

## **Algoritmo Recursivo de Mínimos Cuadrados: Identificación de Modelos de Plantas de Primer Orden para Procesos Industriales**

Gastón Rieiro Soñez<sup>1</sup>, Federico Gabriel D'Angiolo<sup>2</sup>, Fernando Asteasuain<sup>3</sup>, Roberto Osvaldo Mayer<sup>4</sup>

<sup>1</sup> Universidad Nacional de Avellaneda, Argentina  
gastonreiro@hotmail.com

<sup>2</sup> Universidad Nacional de Avellaneda, Argentina  
fdangiolo@undav.edu.ar

<sup>3</sup> Universidad Nacional de Avellaneda, Argentina  
fasteasuain@undav.edu.ar

<sup>4</sup> Universidad Nacional de Avellaneda, Argentina  
rmayer@undav.edu.ar

**Resumen.** En el presente trabajo se analiza el algoritmo Recursivo de Mínimos Cuadrados para identificar los parámetros de una Planta de primer orden. Para esto, el algoritmo se implementa en una placa EDU-CIAA dado que contiene un microcontrolador de 32 bits el cual permite tener un mayor poder de procesamiento, en comparación con una arquitectura de 8 bits.

### **1 Introducción**

Muchos sistemas industriales resultan tener un modelo matemático que describe su dinámica o evolución en función del tiempo. Este modelo permite un análisis mediante sistemas computacionales para lograr un estudio profundo de su comportamiento y de esta manera poder conocer o predecir sus estados transitorios y estables. Por ejemplo, en el caso de un horno eléctrico [1-2], su modelado se puede asemejar a uno de primer orden con lo cual, si se desea estudiar cómo evoluciona la temperatura en función del tiempo, se lo podría realizar con dicho modelo. Para conocer los parámetros o constantes que modelan a un sistema de primer orden, se puede emplear el algoritmo de Mínimos Cuadrados el cual tiene gran efectividad en sistemas embebidos [3].

Si bien el método tratado en [3], es de gran eficacia, es posible detallar algunos inconvenientes a la hora de computar las ecuaciones correspondientes al mismo, cuestiones que se plantean en el presente trabajo y que al mismo tiempo logran ser superadas mediante el Algoritmo de Mínimos Cuadrados Recursivo [4], un método que parte de los mismos conceptos teóricos que su predecesor, pero que difiere en su

desarrollo e implementación. En [5-6] se detallan diferentes situaciones en las que se logra implementar exitosamente.

El objetivo del Algoritmo Recursivo de Mínimos Cuadrados [4] resulta ser el de obtener los parámetros de una planta<sup>1</sup>, en este caso, de primer orden. Este presenta mejoras en cuanto al tiempo de procesamiento, para lo cual, teniendo en cuenta que su implementación será en una EDU-CIAA<sup>2</sup> (arquitectura de 32 bits), resulta importante. Además el estudio de este algoritmo tiene en cuenta cómo afecta al gasto computacional ya que en un sistema embebido, a diferencia de una computadora convencional, los recursos tales como memoria y procesamiento, no son excesivos. Es por esto entonces que resulta importante su análisis.

Actualmente el desarrollo de sistemas de control basados en sistemas embebidos, resulta ser un área de gran interés [7-8-9-10], ya que se necesita procesamiento en tiempo real sobre hardware, que tenga la mayor adaptación posible en la industria.

Este trabajo se distribuye de la siguiente forma: La sección número dos comenta cómo se puede aplicar este algoritmo a una planta de primer orden, describiendo además su funcionamiento. Esto da lugar a la sección tres, donde se comenta la implementación en la placa EDU-CIAA, la cual puede ejecutar al algoritmo en tiempo real. Por último, en la sección cuatro se muestran los resultados para luego poder comentar las conclusiones obtenidas.

## 2 Aplicación del algoritmo en una planta de primer orden

Para la descripción del algoritmo, se utiliza una planta de primer orden, basada en un circuito R-C pues presenta sencillez en su forma de implementación. Sin embargo, el estudio permite tener un mayor alcance a otro tipo de plantas de primer orden como por ejemplo, un horno donde se pueda medir la temperatura.

En la Fig.1 se muestra un diagrama en bloques de la Planta que se desea analizar. La misma se encuentra modelizada en función de la Transformada de Laplace.

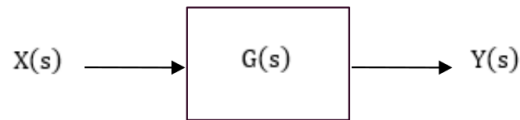
---

1

Sistema físico del cual se desea emular su comportamiento.

2

La EDU-CIAA es una versión de bajo costo de la CIAA-NXP (Computadora Industrial Abierta Argentina). EDU-CIAA es utilizada para promover la enseñanza de sistemas embebidos con el fin de impulsar el desarrollo tecnológico nacional y formar para el trabajo a los estudiantes de distintos niveles educativos.



**Fig. 1.** Planta de primer orden representada como  $G(s)$ , a través de la Transformada de Laplace.

En un sistema de primer orden como este, la función de transferencia resulta ser:

$$G(s) = \frac{k}{1 + s\tau} = \frac{Y(s)}{X(s)} \quad (1)$$

Siendo:

$k$  = Ganancia del sistema.

$\tau$  = Constante de tiempo del sistema de primer orden.

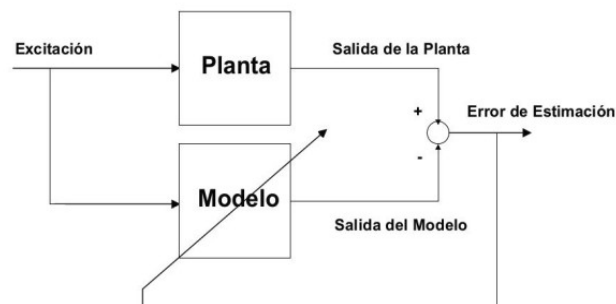
$G(s)$  = Transferencia en términos de la Transformada de Laplace.

$Y(s)$  = Salida del sistema en términos de la Transformada de Laplace. En función del tiempo sería  $y(t)$ .

$X(s)$  = Entrada en términos de la Transformada de Laplace. En función del tiempo sería  $x(t)$ .

En la transferencia  $G(s)$ , se puede observar que los parámetros característicos resultan ser la ganancia  $k$  y la constante de tiempo  $\tau$ .

Para obtener las constantes comentadas, se conecta en paralelo a la planta descrita en la Fig.1, un sistema computacional. Este sistema captura los valores de entrada y de salida ( $x(t)$  e  $y(t)$ ), y en base a estos, se calculan las constantes de la planta. Es importante aclarar que el sistema computacional, que en este caso se trata de EDU-CIAA, utiliza el conversor analógico digital (A/D), para muestrear los valores mediante un período de muestreo,  $T$ . Esto se observa en la Fig.2



**Fig. 2.** Planta de primer orden y sistema computacional. [4]

Una vez obtenidos los valores de entradas y salidas de la planta, el Algoritmo Recursivo de Mínimos Cuadrados calcula las constantes que caracterizan a la Planta. Cabe recordar que en una planta de este estilo, el parámetro a estimar es  $\tau$ , el cual resulta ser el producto entre la resistencia ( $R$ ) y el capacitor ( $C$ ), del circuito. Por su parte, el valor de  $k$  queda definido en el valor  $1$ , para un circuito R-C.

En la Fig. 2 se observa que, cuando el Modelo (EDU-CIAA), termina de calcular las constantes, la respuesta del mismo debe ser similar al de la planta. Para comparar estas salidas, se utiliza el error de estimación, el cual contempla las salidas de la planta y del modelo. Cuando ambas tienen gran similitud, el error tiende a cero.

Para este trabajo, los valores de  $R$  y  $C$  son fijos desde un principio pero luego, en un momento aleatorio, se modifican sus valores con el objetivo de analizar si el Algoritmo Recursivo de Mínimos Cuadrados puede detectar este cambio con éxito y continuar la estimación de parámetros sin problema alguno. A diferencia del trabajo [3], en donde el Algoritmo de Mínimos Cuadrados es estático, esta prueba carece de valor ya que la estimación se realiza en base a valores fijos de  $R$  y  $C$ .

Habiendo descripto el objetivo que se persigue con este algoritmo en la aplicación de plantas de primer orden, a continuación se procede con el análisis del algoritmo, de manera que se pueda comprender cómo funciona.

## 2.1 Descripción del Algoritmo Recursivo de Mínimos Cuadrados

El algoritmo a estudiar se desprende de un método más general, el de Mínimos Cuadrados el cual, cuando los parámetros de la planta cambian, su estimación pierde eficacia. El Algoritmo Recursivo de Mínimos Cuadrados solventa este inconveniente. Para comprender mejor esta idea, a continuación se expresa la ecuación con la cual el algoritmo de Mínimos Cuadrados puede estimar los parámetros de una planta:

$$\theta_k = [\phi_k^T * \phi_k]^{-1} * \phi_k^T * Y_k \quad (2)$$

Siendo:

$\theta_k$ : Vector que contiene las constantes del sistema.

k: Cantidad de instantes considerados para la obtención del  $\theta$ .

$\phi_k$ : Matriz que contiene todos los valores de salidas y entradas del sistema evaluados para cada instante.

$Y_k$ : Vector que contiene los valores de salida evaluados en cada instante.

A continuación, se enumeran las principales desventajas de este algoritmo:

- 1) A mayor **k**, mayor es el tiempo de procesamiento que requiere el algoritmo para obtener  $\theta_k$ .
- 2) La obtención de  $\theta_k$  requiere el cálculo de la inversa de la matriz  $[\phi_k^T \cdot \phi_k]$ , un proceso considerablemente costoso computacionalmente, sobre todo para aplicarlo en microcontroladores. Esto es suponiendo que la matriz sea inversible, es decir, que sea no singular.
- 3) El método mantiene un registro de todas las instancias evaluadas hasta el momento, lo que ocasionará que el modelo pierda fidelidad ante cambios imprevistos de las propiedades de la planta.

El Algoritmo Recursivo de Mínimos Cuadrados resuelve cada una de las problemáticas descritas anteriormente. Permite, al igual que su predecesor, automatizar la búsqueda de la relación causa-efecto entre la excitación y respuesta de un proceso dado, pero de forma recursiva [5-6]. Mediante los valores ingresados en la entrada del sistema, y su salida (Fig.1), se obtiene un vector  $\theta$ , el cual contiene las constantes de la planta de primer orden. Este vector es actualizado en cada iteración, descartando las constantes previas y centrándose únicamente en las actuales.

El algoritmo propuesto para la identificación de este parámetro es el siguiente:

$$P_k = x_k * x_k^T \quad (3)$$

$$K_k = \frac{P_k * x_k}{1 + x_k^T * P_k * x_k} \quad (4)$$

$$\theta'_k = \theta'_{k-1} + K_k * (y_k - x_k^T * \theta'_{k-1}) \quad (5)$$

Siendo:

k: El instante evaluado.

$x_k$ : par entrada-salida.

$P_k$ : Matriz de covarianza.

$K_k$ : Matriz que define el impacto que tiene el error (diferencia entre las salidas) sobre cada coeficiente.

$y_k$ : Salida de la planta evaluada en el instante k.

$\theta_k$ : Vector que contiene las constantes que multiplican a los valores de salidas y entradas evaluados en el instante k.

Como se puede apreciar, en este método se trabaja directamente con  $x_k$ , en lugar de la matriz  $\Phi_k$ , indispensable en el método tradicional. Esto no solo agiliza el algoritmo, sino que además le permite adaptarse ante cambios repentinos de las propiedades intrínsecas de la Planta ( $\tau$ ).

Tampoco se requiere de la inversa de una matriz, lo que agiliza aún más el algoritmo.

De esta manera, se va refinando el valor de  $\theta'$  en cada iteración, buscando que la diferencia del valor de salida entre la planta y el modelo ( $y_k - x_k^T * \theta'_{k-1}$ ) sea cero, provocando que  $\theta'_k = \theta'_{k-1}$ , definiendo así las constantes del sistema.

### 3 Implementación en la placa de desarrollo.

Dado que la planta se encuentra modelizada mediante la Transformada de Laplace (ec.(1)), resulta necesario llevar este modelo a uno discreto para que el microcontrolador pueda accionar. Para lograr esto, se utiliza la transformación de Tustin, la cual viene dada por:

$$s = \frac{2 * z - 1}{T * (z + 1)} \quad (6)$$

Utilizando la ec.(6) en la ec.(1), es posible deducir la expresión de la planta en el dominio de frecuencia discreta (en términos de “z”), es decir:

$$G(z) = \frac{Y(z)}{X(z)} = \frac{z+1}{z+1 + \frac{\tau * 2}{T} * z - 1} \quad (7)$$

Una vez descripta  $G(z)$ , será posible obtener la ecuación en diferencias expresada en términos de “ $k$ ” (tiempo discreto) capaz de ser programada dentro del microcontrolador. Esta ecuación permite obtener un valor de salida a partir de un valor de entrada, simulando así, a la planta de primer orden, de forma digital o discreta. En base a esto, la ecuación en diferencias queda:

$$y[k] = \frac{-T + \tau * 2}{T + \tau * 2} * y[k - 1] + \frac{T}{T + \tau * 2} * (x[k] + x[k - 1]) \quad (8)$$

Siendo:

$y[k]$ : Salida de la planta de primer orden discretizada.

$x[k]$ : Entradas de la planta de primer orden discretizada

$T$ : Período de muestreo. Este resulta necesario en el microcontrolador, para tomar muestras de las entradas y de las salidas.

La cual es posible reducir a la siguiente expresión:

$$y[k] = A * y[k - 1] + B * U_k \quad (9)$$

Donde:

$A$ : coeficiente correspondiente al vector de salida  $y[k-1]$ .

$B$ : coeficiente correspondiente al vector de entrada  $x[k]$ .

$U_k$ : conjunto de entradas evaluadas  $x[k]$ .

A partir de la simplificación de la ec. (9) hacia la ec. (10) es posible apreciar que  $A$  y  $B$  guardan una estricta relación con la constante de tiempo  $\tau$ . Tras definir un

período de muestreo ( $T$ ), si el algoritmo se implementa adecuadamente, deberá ser posible obtener el valor de  $\tau$  a partir de estas. Sin embargo, a la hora de implementar las ecuaciones correspondientes al algoritmo encargado de deducir dichas constantes ( $A$  y  $B$ ), resulta conveniente expresar a  $y[k]$  en su forma vectorial, ya que todo sistema computacional se utiliza el procesamiento de matrices. Por ende, conviene expresar dicha ecuación de la siguiente manera:

$$y_k = x_k^T * \theta_k \quad (10)$$

Siendo:

$$x_k^T = [y_{k-1}, U_k] \quad (11)$$

$$\theta = [A, B]^T \quad (12)$$

No obstante, dado que la planta ya posee sus constantes definidas ( $\tau$ ) lo mismo le corresponde a  $\theta$ , lo que le permite definir una salida específica para cualquier entrada elegida.

Lo que aquí concierne es construir un modelo equivalente que posea como dato aquellos pares de entrada-salida, y como incógnita la constante  $\theta$  (convenientemente denominada  $\theta'$ ), la cual contendrá intrínsecamente a la constante  $\tau$ , que resulta ser el valor a determinar.

De esta forma, queda definido nuestro modelo:

$$y'_k = x_k^T * \theta'_k \quad (13)$$

El código desarrollado para la EDU-CIAA, implementado en **C**, contiene cada una de las ecuaciones descritas al final de la sección 2. La placa, además, se programó para tomar pares de muestras entrada-salida, y así, calcular cada una de estas ecuaciones dentro de un bucle de forma independiente, obteniendo para cada instante un nuevo valor de  $\theta'$ .

#### 4 Resultados obtenidos.

En base a la descripción hecha hasta aquí, a continuación se describen dos ensayos posibles, que permiten la obtención de los resultados: el primero hecho en Scilab, software que permite el análisis matricial para corroborar lo planteado y el



segundo, la implementación en EDU-CIAA, para lograr la implementación del algoritmo.

#### 4.1 Implementación en Scilab

Se realizaron pruebas para diferentes combinaciones de resistencias y capacitores, con el fin de constatar el correcto funcionamiento del programa realizado. Este ensayo consiste en cambiar el valor de la variable  $R$  cada 8 segundos, comenzando por  $R = 147\Omega$  hasta llegar a  $R = 32.970\Omega$ . El valor de  $C$  es  $96 \mu\text{F}$ .

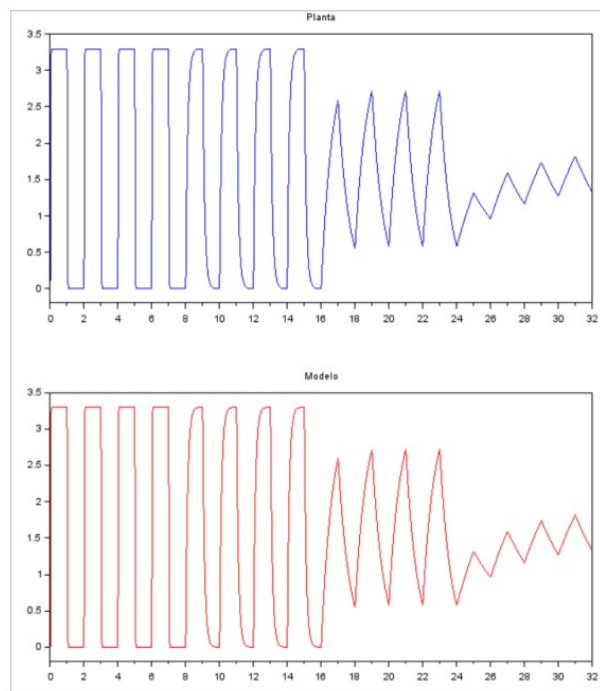


Fig. 3. Comparación Planta-Modelo implementado y graficado en Scilab.

Para construir la Figura 3 se simuló por separado tanto la respuesta de la planta como del modelo. La entrada de ambos sistemas corresponde a una señal cuadrada de  $3.3V_{pp}$ , con un valor medio de  $1.65V$  y un periodo de 2 segundos. La variable  $R$  alterna sus valores en los instantes 8, 16, y 24 segundos, en los que toma el valor de 1.198, 6.750 y  $32.970 (\Omega)$  respectivamente. La misma, junto a la variable  $C = 96 (\mu\text{F})$ , constituyen diferentes valores de  $\tau$ . Fue necesario graficar la Planta y el

Modelo de forma independiente, dado que al hacerlo sobre la misma ventana los gráficos se solapan, obstaculizando la posibilidad de diferenciar entre ambos.

#### 4.2 Implementación en EDU-CIAA.

Una vez comprobado el funcionamiento del algoritmo en Scilab, se procede a implementar el mismo sobre la placa EDU-CIAA, esta vez utilizando un circuito R-C real, a partir del cual el sistema embebido intentará obtener sus constantes ( $\tau$ ). El cambio de las resistencias se ejecuta manualmente tras apreciar una semejanza considerable entre ambas salidas. En cada una de las imágenes que se muestran a continuación, la respuesta de la planta se encuentra en color amarillo. Por su parte, la salida del modelo se ve reflejada en color violeta. Cabe aclarar que estas imágenes fueron tomadas a finales del año 2019 y corresponden a una implementación que, al momento de presentación de este trabajo, no se pudo actualizar. Las modificaciones de estas imágenes no fueron posibles de obtener debido al cierre de la Universidad en la que se realiza la experimentación, a causa del aislamiento social preventivo y obligatorio presente en todo el territorio argentino debido al virus COVID-19.



Fig. 4. Comparación Planta-Modelo con  $R= 32.970\Omega$ ;  $C= 96\mu\text{F}$ ;  $\tau= 3.16512 \text{ s}$ .



Fig. 5. Comparación Planta-Modelo con  $R= 6.750\Omega$ ;  $C= 96\mu F$ ;  $\tau= 0.648$  s



Fig. 6. Comparación Planta-Modelo con  $R= 1.198\Omega$ ;  $C= 96\mu F$ ;  $\tau= 0.115$  s.

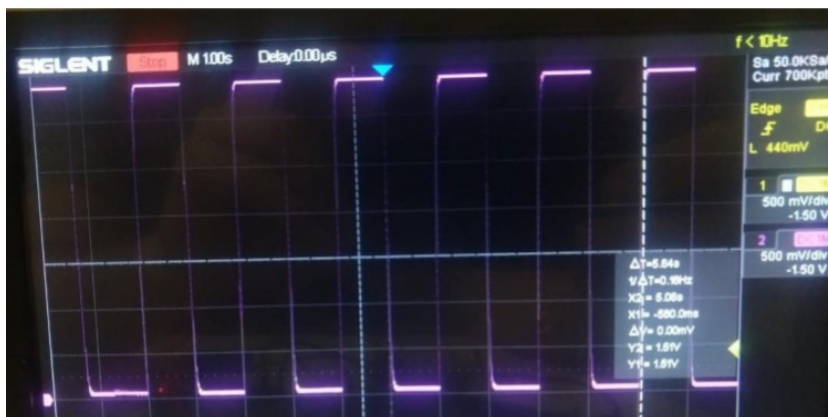


Fig. 7. Comparación Planta-Modelo con  $R= 147\Omega$ ;  $C= 96\mu F$ ;  $\tau= 0.01411$  s.

En las Fig. 5, 6 y 7 se observa cómo el algoritmo permite una rápida adaptación frente a cambios repentinos en la planta. Este procesamiento en tiempo real, es el que facilita su implementación en la industria, ya que en lugares con maquinarias o sistemas que realicen perturbaciones importantes sobre sistemas embebidos, con este algoritmo se puede seguir calculando las constantes de la planta sin problemas.

En la Tabla 1 se muestran las distintas combinaciones con los valores de  $\tau$  (esperados),  $\tau'$  (obtenidos) y %E (porcentaje de error medido).  $\tau'$  es, en realidad, un promedio de los últimos 20 valores de  $\tau'$  obtenidos.

**Tabla 1.** Valores  $\tau$  y  $\tau'$  obtenidos a partir del cálculo y estimación respectivamente. %E representa el error medido en cada caso particular..

C/R	32.970 $\Omega$	6.750 $\Omega$
96 $\mu$ F	$\tau = 3.165s$ $\tau' = 2.981s$ %E= 6.17%	$\tau = 0.648s$ $\tau' = 0.630s$ %E= 2.8%
C/R	1.198 $\Omega$	147 $\Omega$
96 $\mu$ F	$\tau = 0.115s$ $\tau' = 0.112s$ %E= 2.67%	$\tau = 0.0141s$ $\tau' = 0.0135s$ %E= 4.4%

De la **Tabla 1** se puede observar que el algoritmo estima a las constantes de forma aproximada dado el parámetro %E, el cual expresa el porcentaje de error cometido entre  $\tau$  y  $\tau'$ .

## 5 Conclusiones.

Dadas las limitaciones del algoritmo de Mínimos Cuadrados y su implementación en EDU-CIAA [3], este trabajo permitió presentar la puesta en marcha del algoritmo Recursivo de Mínimos Cuadrados, para comprobar las ventajas de este frente al anterior descripto.

Con esto, no solo se demostró la virtud del algoritmo al obtener las constantes de la planta en una implementación real, sino que fue posible percibir su adaptabilidad a través de Scilab, demostrando así su mejora ante el método tradicional. Además, cabe recalcar que la optimización en el tiempo de respuesta resulta fundamental a la hora de implementar este tipo de algoritmos en un sistema embebido, el cual cuenta con recursos limitados si se lo compara con un ordenador convencional.

Se puede decir que este algoritmo resulta conveniente y ventajoso a la hora de modelar el comportamiento de este tipo de sistemas, lo cual resulta de gran importancia para la industria, sobre todo en sistemas que se puedan asemejar a uno de primer orden, como puede ser el caso de una planta térmica. Como trabajo a futuro, se podría evaluar el comportamiento de este algoritmo frente a plantas de segundo orden, para notar su desempeño.

Este trabajo fue el resultado de los temas aplicados en el Laboratorio Abierto de Sistemas Embebidos, tomando a su vez, conocimientos de la materia “Metodología de Identificación y Control”, de la Carrera Ingeniería en Informática de la Universidad Nacional de Avellaneda.

## Referencias

1. Ogata,K: Ingeniería de Control Moderna. 5ta. Ed. PEARSON EDUCACIÓN, S.A., Madrid, 2010. ISBN: 978-84-8322-660-5.
2. Kuo,B., Golnaraghi,F.: Automatic Control Systems. 9th. Ed. Wiley.ISBN-13 978-0470-04896-2.
3. Ane,A., Contreras,D., Mayer, R., D’Angiolo,F.: Identificación de Modelo de Plantas de Primer Orden mediante Mínimos Cuadrados para Procesos Industriales.. JAIIO 2018. ISSN: 2451-7615.
4. Zanini,A.: Teoría de Control para Procesos Industriales. AADECA (2006). ISBN: 978-950-99994-5.
5. Rozaimi,G. Yahaya, S., Mohd,R., Zulfatman.: On-line identification of an electro-hydraulic system using recursive least square. 2009 IEEE Student Conference on Research and Development (SCORED). ISBN: 978-1-4244-5186-9
6. Wenke,X., Fuxiang,L.: Recursive algorithm of Generalized Least Squares Estimator. 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE). ISBN: 978-1-4244-5569-0.
7. Krucheski,J., Olsson,F., Fernández,G., Dondo,D., Araguás, G.: Prototipo de Sistema de Control y Supervisión de Nivel de Líquido para Uso Didáctico. Congreso Argentino de Sistemas Embebidos. 2018. ISBN 978-987-46297-5-3.
8. Gaydou,D., Dipré,M., Olmedo,M., Dondo,D., Araguás, G.: Implementación de un driver de control para motores BLDC sin sensores. Congreso Argentino de Sistemas Embebidos. 2018. ISBN: 978-987-46297-5-3.
9. M. Lee., K. Kong.: Design of sensorless bldc motor driver using variable voltage and back emf differential line. Journal of Institute of Control, Robotics and Systems, vol. 21, no.10.
10. T.-S. Kim., B.-G. Park., D.-M. Lee., J.-S. Ryu., and D.-S. Hyun.: A new approach to sens less control method for brushless dc motors. International Journal of Control, Automation, and Systems, vol. 6, no. 4, pp. 477–487, Aug 2008