en la computadora.
<code>disconnectiiDAQ()</code>	Desconecta el sistema iiDAQ y cierra el puerto serie utilizado.
<code>configurationDAC(s, value)</code>	Genera un voltaje usando el MCP4725, en el rango de 0 a 5 volts. ▪ value : Entero [0-4095].
<code>signalGenerator(s, channel, waveForm, Frequency)</code>	Genera una señal a través del AD9833. ▪ channel : Entero para seleccionar canal [0, 1] ▪ waveForm : Cadena para selección de forma ["Sine", "Triangle", "Square"] ▪ Frequency : Valor de tipo entero para frecuencia [0-1E7]
<code>configurePWM(s, pin, dutyCycle)</code>	Genera señales PWM en la terminal indicada y configura el ciclo de trabajo. ▪ pin : Entero para seleccionar terminal [0, 1, 2] ▪ dutyCycle : Entero para definir % ciclo de trabajo [0-100]
<code>configureDigitalPin(s, digitalPin, value, type)</code>	Genera señales digital de entrada o de salida en el pin requerido. ▪ digitalPin : Entero para seleccionar pin [0, 1, 2, 3] ▪ value : Entero para definir estado lógico bajo (0) o alto (1) ▪ type : Entero para definir entrada (0) o salida (1)
<code>configureADC(s, mode, gain)</code>	Configura el ADS1115. ▪ mode : Entero para seleccionar modo <i>single-ended</i> (0) o diferencial (1) ▪ gain : Entero para definir ganancia [0-5]
<code>analogRead(s, pin)</code>	Adquiere el valor del ADS1115. ▪ pin : Entero para seleccionar pin [0, 1, 2, 3]

Identificación. Para cumplir el objetivo de identificación, deben considerarse las señales de entrada/salida para estimar la función de transferencia (FT) del sistema (Pintelon and Schoukens, 2012). La Figura 4(a) muestra los datos adquiridos para la señal escalón de entrada $u_e \approx 5$ V y el voltaje en el capacitor $y(t)$, medidas con el convertidor analógico digital en modo *single-ended*. Una entrada *single-ended* (*de un solo extremo*) mide el voltaje entre el canal de entrada de nivel alto y bajo de tierra común a todas las entradas. Este tipo de configuración permite que el sistema DAQ admita más canales, pero

es susceptible al ruido de la conexión a tierra común. Una vez que las señales son obtenidas, la FT se estima mediante el *System Identification toolbox* de MATLAB™, el cual, proporciona herramientas para construir modelos de sistemas dinámicos, a partir de datos medidos (MathWorks, 2022). Particularmente, se estima la FT en tiempo continuo de primer orden

$$\hat{H}(s) = \frac{K}{s + a}, \quad (3)$$

con $K = 4,46$ y $a = 4,406$. Así, es fácil notar que la constante de tiempo es $\tau = 1/a = 0,227$, el cual es muy cercano al valor nominal $\tau = RC$. Con dicho resultado, se corrobora que el sistema iiDAQ es útil para adquirir señales de un sistema dinámico real y, a su vez, éstas son compatibles con otras funciones del ambiente de MATLAB™.

Controlador. Considere el controlador proporcional-integral (PI)

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau') d\tau', \quad (4)$$

en donde $u(t)$ y $e(t)$ son las señales de control y error, respectivamente, K_p y K_i son las ganancias del controlador. Así, se establecen arbitrariamente los siguientes parámetros de diseño:

- Tiempo de asentamiento

$$t_s = \frac{4}{\sigma} < 1 \implies \sigma > 4. \quad (5)$$

- Tiempo pico

$$t_p = \frac{\pi}{\omega_d} < 0,350 \implies \omega_d = \frac{\pi}{0,350} \approx 8,98. \quad (6)$$

- Sobrepaso máximo

$$M_p = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} < 0,25 \implies \zeta \approx 0,215. \quad (7)$$

Donde σ es la parte real y ω_d la parte imaginaria del polo $P = 4 \pm 8,98j$. Con ello, se obtienen los valores de ganancias de la parte proporcional K_p e integral K_i

$$K_p = 2,5 \quad \text{y} \quad K_i = 20. \quad (8)$$

Luego, para controlar el voltaje del capacitor a través del iiDAQ es necesario trabajar en tiempo discreto. Para ello, se discretiza el controlador utilizando el método de Euler hacia atrás (*backward*), y se obtienen los parámetros

$$a_0 = 1 + (N \cdot t_s) \quad (9)$$

$$a_1 = -(2 + N \cdot t_s)$$

$$a_2 = 1$$

$$b_0 = K_p \cdot (1 + (N \cdot t_s)) + K_i \cdot t_s \cdot (1 + (N \cdot t_s))$$

$$b_1 = -(K_p \cdot (2 + (N \cdot t_s)) + (K_i \cdot t_s))$$

$$b_2 = K_p m,$$

en donde $t_s = 0,1$ segundos, es el tiempo de muestreo y N es el número de muestras. Con ello, la ley de control está dada por la siguiente ecuación recursiva

$$\begin{aligned} K_{e_0} \cdot y[n] + K_{e_1} \cdot y[n-1] + K_{e_2} \cdot y[n-2] = \\ K_{u_1} \cdot u[n-1] + K_{u_2} \cdot u[n-2], \end{aligned} \quad (10)$$

en donde $K_{u_1} = a_1/a_0$, $K_{u_2} = a_2/a_0$, $K_{e_0} = b_0/a_0$, $K_{e_1} = b_1/a_0$ y $K_{e_2} = b_2/a_0$. Para verificar el desempeño

del controlador, se genera una señal de referencia para el voltaje del capacitor. En la Figura 4(b) se muestra cómo el voltaje medido en el capacitor $y(t)$ sigue correctamente a la señal de referencia $r(t)$. Lo anterior puede ser verificado a través de la señal de error $e(t)$ mostrada en la Figura 4(c). Se puede observar que, el tiempo de respuesta es relativamente corto y el sobre paso es nulo, cumpliendo con los criterios de diseño del controlador. Finalmente, la Figura 4(d), muestra la señal de control $u(t)$ generada a través del DAC del sistema iiDAQ.

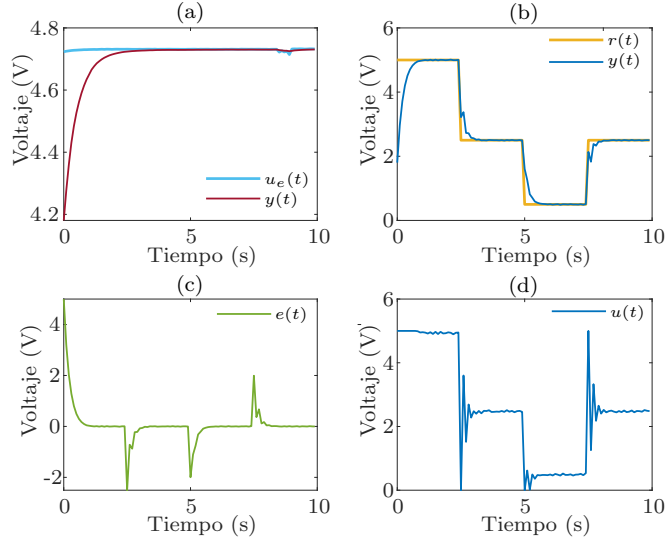


Figura 4. Identificación y control de un circuito RC. (a) Señales de entrada $u_e(t)$ y salida $y(t)$ medidas con el ADC. (b) Señal de referencia $r(t)$ y voltaje del capacitor controlado $y(t)$. (c) Señal de error $e(t)$ dinámico del sistema de control. (d) Señal de control $u(t)$ generada por el DAC.

4.3 Controlador adaptable de un motor DC

Para ilustrar las prestaciones del sistema iiDAQ, y su compatibilidad directa con MATLAB™, el tercer caso de estudio presenta el control de un motor DC con base en el módulo TY36A/EV (Figura 5(a)). En primera instancia, es necesario obtener la señal del transductor de velocidad o encoder, acoplado a la flecha del motor. Para ello, y con fines didácticos, se propone la construcción del tacómetro mostrado en la Figura 5 (b). Dicho circuito muestra una forma sencilla, pero versátil, de convertir la frecuencia de pulsos provenientes del encoder en una señal de voltaje. Así, este último, puede ser fácilmente digitalizado por el sistema iiDAQ para tener la señal de retroalimentación para implementar el algoritmo de control STR (*self tuning regulator*) (Åström and Wittenmark, 2008). El cual es un tipo de sistema de control adaptativo compuesto por dos lazos, un lazo interno (regulador de retroalimentación lineal ordinario y de proceso) y un lazo externo (estimador de parámetros recursivo y cálculo de diseño que ajusta sus parámetros).

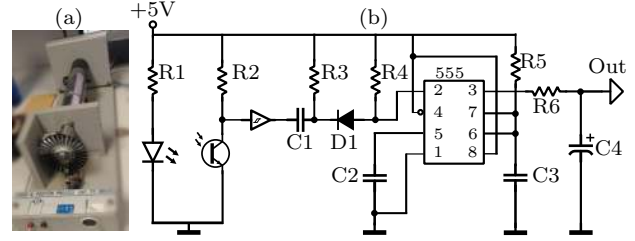


Figura 5. (a) Módulo TY36A/EV del motor DC. (b) Tacómetro construido a partir de un circuito convertidor frecuencia-voltaje con componentes discretos, para la medición de velocidad del motor DC.

Ahora, considere que el proceso está dado por el sistema

$$A(q)y(t) = B(q)u(t) + v(t), \quad (11)$$

en donde, $A \in \mathbb{R}^{na}$ y $B \in \mathbb{R}^{nb}$ son polinomios en el operador de cambio hacia adelante q , y es la velocidad medida, u es el voltaje aplicado en las terminales del motor y v es el ruido de medición. El primer objetivo del algoritmo STR es estimar los coeficientes de los polinomios en (11), mediante el método de mínimos cuadrados recursivos (RLS: *recursive least-squares*), dado por

$$e(t) = y(t) - \phi^\top(t)\theta(t-1) \quad (12)$$

$$\theta(t) = \theta(t-1) + k(t)[y(t) - e(t)] \quad (13)$$

$$k(t) = P(t-1)\phi(t)[\lambda I + \phi^\top(t)P(t-1)\phi(t)]^{-1} \quad (14)$$

$$P(t) = [I - k(t)\phi^\top(t)]P(t-1)/\lambda \quad (15)$$

en donde $e(t)$ es el error que se desea minimizar, $\theta = [a_1 \ a_2 \ \dots \ a_n \ b_1 \ b_2 \ \dots \ b_n]$ son los parámetros del modelo, $k(t)$ es la ganancia de Kalman y $P(t)$ es la matriz de covarianzas. Particularmente, para el caso del motor, se tiene que el sistema tiene cuatro parámetros a estimar, i.e. $\theta = [a_1 \ a_2 \ b_1 \ b_2]$. Con base en los parámetros estimados por RLS, se busca sintonizar los parámetros del controlador empleando la técnica MDDP (*minimum-degree pole placement*), cuya solución está dada por la ecuación de Diophantine

$$AR + BS = A_c \quad (16)$$

y el objetivo es estimar R , S y A_c de forma recursiva. Finalmente, la ley de control está dada en dos casos.

- Caso 1: con cancelación de cero

$$u(t) = bu(t-1) + t_0y_r(t) - s_0(t) - s_1y(t-1).$$

- Caso 2: sin cancelación de cero

$$u(t) = r_1u(t-1) + t_0y_r(t) - s_0(t) - s_1y(t-1).$$

Las Figuras 6 y 7 muestran el resultado del controlador adaptable para el caso 1 y caso 2, respectivamente. En el primer caso, es posible observar que la señal controlada $y(t)$ sigue relativamente bien a la señal de referencia $r(t)$ (Figura 6(a)). Sin embargo, el esfuerzo de control señalado por $u(t)$ (Figura 6(b)) es mayor al inicio del proceso de estimación, lo cual, es de esperarse debido a las condiciones iniciales del algoritmo RLS. Finalmente, las Figuras 6(c) y (d) muestran la estimación en línea de los parámetros del

sistema 11. Ahí, es posible observar que los parámetros convergen de forma aceptable. Para el segundo caso, i.e., sin cancelación de cero, la Figura 7(a) muestra que la velocidad controlada $y(t)$ sigue correctamente a la referencia $r(t)$, con la presencia de algunas oscilaciones debidas, presumiblemente, al ruido de la medición. Por su parte, la señal de control $u(t)$ mostrada en la Figura 7(b) es más suave respecto al primer caso. Por último, las Figuras 6(c) y (d) muestran la evolución de la estimación paramétrica del modelo 11. En dicho proceso, se observa que la incertidumbre disminuye respecto al primer caso; sin embargo, se espera sesgo en la estimación debido a la presencia del cero en la ley de control.

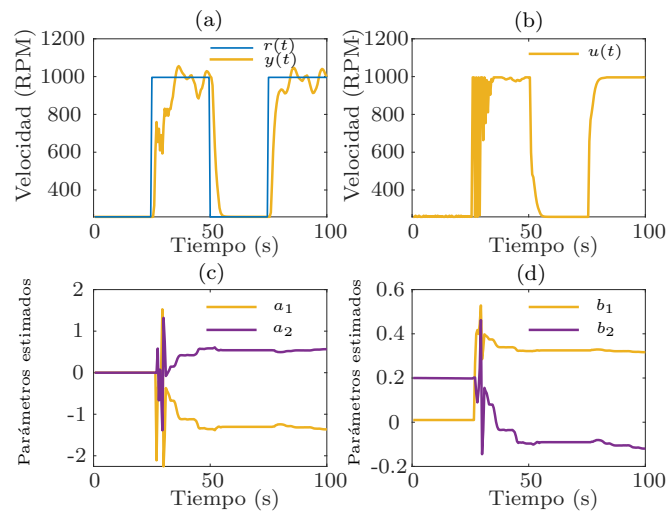


Figura 6. Controlador STR con cancelación de cero. (a) Velocidad controlada $y(t)$ y velocidad de referencia $r(t)$. (b) Señal de control $u(t)$. (c) y (d) Estimación de parámetros con RLS.

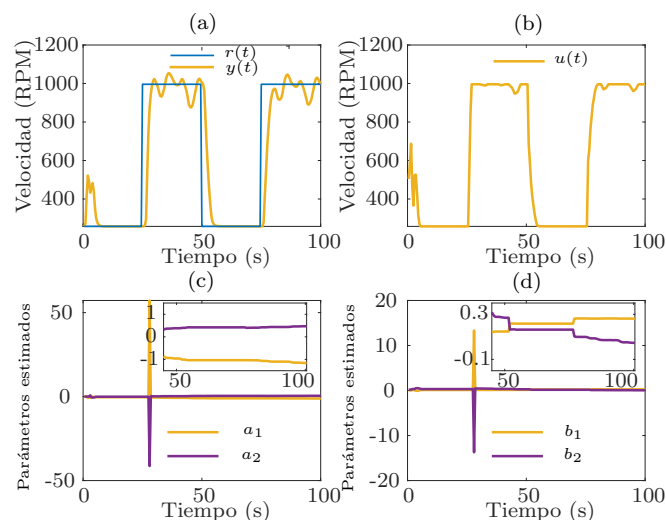


Figura 7. Controlador STR sin cancelación de cero. (a) Velocidad controlada $y(t)$ y velocidad de referencia $r(t)$. (b) Señal de control $u(t)$. (c) y (d) Estimación de parámetros con RLS.

5. CONCLUSIONES

En este trabajo se presentó al sistema iiDAQ como una herramienta didáctica para la enseñanza de tópicos de instrumentación y control. El iiDAQ está construido con *hardware* de fácil acceso, y es compatible con el ambiente MATLAB™ de forma amigable e intuitiva. La versatilidad del iiDAQ se mostró en tres casos de estudio con diferentes niveles de complejidad para fines didácticos. Los resultados obtenidos permiten vislumbrar nuevos paradigmas referentes a sistemas de adquisición de datos y su aplicación en entornos académicos. En próximas iteraciones, se pretende que el iiDAQ sea usado para explorar una mayor cantidad de aplicaciones, como la medición de respuesta en frecuencia, la identificación de modelos en espacio de estados y control no lineal, por mencionar algunos. Los avances mostrados en este trabajo son ilustrativos y, por ende, el desempeño mostrado puede ser mejorado de acuerdo con los futuros casos de estudio que se presenten. Asimismo, se pretende someter el sistema a una evaluación de estudiantes y docentes para obtener retroalimentación sobre el desempeño del iiDAQ y su aplicación en entornos académicos.

REFERENCIAS

- Åström, K. and Wittenmark, B. (2008). *Adaptive Control*. Dover Books on Electrical Engineering. Dover Publications.
- Aunkaew, S., Tantivivat, S., and Ibrahim, S.Z. (2021). Development of the educational tool for optimized algorithm model using matlab gui. In *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, 1–6.
- Blanco, J., Ferrero, F., Valledor, M., and Campo, J. (2014). A low-cost open-source data acquisition system. In *2014 IEEE 11th International Multi-Conference on Systems, Signals & Devices (SSD14)*, 1–6. doi: 10.1109/SSD.2014.6808795.
- Enériz, D., Medrano, N., Calvo, B., and Pérez-Bailón, J. (2020). A wireless instrumentation control system based on low-cost single board computers. In *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 1–5.
- González, A., Olazagoitia, J.L., and Vinolas, J. (2018). A low-cost data acquisition system for automobile dynamics applications. *Sensors*, 18(2).
- John Park, S.M. (2003). Definition of data acquisition and control. In *Data Acquisition for Instrumentation and Control Systems*, 3–11.
- MathWorks (2022). <https://la.mathworks.com/products/sysid.html>, 22 de Agosto de 2022.
- Morris, A. and Langari, R. (2012). Measurement and instrumentation: theory and application. In *Academic Press*.
- Pintelon, R. and Schoukens, J. (2012). An introduction to identification. In *System Identification*, 1–27.
- Zhang, T. (2012). Instrumentation, measurement, circuits and systems. In *Springer*.