

# Sintonización de un controlador PID implementado en un PLC haciendo uso de inteligencia de enjambres

## Auto-tuning of a PID controller implemented in a PLC using swarm intelligence

Arturo Duque Marín<sup>1</sup>, Jesus A. Lopez<sup>2</sup>, Andres Felipe Navas<sup>3</sup>

<sup>1</sup>Ingeniero Mecatrónico, Estudiante, Universidad Autónoma de Occidente, Cali, Colombia

<sup>2</sup>Doctor en Ingeniería, Docente Tiempo Completo, Universidad Autónoma de Occidente, Cali, Colombia

<sup>3</sup>Ingeniero Mecatrónico, Universidad Autónoma de Occidente, Cali, Colombia

E-mail: jalopez@uao.edu.co

Recibido 08/09/2016  
Aceptado 10/11/2016

Cite this article as: A. Duque, J.A. López, A.F. Navas, "Auto-tuning of a PID controller implemented in a PLC using swarm intelligence", *Prospectiva*, Vol 15, N° 1, 35-41, 2017.

### RESUMEN

Este trabajo muestra la implementación en un PLC Allen Bradley de una técnica de inteligencia de enjambres cuya función es determinar el modelo matemático de un sistema o proceso. Con el modelo del sistema encontrado, se calculan los parámetros de un controlador PID que garantizan un comportamiento deseado del sistema. La técnica de inteligencia de enjambres usada para la obtención del modelo del proceso es conocida como optimización por enjambre de partículas.

Para probar el funcionamiento del algoritmo implementado en el PLC, se realizaron simulaciones de sistemas dinámicos en un computador de escritorio con el software Matlab, desde este software se estableció conexión con el PLC usando como estándar de comunicación el OPC (OLE for Process Control). De esta manera, Matlab envía información sobre el estado del proceso, y el PLC estima el modelo matemático del sistema, sintoniza el controlador PID para luego enviar al PC una acción de control adecuada. Los resultados mostraron que al comparar el funcionamiento de un PID fijo con el PID auto-sintonizado sus desempeños son similares lo cual es un buen punto de partida para futuras mejoras al esquema presentado. Una de las principales conclusiones de este trabajo es la posibilidad de utilizar algoritmos de control adaptivo usando inteligencia de enjambres en un ambiente industrial.

**Palabras clave:** Automatización; PLC; Inteligencia artificial; PID; Inteligencia de enjambres; Auto-Sintonía.

### ABSTRACT

This project implements a swarm intelligence algorithm in an Allen Bradley PLC. The task of the algorithm is to estimate a mathematical model of a system or process to allow finding parameters of a PID controller. The algorithm is named particle swarm intelligence and it emulates the behavior of a flock of birds when they are flying.

The test of the algorithm in the PLC was made using mathematical models of systems that were simulated in Matlab running over a personal computer (PC). So, a connection between the PLC and PC was made using OPC (OLE for Process Control). This way, Matlab sends information about the state of the process, and the PLC estimates the mathematical model of the system, tuning the PID controller and then it sends to PC a suited control action. The results show that the performance of the system controlled with the adaptive PID is similar to the system controlled using a standard PID. This project shows the potential of to use adaptive control with swarm intelligence on industrial environments.

**Key words:** Automation; PLC; Artificial intelligence; PID; Swarm intelligence; Auto-Tuning.

## 1. INTRODUCCIÓN

En la industria, la técnica de control más usada es el PID, esto se debe gracias a su simplicidad y efectividad durante su funcionamiento. A pesar de sus grandes ventajas, los controladores PID en muchas ocasiones no son sintonizados de tal forma que alcancen su máximo potencial, para lograr esto se requiere de un amplio conocimiento en el comportamiento del sistema a controlar y del efecto de las acciones proporcional, integral y derivativa. Debido a esto, se hace necesaria la presencia de un técnico o ingeniero con una gran experticia en el tema para poder obtener del PID su mejor rendimiento. Teniendo en cuenta lo anterior, a nivel industrial se han desarrollado herramientas para auto sintonizar los controladores PID, que definen los parámetros del controlador y permiten su óptimo funcionamiento.

Una manera de sintonizar el controlador PID es identificando los parámetros del proceso y con ellos, calcular las constantes del controlador que puedan permitir la mejor acción de control. De esta manera, se entra en el campo del control adaptativo, que como su nombre lo indica, es aquel que puede modificar su comportamiento en respuesta a cambios en la dinámica de un sistema [1]. Haciendo uso de este tipo de esquemas, es posible controlar sistemas que presentan comportamientos no lineales o que sufren de perturbaciones durante su funcionamiento, ocasionando el cambio de parámetros que definen el funcionamiento del proceso.

Por otro lado, la inteligencia de enjambres es un área de estudio de la computación evolutiva que se basa en el comportamiento colectivo de sistemas descentralizados y auto-organizados. El término Inteligencia de Enjambre fue introducido por Gerardo Beni y Jing Wang en un trabajo sobre Sistemas Celulares Robóticos [2]. En general, los algoritmos bioinspirados han sido desarrollados con base en estudios de los comportamientos de organismos vivos frente a situaciones de búsqueda de alimentos o recursos para su supervivencia. En 1995 James Kennedy y Russ C. Eberhart desarrollaron el método de optimización de Enjambres de Partículas (PSO – Particle Swarm Optimization), donde experimentaron con algoritmos que modelaban el comportamiento del vuelo de algunos pájaros o el movimiento de los bancos de peces. Este comportamiento colectivo que emerge de un grupo de agentes sociales ha sido llamado “Inteligencia de Enjambres” [3, 4].

El esquema de control adaptativo usado en este trabajo fue propuesto en [5]. En dicho trabajo se usó como técnica para estimar los parámetros del sistema una optimización por forrajeo bacteriano y como esquema de control un controlador por equivalencia. Los autores

[6] tomaron el esquema de control adaptativo anterior e implementaron el PSO para estimar los parámetros del sistema y como controlador el PID. Es importante resaltar que los resultados mostrados en [5] y [6] corresponden a simulaciones mientras en el presente trabajo, se plantea un esquema de implementación con herramientas usadas a nivel industrial.

En la literatura se han reportado otras aplicaciones del PSO en el campo del control automático. En [7] se usa PSO para estimar los parámetros de un controlador PID para un motor sin escobillas. En [8] se realiza una comparación entre PSO y un algoritmo genético para la sintonización de controladores PID.

En el campo del control adaptativo se ha usado variaciones de PSO para encontrar los parámetros de un sistema no lineal y de un controlador PID [9], también se ha usado PSO para estimar los parámetros de una PID difusa [10]. Algo que tienen en común los trabajos mencionados, es que el análisis se realiza haciendo uso de simulaciones en contraste, en este trabajo se muestra como realizar la auto-sintonización de un control PID en con herramientas industriales como son los PLC y el estándar de comunicación OPC (OLE for Process Control).

Este trabajo se ha organizado de la siguiente manera: inicialmente se muestran los conceptos teóricos necesarios utilizados en el artículo, a continuación, se describe la metodología usada para validar la propuesta del trabajo, luego se describe los resultados de las pruebas realizadas para finalizar con las conclusiones.

## 2. TEORÍA

### 2.1. Optimización por Enjambres de Partículas (PSO)

Este tipo de técnica se caracteriza por modelar el comportamiento de bandadas de aves y cardúmenes donde cada individuo corresponde a una posible solución al problema. Estos individuos o partículas se mueven a través de un espacio multidimensional, en donde cualquier cambio de estado o posición que haga una partícula son influenciados por la experiencia o conocimiento de sus vecinos [11].

El esquema básico para representar este tipo de algoritmos inicia con la definición de las partículas. Cada partícula está constituida por un vector de  $D$  dimensiones (número de variables del problema), una posición  $x_i$ , y una velocidad  $v_i$  que se encarga de definir el comportamiento de la partícula [4], tal como se muestra a continuación en la ecuación 1:

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (1)$$

$i$  define la  $i$ -ésima partícula.

$d$  define la  $d$ -ésima componente o variable de la partícula.

La ecuación 2, define la velocidad para las partículas:

$$v_{id}(t+1) = \omega \cdot v_{id}(t) + \phi_1 \cdot rand_1 \cdot (pBest_i - x_{id}(t)) + \phi_2 \cdot rand_2 \cdot (gBest - x_{id}(t))$$

Para el cálculo de velocidad de la partícula se tienen tres componentes. El primer término  $\omega \cdot v_{id}(t)$ ; le da a la partícula una inercia, de tal manera que la partícula tienda a seguir el comportamiento que llevaba y así amortiguar la velocidad de la misma [6]. El término  $\phi_1 \cdot rand_1(pBest_i - x_{id}(t))$ ; permite que la velocidad de la partícula sea influenciada por las mejores posiciones que la misma ha tenido durante su recorrido por el espacio de búsqueda, a este término se le conoce como componente cognitiva [10], y el término  $\phi_2 \cdot rand_2(gBest - x_{id}(t))$ ; permite que la partícula pueda dirigirse hacia las mejores posiciones que han encontrado otras partículas dentro del enjambre, a este término se le conoce como el componente social [8]. De esta manera, si una partícula encuentra una muy buena posición, las otras tendrán la tendencia a dirigirse hacia ese punto óptimo encontrado.  $\omega$  representa la constante de inercial,  $\phi_1$  representa el coeficiente cognitivo o local y  $\phi_2$  representa el coeficiente social.  $pBest$  representa la mejor posición de la partícula y  $gBest$  representa la mejor solución que se ha encontrada hasta el momento.

El pseudocódigo del algoritmo usado en este trabajo se muestra en la figura 1.

**Figura 1.** Pseudocódigo del Algoritmo PSO gBest.

**Figure 1.** Pseudocode of PSO Algorithm PSO.

---

```

S ← Inicializar Cúmulo()
While no se alcance la condición de parada do:
    For i = 1 to size(S) do:
        Evaluar cada particular  $x_i$  del cúmulo S
        if  $fitness(x_i)$  es mejor que  $fitness(pBest_i)$  then:
             $pBest_i \leftarrow x_i$ ;  $fitness(pBest) \leftarrow fitness(pBest_i)$ 
        end if
        if  $fitness(pBest_i)$  es mejor que  $fitness(gBest)$  then:
             $gBest \leftarrow pBest_i$ ;  $fitness(gBest) \leftarrow fitness(pBest_i)$ 
        end if
    end for
    for i = 1 to size(S) do:
         $v_i \leftarrow \omega \cdot v_i + \phi_1 \cdot rand_1 \cdot (pBest_i - x_i) + \phi_2 \cdot rand_2 \cdot (gBest - x_i)$ 
         $x_i \leftarrow x_i + v_i$ 
    end for
end while
Salida: la mejor solución encontrada

```

---

Fuente: J. M. García

## 2.2 OPC (OLE For Process Control)

OPC [13] es un estándar para la conectividad de datos usado en la industria para facilitar la comunicación entre sistemas de control industrial y aplicaciones de diferentes características, permitiendo el envío y la recepción de datos sin que sea necesaria la compatibilidad entre los drivers de comunicación de los equipos.

De esta manera, OPC es un conjunto de interfaces estándar basado en tecnología OLE/COM de Microsoft. La aplicación de la interface estándar OPC hace posible la interoperabilidad entre las aplicaciones de automatización/control, sistemas/dispositivos de campo etc.

## 3. METODOLOGÍA

### 3.1 Ajuste de parámetros del PSO

Para garantizar un correcto funcionamiento del PSO hay que ajustar los parámetros del mismo. En el presente trabajo se realizaron varias pruebas para estimar los modelos de diferentes procesos. La tabla 1 muestra los parámetros obtenidos.

**Tabla 1.** Parámetros de Funcionamiento del Algoritmo de Enjambres.

**Table 1.** Parameters of the Swarm Algorithm.

Símbolo	Valor	Descripción
$\omega$	0.5	Peso Inercial
$\phi_1$	1	Componente "Local"
$\phi_2$	3	Componente "Social"
$S$	50	Tamaño del Enjambre
$I$	12	Iteraciones

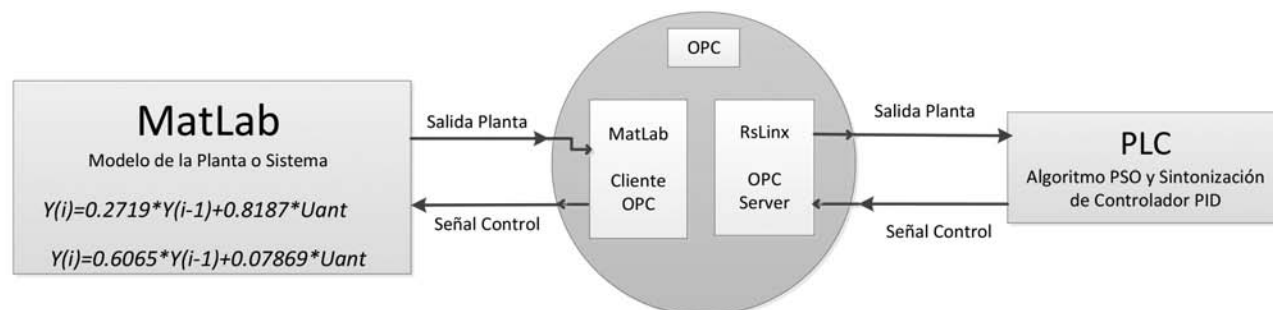
Fuente: los autores

### 3.2 Esquema experimental propuesto

Para realizar las pruebas experimentales se trabajó con un PLC Compact Logix L32E. En algoritmo fue implementado haciendo uso de lenguaje estructurado con el que cuenta dicho PLC. Para permitir interactuar el PLC con otros dispositivos se usó el protocolo de comunicación OPC. Así, en un computador personal, con Matlab se simuló el comportamiento del sistema de tal manera que la salida del mismo es enviada a través de este protocolo hasta el PLC. En el PLC se estiman los parámetros de la planta, se sintoniza un PID de acuerdo a los mismos y se genera la señal de control que es enviada de nuevo al computador personal para calcular la nueva salida de la planta con el modelo implementado en Matlab. En la figura 2 se muestra el esquema de funcionamiento descrito.

**Figura 2.** Esquema experimental propuesto.

**Figure 2.** Proposal experimental scheme.

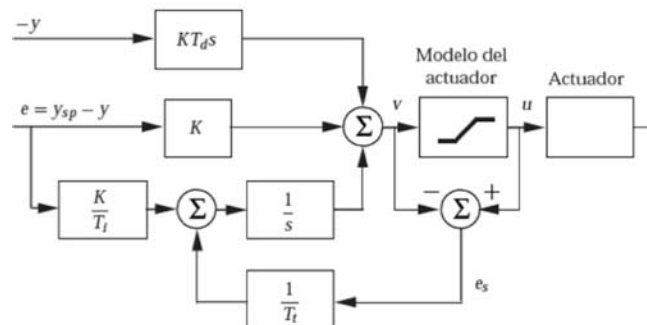


**Fuente:** los autores

Para evitar problemas con el esquema de control, que como se ha mencionado corresponde a un controlador PID, se adicionó una estructura anti-windup [14] con el propósito de hacer que la acción integral no se saturará y ocasionará de esta manera un comportamiento no deseado en el proceso. La figura 3 muestra el esquema usado.

**Figura 3. Esquema Anti-Windup.**

Figure 3. Anti-WindUp Scheme.



**Fuente:** K. J. Åström, T. Hägglun [14]

## 4. RESULTADOS Y DISCUSIÓN

Con el esquema experimental definido, se realizaron diferentes pruebas para verificar el comportamiento del algoritmo propuesto. En estas pruebas se obtienen gráficas del comportamiento del controlador, así como de la respuesta de la planta cuando se definen diferentes referencias, de igual manera se estudia el algoritmo cuando se presenta un cambio de planta o proceso mientras este está funcionando. Se genera además, una comparación entre un controlador PID sin la sintonización y un PID sintonizado.

Para las pruebas realizadas se usó la información de la tabla 2 que representa el comportamiento deseado en el proceso.

**Tabla 2.** Comportamiento deseado del proceso.

**Table 2.** Desired behavior of the process.

Características deseadas	Ecuación característica
M <sub>p</sub> =2% t <sub>s</sub> =3 Seg	Z <sup>2</sup> + 1.74*Z +0.7661

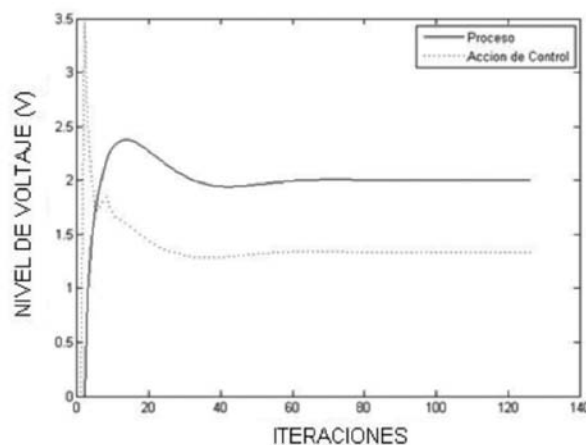
**Fuente:** los autores

#### 4.1. Seguimiento de una referencia constante

La primera prueba consistió en dejar la referencia constante, en la figura 4 se muestra el comportamiento del proceso y la acción de control ejercida por el algoritmo durante su ejecución.

**Figura 4.** Salida del proceso y acción de control con una referencia constante.

**Figure 4.** Process output and control action when the reference is constant.



**Fuente:** los autores

La gráfica muestra la capacidad del algoritmo para modelar el sistema, sintonizar el controlador PID y generar la acción de control necesaria sobre el proceso.

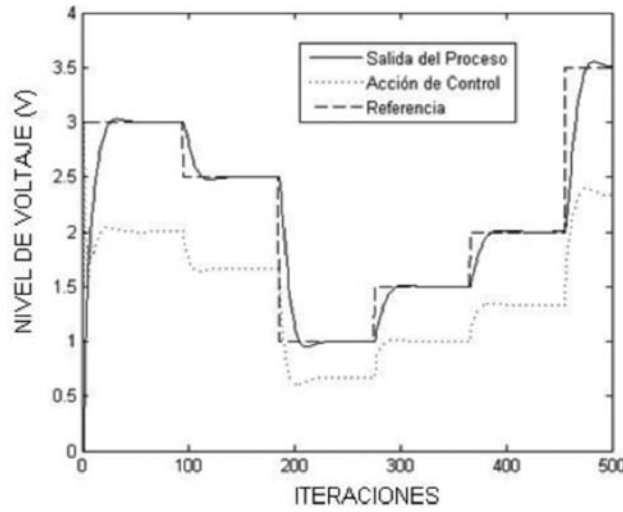


#### 4.2 Seguimiento de una referencia variable

La segunda prueba consistió en someter el algoritmo a cambios en la referencia que el sistema controlado debe seguir. En la figura 5 se presenta el comportamiento del sistema cuando se realizó esta prueba.

**Figura 5.** Salida del proceso y acción de control con cambios de referencia.

**Figure 5.** Process output and control action when the reference is changed.



Fuente: los autores

Con las dos pruebas realizadas anteriormente, se puede afirmar que el esquema propuesto funciona adecuadamente, permitiendo estimar el modelo del sistema y llevar el proceso hasta el punto de consigna deseado.

#### 4.3 Seguimiento de una referencia constante con modificación del sistema

La tercer prueba consistió en dejar la referencia constante, pero modificar la dinámica del sistema para comprobar la habilidad del algoritmo PSO para estimar parámetros del proceso. La figura 6 ilustra este comportamiento.

En la figura 6 se observa como a pesar de cambiar la dinámica del sistema a controlar, el algoritmo es capaz de generar las modificaciones necesarias para mantener la salida en el valor deseado.

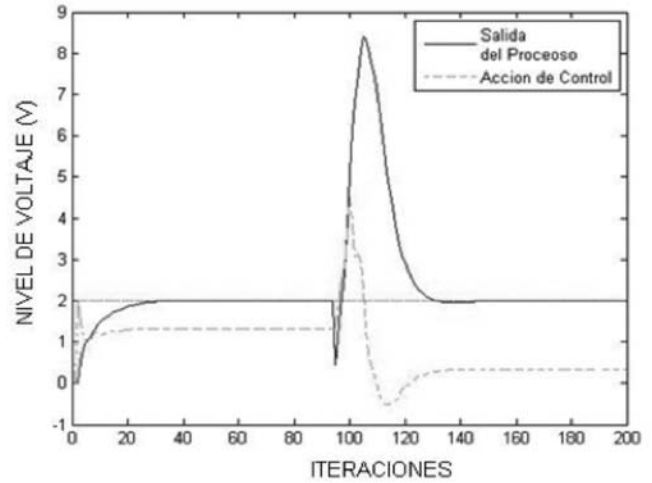
#### 4.4. Comparación PID adaptativo vs PID convencional

Ahora, se muestra el análisis comparativo que se realizó entre un controlador PID sin la sintonización y PID sintonizado con el esquema propuesto, para esta prueba comparativa el controlador PID se sometió cambios de referencias y al cambio de proceso.

Las figuras 7 y 8 muestran los resultados de trabajar con un PID fijo o convencional.

**Figura 6.** Salida del proceso y acción de control con cambios de sistema.

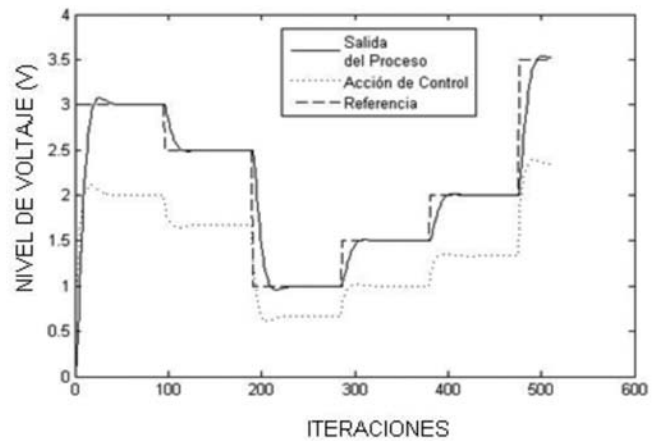
**Figure 6.** Process output and control action when the process is changed.



Fuente: los autores

**Figura 7.** Salida del proceso y acción de control con controlador PID fijo.

**Figure 7.** Process output and control action with a constant PID.



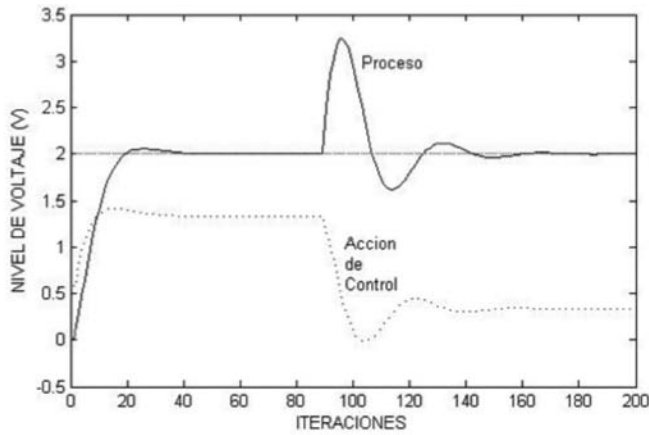
Fuente: los autores

Cuando se generan cambios de referencia, gráficamente no se alcanza a percibir diferencias entre el controlador PID sin la estimación de parámetros y el PID auto-sintonizado, quizá la diferencia más notable se presenta en el comportamiento que presentan ambos en la prueba donde se hace un cambio de planta.

Para poder comparar de manera cuantitativa el comportamiento del controlador y del algoritmo se calcularon los siguientes índices de desempeño [15], donde  $r(k)$  es el valor en  $k$ -ésimo instante de muestreo de la referencia,  $y(k)$  es el valor en  $k$ -ésimo instante de muestreo de la salida del sistema,  $u(k)$  es el valor en  $k$ -ésimo instante de muestreo de la acción de control y  $N$  es la cantidad de muestras obtenidas durante la prueba:

**Figura 8.** Salida del proceso y acción de control con controlador PID fijo cuando el proceso cambia.

**Figure 8.** Process output and control action with a constant PID when the process is changed.



**Fuente:** los autores

**Índice de desempeño de error:** este se encarga de evaluar la diferencia entre el valor deseado y el valor obtenido en la salida. Se definen dos índices, el promedio del error al cuadrado (ecuación 3) y el promedio del error absoluto (ecuación 4).

$$I1 = \frac{1}{N} \sum_{K=0}^N (r(k) - y(k))^2 \quad (3)$$

$$I2 = \frac{1}{N} \sum_{K=0}^N |r(k) - y(k)| \quad (4)$$

**Índice de desempeño de esfuerzo de control:** evalúa el promedio del valor absoluto de la acción de control (ecuación 5).

$$I3 = \frac{1}{N} \sum_{K=0}^N |u(k)| \quad (5)$$

**Índice suavidad esfuerzo de control:** compara la diferencia entre una muestra anterior de la señal de control y la actual (ecuación 6). Permite conocer que tan bruscos son los cambios de la señal de control.

$$I4 = \frac{1}{N-1} \sum_{K=0}^N (u(k) - u(k-1))^2 \quad (6)$$

En la tabla 3 se muestran los resultados de evaluar los índices presentados cuando cambia la referencia.

La tabla 3 muestra de manera clara que no se presenta

un comportamiento muy diferente entre el controlador y el PSO con el PID, sin embargo, gráficamente se alcanza a percibir que el algoritmo PSO impide la oscilación del sistema cuando este se acerca a la referencia deseada.

Para analizar el comportamiento de ambos algoritmos cuando se cambia el proceso se tiene la tabla 4 donde se muestra los índices de desempeño para esa condición.

**Tabla 3.** Índices de desempeño ante cambios de referencia.

**Table 3.** Performance indexes when the reference is changed.

INDICE	PID Auto-sintonizado	PID Fijo
$I_1$	0,1258	0,1321
$I_2$	0,1175	0,1127
$I_3$	1,4222	1,3957
$I_4$	0,0209	0,0015

**Fuente:** los autores

**Tabla 4.** Índices de desempeño ante un cambio de proceso.

**Table 4.** Performance indexes when the process is changed.

ÍNDICE	PID Auto-sintonizado	PID Fijo
$I_1$	2,045	0,1688
$I_2$	0,5179	0,1756
$I_3$	0,912	0,7728
$I_4$	0,0487	7,92E-04

**Fuente:** los autores

El sobre pico que se observa en la figura 6 hace que el valor de los índices sea más elevado que los mostrados por el controlador PID sin la estimación, es posible decir que el algoritmo PSO permite la estimación rápido del modelo del proceso y lleva a cabo el control del mismo sin inconveniente alguno. Cabe resaltar que el algoritmo PSO con el controlador no deja que el sistema oscile y en cambio lleva el proceso al punto de consigna en menor tiempo que el usado por el controlador PID sin estimación.

## 5. CONCLUSIONES

Dentro del ámbito de control industrial el PLC es el dispositivo electrónico por excelencia y la necesidad de desarrollar algoritmos sobre los mismos, que sean cada vez más versátiles y útiles para diversas plantas o sistemas se hace prioritaria.

El uso de algoritmos de inteligencia de enjambres puede brindar solución a los problemas de sintonización de un controlador PID, permitiendo así la oportunidad de controlar sistemas no lineales o con dinámicas muy complejas.

Al momento de la implementación se hizo posible observar que estos algoritmos pueden brindar robustez al control de una planta; ya que en las pruebas respondió bien ante cambios y a diferentes condiciones y que aún con la gran cantidad de variables que necesita y del aparente alto requerimiento en el procesamiento, el PLC respondió adecuadamente y no presentó inconveniente alguno para ejecutar el algoritmo completo.

Uno de los principales aportes de este trabajo es mostrar la viabilidad de implementar técnicas de control basadas en inteligencia de enjambres en ambiente industriales. Lo anterior se fundamenta en el uso de un PLC y un estándar de comunicación como el OPC.

La comparación realizada con los índices de desempeño usados mostró que el comportamiento de los dos esquemas PID implementados, el fijo y el adaptativo, presentan características similares. Lo anterior sirve de base para futuros trabajos donde usando comparaciones similares se puede llegar a una PID adaptativo que sea más robusto y muestre claras ventajas sobre el PID convencional.

Como trabajo futuro se vislumbra probar otras técnicas de inteligencia de enjambres en el esquema de control adaptativo propuesto. Con las diversas técnicas implementadas se puede generar una comparación para saber cuál es la que mejor responde para este tipo de aplicación.

## REFERENCIAS

[1] Colaboradores de Wikipedia (2016, Ago.) Adaptive Control [Internet]. Wikipedia, La enciclopedia libre. Disponible desde: <[https://en.wikipedia.org/wiki/Adaptive\\_control](https://en.wikipedia.org/wiki/Adaptive_control)> [Acceso 04 de agosto 2016].

[2] G. Beni, J. Wang, "Swarm Intelligence in celular robotic systems". En: Robots and Biological Systems: Towards a New Bionics? Volume 102 of the series NATO ASI Series, Springer Berlin Heidelberg, p. 703-712, 1993.

[3] E. Bonabeau, G. Theraulaz, "Swarm Smarts", Scientific American, 282 (3), 72-79, 2000.

[4] M. Muñoz, J. López, E. Caicedo, "Inteligencia de Enjambres: Sociedades para la solución de problemas (una revisión)", Ingeniería e Investigación, 28(2), 119-130, 2008.

[5] K. Passino, "Biomimicry of bacterial foraging for distributed optimization and control", IEEE Control Systems Magazine, 22(3), 52-67, 2002.

[6] A. F. Navas, J. A. López (2013, Oct.) Inteligencia de enjambres aplicada al control adaptativo. II Congreso Internacional de Ingeniería Mecatrónica y Automatización. Bogotá. 2013.

[7] H.E.A. Ibrahima, F.N. Hassanb, A. O. Shomerc, "Optimal PID control of a brushless DC motor using PSO and BF techniques", Ain Shams Engineering Journal, 5(2), 391-398. 2014.

[8] C. R. López, "Evaluación de desempeño de dos técnicas de optimización bio-inspiradas: Algoritmos Genéticos y Enjambre de Partículas", Revista Tekhnê, 11(1), 49-58, 2014.

[9] A. Alfi, H. Modares, "System identification and control using adaptive particle swarm optimization", Applied Mathematical Modelling, 35(3), 1210-1221, 2011.

[10] J. Chiou, S. Tsai, M. Liu, "A PSO-based adaptive fuzzy PID-controllers", Simulation Modelling Practice and Theory, 26, 49-59, 2012.

[11] A. Engelbrecht, Computational Intelligence An Introduction, Universidad de Pretoria, South Africa: John Wiley & Sons Ltd, Edicion 2, 2007.

[12] J. M. García, (2006) Algoritmos Basados en Cúmulos de Partículas para la Resolución de Problemas Complejos. Proyecto Fin de Carrera (Ing. Informática), Universidad de Málaga (España).

[13] Colaboradores de Wikipedia (2016, Feb) OPC [Internet]. Wikipedia, La enciclopedia libre. Disponible desde: <<https://es.wikipedia.org/wiki/OPC>> [Acceso 1 de abril 2016].

[14] K. J. Åström and T. Häggglun, Control PID Avanzado, España: Pearson Educación S.A. , 2009, pp. 80-91

[15] G. E. Polanco, O. E. Soto, J. A. López (2013, Oct.) Análisis comparativo de técnicas de control convencional e inteligente con los sistemas de articulación flexible y bola viga. II Congreso Internacional de Ingeniería Mecatrónica y Automatización. Bogota. 2013.