

Desarrollo de un Sistema de Identificación de Procesos Industriales en Línea

Paul Aluisa, Rodolfo Gordillo, Luis Orozco

Departamento de Eléctrica y Electrónica, Universidad de las Fuerzas Armadas-ESPE,
Sangolquí, Ecuador

psaluisa@outlook.com

rxgordillo@espe.edu.ec

laorozco@espe.edu.ec

Resumen— Este documento es una recopilación del proceso de desarrollo de un sistema completo de identificación en línea para obtener un modelo matemático de procesos industriales, además se muestra el desarrollo de un sistema de emulación de plantas industriales, lo que va a permitir realizar las pruebas necesarias para el análisis de los resultados de los algoritmos de identificación paramétrica basados en mínimos cuadrados.

I. INTRODUCCIÓN

El uso de modelos de los sistemas físicos permite estudiar un problema sin tener la necesidad de experimentar, permitiendo a su vez detectar fallas y errores en los diseños. Así, se pueden evitar los riesgos de experimentos, analizar comportamientos y resultados de forma segura, reducir costos de desarrollo, realizar análisis de fallas, etc.

El proyecto consiste en desarrollar un sistema de identificación de procesos industriales en línea utilizando la plataforma open-source Arduino para la adquisición de datos y Matlab/Simulink para su respectivo procesamiento y el desarrollo de las rutinas de identificación paramétrica basadas en algoritmos de mínimos cuadrados.

Este artículo recopila los conceptos necesarios para comprender la emulación de procesos industriales a través de la discretización de funciones de transferencia en tiempo continuo, además del desarrollo de los algoritmos utilizados en la fase de identificación como son LMS (*Least Mean Squares*), RLS (*Recursive Least Squares*) y Lattice.

II. METODOLOGÍA

El proyecto realizado utiliza un emulador de plantas industriales para comprobar el funcionamiento de varios algoritmos de identificación.

El emulador se ha implementado dentro de un PLC (Controlador Lógico Programable), monitoreado por una interfaz gráfica desarrollada en Matlab®, la misma que permite seleccionar el tipo de función de transferencia que se va a emular diferenciados por el algoritmo de identificación que se va a utilizar, entre los cuales se encuentran LMS-RLS y Lattice de primer y segundo orden.

La interfaz se encarga de discretizar el modelo ingresado en forma paramétrica y enviar los parámetros al PLC donde se implementa el algoritmo de programación

que resuelve la ecuación a diferencias obtenida de la discretización del modelo ingresado. Para desarrollarla se necesitó un PLC Allen-Bradley CompactLogix 1768-L43 mostrado en la Fig. 1.

Dentro de este artículo se desarrolla el método de discretización utilizado, Retenedor de Orden Cero (ZOH) para una mejor comprensión del desarrollo del algoritmo de programación del PLC



Fig. 1 CompactLogix 1768-L43

La parte principal del proyecto es la de identificación, una vez generadas las señales de entrada (u) y salida (y) de la planta a través del emulador, se debe desarrollar el sistema de identificación.

Para sistema de identificación se utiliza Simulink® como plataforma de desarrollo, los diagramas desarrollados en Simulink® contienen un bloque de función de Matlab® en el que se desarrolla el algoritmo de identificación deseado, el bloque requiere como entradas las señales de entrada y salida de la planta emulada, dichas señales se adquieren a través de la tarjeta Arduino Mega 2560 mostrada en la Fig. 2 que poseen compatibilidad con Simulink para funcionar en modo “External” y de esta manera adquirir los datos de emulación generados por el PLC en línea.

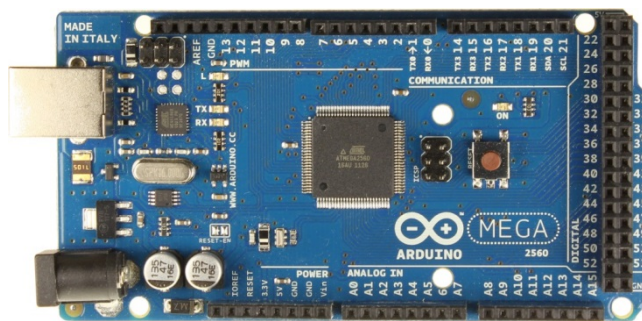


Fig. 2 Arduino Mega 2560

III. DESARROLLO

A. Discretización

Existen distintos procedimientos para obtener sistemas en tiempo discreto que se comporten aproximadamente igual a un sistema en tiempo continuo dado. Esta operación suele denominarse discretización.

La idea de discretizar un sistema es obtener $G_a[z]$ tal que se comporte aproximadamente como $G(s)$, en el sentido de que si se aplican a $G_a[z]$, muestras $u[k]$ de la entrada $u(t)$ aplicada a $G(s)$, resulte que su salida $y_a[k]$ coincida lo suficiente con las muestras de $y(t)$, tal y como se muestra en la Fig. 3.

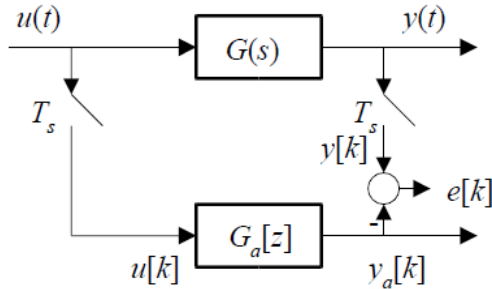


Fig. 3 Diagrama de Bloques Discretización

1) Retenedor de Orden Cero (ZOH)

La función de transferencia del retenedor de orden cero está dada mediante:

$$R(s) = \frac{1 - e^{-sT}}{s}$$

Para obtener el equivalente discreto de un sistema continuo se aplica la transformada Z al conjunto Retenedor-Planta de la siguiente manera:

$$G_a[z] = Z \left\{ \frac{1 - e^{-sT}}{s} G(s) \right\}$$

$$G_a[z] = (1 - e^{-sT})|_{e^{sT}=z} Z \left\{ \frac{G(s)}{s} \right\}$$

De donde se obtiene que el equivalente discreto de un sistema continuo utilizando un retenedor de orden cero viene dado por:

$$G_a[z] = (1 - z^{-1}) Z \left\{ \frac{G(s)}{s} \right\}$$

O

$$G_a[z] = \frac{z-1}{z} Z \left\{ \frac{1}{s} G(s) \right\}$$

La discretización se utilizará para realizar la emulación de plantas continuas, debido a que en el PLC se debe utilizar un modelo discreto para implementar el algoritmo de programación.

B. Identificación

La identificación de un sistema consiste en la obtención de un modelo matemático que caracteriza la dinámica de la planta y con ello se puede predecir su comportamiento. Así se podrán elaborar modelos matemáticos con la suficiente exactitud que permitan la aplicación de técnicas y

algoritmos de control conocidos, con un grado adecuado de sintonía de sus parámetros y con un rendimiento superior en las instalaciones donde estos algoritmos hayan sido implementados.

La identificación paramétrica está basada en los métodos de minimización del error de predicción, que se obtiene de la diferencia entre la salida real de la planta y el modelo paramétrico estimado derivados de la teoría de mínimos cuadrados. Esta teoría se basa en asumir que el proceso puede ser representado por un modelo de estructura fija, generalmente una ecuación lineal a diferencias, lo que quiere decir que dicho modelo tiene naturaleza discreta, como se muestra en la Fig. 4.

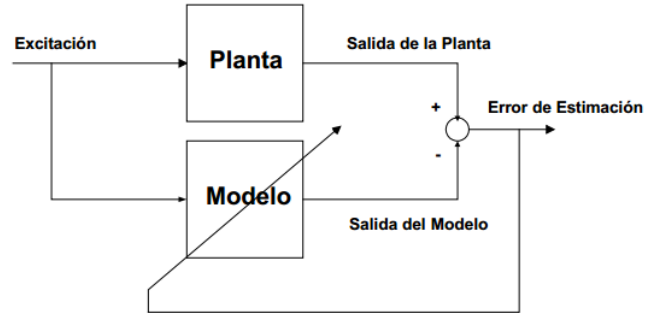


Fig. 4 Diagrama de Bloques Identificación

1) Identificación por Mínimos Cuadrados

Es una técnica de análisis numérico que se encuentra dentro de la optimización matemática, en la cual, dados un conjunto de pares ordenados: variable independiente, variable dependiente, y una familia de funciones, se intenta encontrar la función que mejor se aproxime a los datos, de acuerdo con el criterio de mínimo error cuadrático.

La razón principal para el uso del método de mínimos cuadrados es que a parte de su visión intuitiva, posee una serie de propiedades estadísticas simples y posibilita la implementación de una forma recursiva lo suficientemente simple. El método de mínimos cuadrados, define una estrategia, bastante utilizada para obtener un buen ajuste de los parámetros a encontrar. Este método permite la identificación de parámetros en los modelos lineales.

Del método de identificación por mínimos cuadrados se desprenden dos algoritmos principales, Mínimos Cuadrados no Recursivos (LMS) y Mínimos Cuadrados Recursivos (RLS), como alternativa se presenta el algoritmo LS-Lattice. Este algoritmo es básicamente una estructura en cascada para la identificación, basado en filtros de predicción forward y backward, como se muestra en la Fig. 5. El algoritmo se basa en el desarrollo de mínimos cuadrados para reducir el error cuadrático, de los filtros mencionados anteriormente, e ir actualizando los coeficientes de reflexión en función de tiempo y así calcular los parámetros del sistema identificado, el orden del sistema va a determinar el número de etapas que tendrá el identificador colocados en cascada.

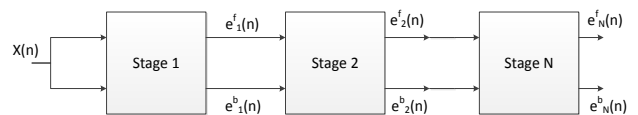


Fig. 5 Diagrama de Bloques LS-Lattice

C. Implementación del Sistema de Emulación

El emulador debe cumplir con las funciones de generar la señal de entrada hacia la planta (u) y la de resolver la ecuación a diferencias que representa el proceso industrial, generando la señal de salida (y); además de enviar dichas señales para su monitoreo mediante comunicación DDE (*Dynamic Data Exchange*) entre el PLC y Matlab, y también mandar las salidas a través del módulo analógico del PLC para integrar estos datos con el sistema de identificación en línea mediante su adquisición con la tarjeta Arduino Mega 2560.

La primera función del emulador, la cual corresponde a la generación de la señal de entrada, se desarrolla de tal manera que pueda ser seleccionada de entre tres opciones:

- Escalón Unitario
- Sinusoidal
- PRBS

Para la generación del escalón unitario simplemente se debe establecer el valor de $u(k) = 1$ antes del desarrollo de la ecuación a diferencias y sin que sufra algún cambio durante cada iteración.

Para generar una función seno se utiliza la instrucción "SIN" del PLC, la variable tiempo se establece con la ayuda del valor del acumulador de un timer (T2.ACC) ubicado en el programa principal el cual inicia su cuenta al momento en que se da esa orden desde la interfaz de monitoreo, siendo necesaria la transformación a radianes con la instrucción "RAD", se aumenta un valor el cual se puede modificar para cambiar la frecuencia de la senoide.

$$u(k) = \text{SIN}\left(\text{RAD}\left(\frac{\text{T2.ACC}}{24}\right)\right)$$

Debido a que la señal PRBS puede ser tomada como de naturaleza digital, se puede utilizar registros de desplazamiento, como el de la Fig. 6, para generar la variación del bit de salida y obtener una variación en el ancho del pulso, lo que hace que se obtenga un buen rango de frecuencias y la similitud a una señal de ruido blanco.

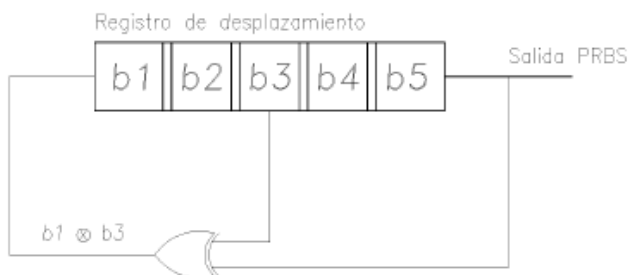


Fig. 6 Generación Señal PRBS

La segunda función principal del sistema emulador de procesos o plantas industriales es la de resolver la ecuación a diferencias del proceso a emular y obtener la señal de salida $y(k)$ a partir de la entrada $u(k)$, para dicho efecto se establecen cuatro tipos principales de procesos a emular, todos en tiempo discreto debido a que el PLC va a ejecutar las iteraciones en los intervalos de tiempo establecidos por el período de muestreo.

Debido a que el sistema de emulación tiene el fin específico de servir como modelo de pruebas para el sistema de identificación, debe acoplarse a los requerimientos finales, es decir, proveer modelos a identificarse con el algoritmo de mínimos cuadrados ya sea normal o recursivo, y con el algoritmo LS-Lattice, en ambos casos se da la posibilidad de identificar modelos de primer y segundo orden, por lo que el emulador tendrá la misma funcionalidad, dando así los cuatro modelos que se van a generar.

El primer modelo es de primer orden y se genera para ser identificado por el algoritmo LMS o RLS, de lo que se obtiene la función genérica en tiempo discreto.

$$G_a[z] = \frac{Y(z)}{U(z)} = \frac{k_1 z^{-1}}{1 + k_2 z^{-1}}$$

$$y(k) = k_1 u(k-1) - k_2 y(k-1)$$

El segundo modelo es de segundo orden y se genera para ser identificado por el algoritmo LMS o RLS, de lo que se obtiene la función genérica en tiempo discreto.

$$G_a[z] = \frac{Y(z)}{U(z)} = \frac{k_1 z^{-1} + k_2 z^{-2}}{1 + k_3 z^{-1} + k_4 z^{-2}}$$

$$y(k) = k_1 u(k-1) + k_2 u(k-2) - k_3 y(k-1) - k_4 y(k-2)$$

Para los dos últimos modelos de procesos a emular se determina los parámetros discretos directamente, y se usa con el algoritmo de identificación LS-Lattice para funciones de transferencia de primer y segundo orden.

$$G_a[z] = \frac{Y(z)}{U(z)} = \frac{1}{1 + k_1 z^{-1}}$$

$$y(k) = u(k) - k_1 y(k-1)$$

La última posibilidad de modelo a emular corresponde a la identificación con el algoritmo LS-Lattice de sistemas de segundo orden.

$$G_a[z] = \frac{Y(z)}{U(z)} = \frac{1}{1 + k_1 z^{-1} + k_2 z^{-2}}$$

$$y(k) = u(k) - k_1 y(k-1) - k_2 y(k-2)$$

El programa del PLC posee una rutina principal y 4 subrutinas las que contienen la generación de las señales de entrada y salida para los cuatro diferentes tipos de funciones de transferencia que serán sujeto de emulación.

D. Implementación del Sistema de Identificación

Para implementar el sistema de identificación se ha utilizado la herramienta Simulink® de Matlab®; como ya se mencionó anteriormente, con esta herramienta se puede

usar una programación gráfica en diagramas de bloques, la parte de adquisición de datos se la realiza a través de los bloques de entradas analógicas de Arduino que se encuentran en una librería especial de Simulink®; además, se tiene la facilidad de incorporar un bloque de Retenedor de Orden Cero (ZOH) para realizar la discretización, la parte principal del diagrama de bloques se encuentra en el bloque función de Matlab® que contiene el algoritmo de identificación, tiene como entradas las señales $u(k)$ y $y(k)$ obtenidas de los bloques de Arduino, y como salidas los parámetros discretos estimados.

La Fig. 7 representan la implementación de los algoritmos de identificación en Simulink®, en las mismas se observan los bloques mencionados anteriormente, adicionando bloques para poder visualizar la convergencia de los parámetros, la comparación entre la salida real de la planta y la salida estimada por el algoritmo, y también la facilidad de exportar estos datos a la línea de comandos de Matlab®.

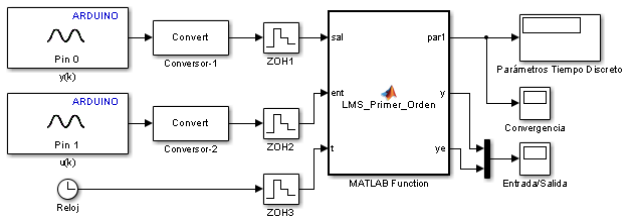


Fig. 7 Diagrama de bloques sistema de identificación de primer orden-LMS

El algoritmo LS-Lattice solo requiere la señal de salida de la planta emulada y (k) , por lo que el bloque función de Matlab solo requiere una entrada a diferencia de los algoritmos LMS y RLS que requieren de dos.

IV. RESULTADOS

A. Pruebas del Sistema de Emulación

Una vez implementados los algoritmos de generación de la señal de entrada a la planta y las ecuaciones a diferencias dentro del PLC, se pueden observar los resultados en la interfaz gráfica de Matlab que monitorea el sistema de emulación.

En las gráficas de la interfaz se muestran las señales de entrada (u) y salida (y) generadas, siguiendo el comportamiento de la planta que se ingresó a través de la misma interfaz, los parámetros ingresados representan una función de transferencia en tiempo continuo, una vez discretizada la planta se envían los parámetros al PLC para que resuelva la ecuación a diferencias.

Las Fig. 8 muestra un ejemplo de la respuesta generada por el sistema de emulación, para una planta de segundo orden con entrada sinusoidal.

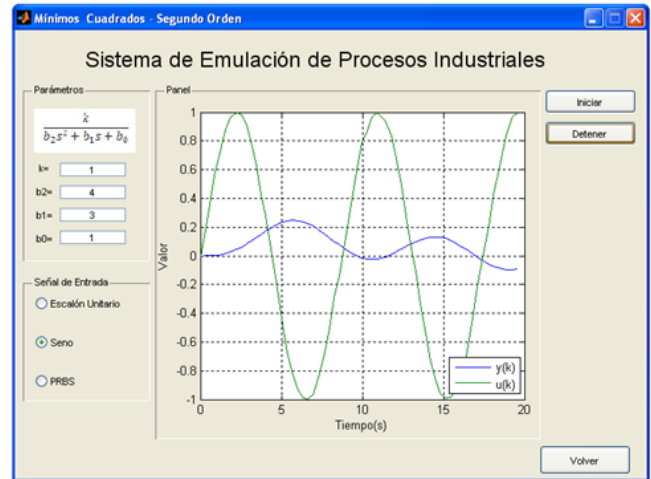


Fig. 8 Emulación de planta para mínimos cuadrados - segundo orden, entrada sinusoidal

Estas señales de entrada y salida son transmitidas a través del módulo analógico del PLC.

B. Pruebas del Sistema de Identificación

Los datos adquiridos por la tarjeta Arduino Mega 2560 a través de los bloques de la librería Arduino para Simulink, son escalados por la función de Matlab para que el algoritmo de identificación pueda interpretarlos de la forma adecuada y así estimar los parámetros de la planta, para los algoritmos LMS y RLS a parte de la señal de convergencia de los parámetros estimados, se genera las señales de valor real y estimado de la salida del sistema, y así poder realizar una comparación y verificar si difieren una de la otra.

La Fig. 9 muestra las señales de salida estimada y real generadas por el algoritmo de identificación LMS de primer orden, en la figura se puede observar que la señal estimada logra seguir de una manera aceptable a la salida real del sistema.

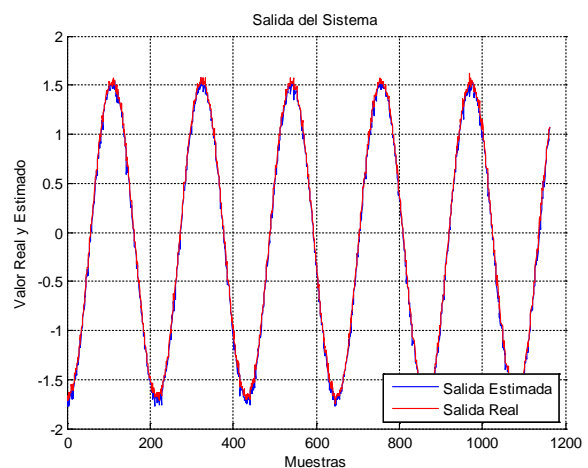


Fig. 9 Salida del sistema, real y estimada (sist. primer orden-LMS)

Las Fig. 10 y Fig. 11 muestran la convergencia de los parámetros $W1$ y $W2$, resultado de la estimación del algoritmo LMS para un sistema de primer orden, se puede observar en ambos una oscilación en el valor, pero hay que destacar que la tendencia a seguir el valor real está marcada

y la variación es mínima mirando la escala del gráfico.

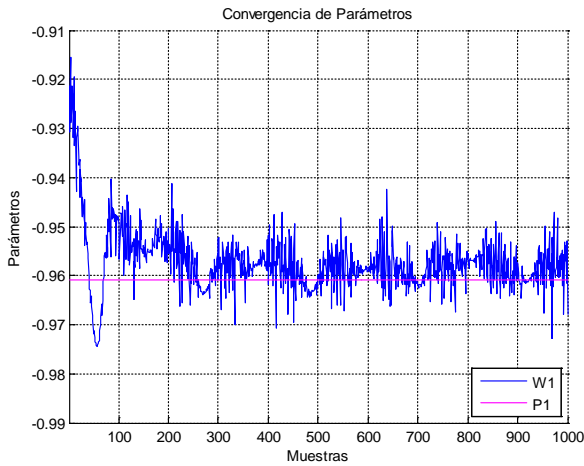


Fig. 10 Convergencia de Parámetros W1 (sist. primer orden-LMS)

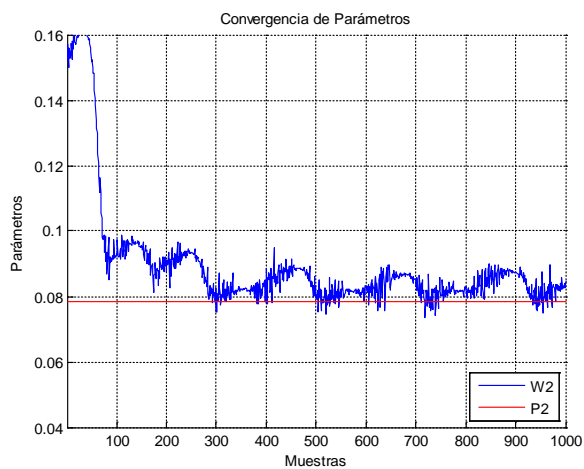


Fig. 11 Convergencia de Parámetros W2 (sist. primer orden-LMS)

El siguiente algoritmo que presentó buenos resultados en la identificación en tiempo real fue el RLS para sistemas de primer orden, en la Fig. 12 se puede observar una mejor respuesta de la señal de salida estimada en comparación al algoritmo LMS.

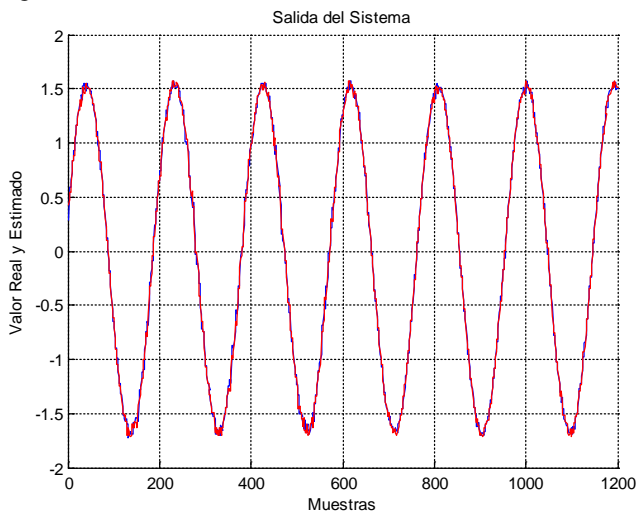


Fig. 12 Salida del sistema, real y estimado (sist. primer orden-RLS)

Las Fig. 13 y Fig. 14 muestran la convergencia de los parámetros W1 y W2, resultado de la estimación del

algoritmo RLS para un sistema de primer orden, se puede observar en ambos una oscilación mínima en el valor estimado manteniendo la tendencia hacia el valor real.

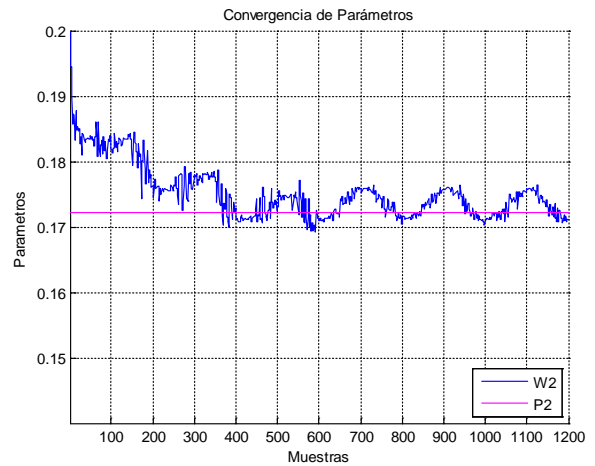


Fig. 13 Convergencia de Parámetros W1 (sist. primer orden-RLS)

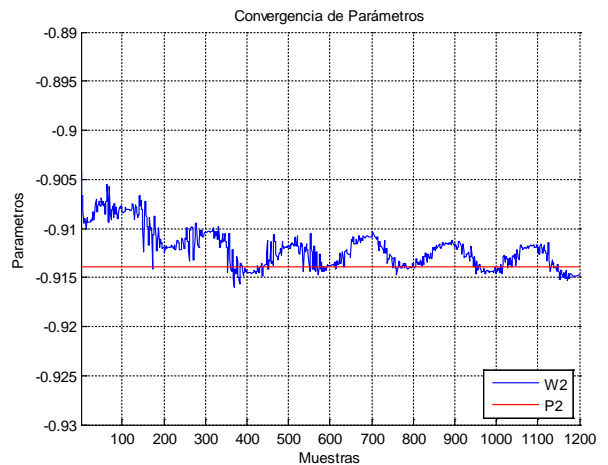


Fig. 14 Convergencia de Parámetros W2 (sist. primer orden-RLS)

El algoritmo de identificación LS-Lattice tiene buenos resultados para sistemas de primer orden como se observa en la Fig. 15, en las gráficas se puede observar la convergencia de los parámetros estimados que se aproximan hacia el valor deseado de los parámetros que son los generados por el sistema de emulación.

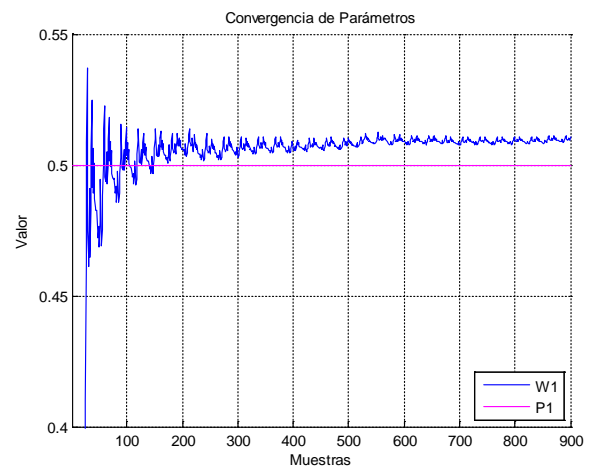


Fig. 15 Convergencia de Parámetros (sist. primer orden-Lattice)

V. CONCLUSIONES

El proyecto desarrollado permite realizar la identificación de varios tipos de procesos industriales generados por el sistema de emulación de procesos industriales destacando el algoritmo RLS como el más óptimo para la emulación de plantas en línea.

En comparación con el algoritmo LMS, el algoritmo RLS presenta más ventajas en la simulación por su naturaleza recursiva, mayor rapidez de convergencia y más exactitud en la estimación de los parámetros, pero se ve aumentada la complejidad computacional por las operaciones matriciales que se realiza a lo largo de la ejecución del algoritmo.

La desventaja de algoritmo RLS implementado se ve reflejada en la ejecución del sistema en tiempo real para sistemas de segundo orden donde el algoritmo se vuelve inestable, esto se debe al ruido al que es susceptible la tarjeta arduino, dicho ruido debería verse opacado con la variación del factor de olvido para no tomar en cuenta algunos valores pasados del par entrada/salida, la desventaja es la falta de implementación de un algoritmo de actualización del factor de olvido y así evitar el efecto de apagado antes de la convergencia de los parámetros.

Como una opción adicional el algoritmo LS-Lattice es otra opción para la estimación de parámetros que, en comparación con el algoritmo RLS, muestra una mejor estabilidad ya que no existe la desventaja de necesitar un factor de olvido variable, sino que, en este caso los coeficientes de reflexión, se van actualizando con cada iteración y para cada etapa del estimador.

REFERENCIAS

- [1] B. V. Tamani, «Obtención de Modelos de Procesos Mediante Métodos de Identificación Recursiva,» Universidad Nacional Mayor de San Marcos, Lima, 2007.
- [2] M. d. C. Meizoso López, A. Piñon Pazos y R. Ferreiro Garcia, «Implementación de Modelos de Sistemas Físicos Mediante Autómatas Programables y Circuitos Analógicos,» Universidad de La Coruña, La Coruña.
- [3] I. G. Martín, «Introducción a Arduino,» 02 11 2006. [En línea]. Available: <http://lagunak.gisa-elkartea.org/>. [Último acceso: 2012].
- [4] G. Miramontes, Procesamiento Digital de Señales Introducción con Teoría y Práctica, México: Cuerpo Académico de Procesamiento Digital de Señales, 2005.
- [5] V. M. S. Esquivel, Análisis de Sistemas y Señales con Cómputo Avanzado, Mexico: Unidad de Apoyo Editorial UNAM.
- [6] A. A. C. Vega, «Control Digital,» 2004.
- [7] L. Pagola, Control Digital, Madrid: Universidad Pontificia Comillas, 2002.
- [8] K. Ogata, Sistemas de Control en Tiempo Discreto Segunda Edición, Mexico: Prentice Hall, 1996.
- [9] A. Aguado Behar y M. Martínez Iranzo, Identificación y Control Adaptativo, Madrid: Pearson Educación, 2003.
- [10] D. Rodríguez Ramírez y C. Bordóns Alba, Apuntes de Ingeniería de Control, 2005.
- [11] R. Enríquez Herrador, «Guía de Usuario de Arduino,» Universidad de Córdoba, Córdoba, 2009.
- [12] Arduino, «Arduino,» [En línea]. Available: <http://arduino.cc/en/>. [Último acceso: 2012].
- [13] B. Evans, «Arduino Programming Notebook,» 2011. [En línea]. Available: <http://www.ardumania.es/>. [Último acceso: 2012].
- [14] S. Mendoza Jiménez y Y. Guillén Garcia, «Controladores Lógico Programables (PLC'S),» Expressa.
- [15] R. A. «Controladores CompactLogix L4x,» [En línea]. Available: <http://ab.rockwellautomation.com/es/>. [Último acceso: 2013].
- [16] J. A. Sirgo Blanco, «Autómatas Programables,» Universidad de Oviedo, Asturias, 2008.
- [17] K. Ogata, Ingeniería de Control Moderna 3ed., México: Prentice Hall, 1998.
- [18] A. Roca Cusidó, Control de Procesos, 2ed, México: Alfaomega, 2002.
- [19] A. S. Thomas, Adaptive Signal Processing (Theory and Applications), New York: Springer-Verlag, 1986.