

Sintonización de un PID Siemens® utilizando algoritmos evolutivos

Henry A. Hernández Martínez¹, Karen N. Florido Angarita², Lely Luengas³

heahernandezma@unal.edu.co, knflorida@correo.udistrital.edu.co, lelyluco@gmail.com

Universidad Nacional de Colombia¹, Universidad Distrital Francisco José de Caldas²³
Cl. 68D Bis A Sur N° 49F - 70 Candelaria La Nueva
Bogotá - Colombia

Resumen: Este artículo muestra el desarrollo de una aplicación para la sintonización de un controlador PID, utilizando una estrategia basada en algoritmos evolutivos. Dicha estrategia, al pasar 50 generaciones (las cuales tardan 60 minutos en establecerse), encuentra un valor aceptable para sintonizar el controlador. Se realizó una simulación del proceso teniendo en cuenta el modelo discreto del proceso a controlar, la cual permite evaluar el desempeño de cada generación, con el fin de establecer y comprobar los mejores valores de las constantes Proporcional, Integral y Derivativa que mantienen el sistema en un estado estable. Las constantes halladas usando este método fueron probadas en un PLC Siemens® S7-300, realizando un acople a un sistema de control proceso, en el cual se controla la variable de proceso "caudal" a través de una válvula proporcional y un transmisor de flujo. Este sistema permite la sintonización de un controlador sin tener un conocimiento experto a diferencia de los métodos convencionales de ensayo y error (Zieger Nichols, Cohen y Coon, entre otros).

Palabras clave: PID, Controlador, Amortiguado, Estabilidad, LTI, Variable Discreta.

Abstract: This article shows the development of an application for tuning a PID controller using a strategy based on evolutionary algorithms, such a strategy to spend 50 generations (which takes 60 minutes to establish), finds an acceptable value to tune the controller. The simulation of the process was carried out taking into account the discrete model of process control, which allows to evaluate the performance of each generation, in order to establish and check the best values of the proportional constants, Integral and Derivative maintaining the system a stable state. The constants found by this method were tested in a S7-300 PLC Siemens®, performing a coupling to a process control system in which the process variable "flow" is controlled via a proportional valve and a flow transmitter. This system allows tuning a controller without expert knowledge, unlike conventional methods of trial and error (Zieger Nichols, Cohen and Coon, among others).

Keywords: PID controller, Cushioned, stability, LTI, Discrete Variable.

1 Introducción

Los controladores PID tienen gran acogida en la industria debido a que son estrategias de control versátiles y robustas que se pueden aplicar casi en cualquier proceso de lazo cerrado.

Algunos procesos donde son utilizados estos controladores suelen desestabilizarse debido al ingreso de perturbaciones no lineales al sistema, las cuales pueden ser representadas por el desgaste por uso de los equipos, vibraciones o condiciones nuevas de funcionamiento, lo que ocasiona un desajuste del controlador para tratar de adaptarse a sus nuevas condiciones de funcionamiento.

Se presentan situaciones donde la tarea de sintonización no es fácil, pues un factor influyente es la complejidad del sistema y a veces encontrar las mejores constantes de funcionamiento no es una tarea sencilla, ya que algunos de los métodos de sintonización se basan en el ensayo y error [1].

Este trabajo muestra la sintonización de controladores PID utilizando una técnica de actualización de constantes con ayuda de algoritmos evolutivos, con el fin de proponer una nueva estrategia que permita el control de un sistema, así la información de dicho sistema sea poca o nula. La propuesta tiene como finalidad emular un proceso de auto-tuning de controladores que actualmente se encuentra disponible en herramientas computacionales destinadas para tal fin [2].

Para la sintonización, en primer lugar, se propone un conjunto de soluciones iniciales, en segundo lugar, se seleccionan de acuerdo con su adaptabilidad al medio, tercero son transformadas para mejorar las soluciones actuales y, por último, evaluadas de acuerdo con el objetivo del problema.

2 Estado del arte

El área de control de procesos que trabaja la sintonización de lazos ha propuesto varias metodologías para llevar a cabo este proceso, algunas de ellas se basan en las curvas de reacción [3], otros métodos son gráficos y trabajan con ayuda de una simulación por computadora [4] y se tienen algunos métodos matemáticos que se basan en la ecuación general de un sistema [5].

Se encuentra presente una metodología para la implementación de controladores basados en lógica difusa en un PLC S7-300 empleando el lenguaje de programación SCL (Lenguaje de control estructurado) de STEP 7. Esto se realiza mediante la elaboración de la función difusa, la cual es embebida dentro del controlador para realizar el control de un sistema cualquiera. Este esquema permite la implementación de controladores adaptativos o autosintonizados y, por último, muestran un ejemplo de aplicación para un sistema de tiempo discreto simulado en el PLC, empleando un controlador difuso PI [6].

Algunos autores no realizan la tarea de autoadaptación del controlador sino que prefieren realizar un monitoreo

continuo del sistema de control, entre los cuales se encuentra un proyecto que detecta, de una manera eficiente, las alarmas que se presentan en la línea de producción. Esto se logró mediante el uso de pantallas del HMI las cuales tienen una interfaz gráfica destinada para tal fin y se trabajó el software S7 haciendo uso de una red PROFIBUS para intercomunicar 2 PLC S7 300, un PLC S7 400 y un HMI MP377 de 12" [7].

Realizando una búsqueda paralela a este tema, se encuentra que se han propuesto algunos algoritmos genéticos aplicados en el área de computación evolutiva con el objetivo de obtener mejor rendimiento y soluciones a problemas de optimización con un alto grado de dificultad. Los algoritmos genéticos fueron propuestos por el profesor John Holland (1992). Están basados en la selección natural presente en la naturaleza y son una herramienta potente para la resolución de problemas de optimización [8].

Existen trabajos donde se ha aplicado la teoría de algoritmos genéticos. Por ejemplo, un proyecto que tiene como objetivo diseñar un sistema multiplataforma de verificación de riego y escalable de control integrado a partir de una maqueta que puede ser visualizada en internet, donde el término escalable indica la posibilidad de implantación del sistema desarrollado tanto a nivel de laboratorio como a nivel industrial [9].

Otros trabajos que combinan el desarrollo de controladores inteligentes (control fuzzy, redes neuronales, entre otras.) en el control de movimiento en plataformas robóticas. Los cuales emulan un comportamiento de aprendizaje adaptativo a partir de la experiencia usando algoritmos que aseguren la resolución de los problemas que pueden ser solucionados por un robot. Teniendo en cuenta que una de las limitantes en la realización de dichas tareas es la cantidad de recursos necesarios para poder ejecutar el código [10].

También se puede encontrar un método de modelado paramétrico de estructuras de barras para fabricación aditiva diseñado para piezas de dimensiones considerables. Seguidamente, a partir del modelado paramétrico anterior, se realizaron otros trabajos de gran interés centrados en la optimización de la estructura de barras, con el fin de completar una metodología que permita obtener piezas más ligeras y con buenas propiedades mecánicas [11].

3 Metodología

Para la realización de este proyecto se siguieron una serie de pasos, empezando por la caracterización del sistema, luego una simulación por computadora y por último la validación utilizando un PLC Siemens® de la gama S7-300, la estrategia utilizada se observa en la Figura 1.

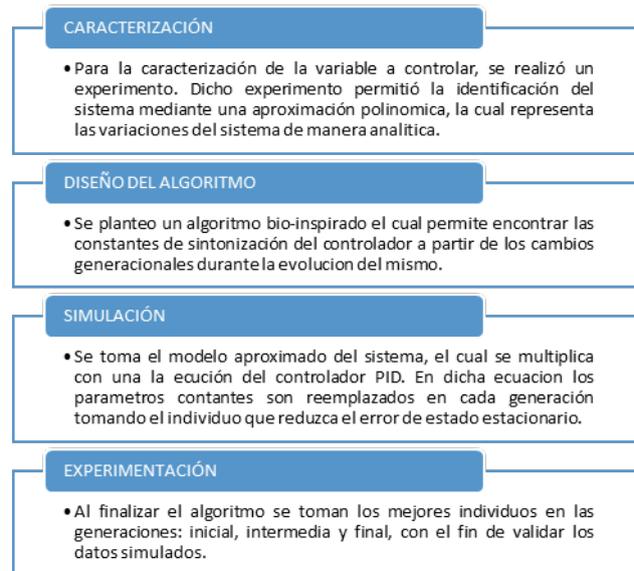


Figura 1: Estrategia utilizada

3.1. Caracterización

Para la caracterización del sistema, se realizó una aproximación matemática utilizando una herramienta computacional destinada a la identificación de sistemas. Los datos se obtuvieron a través de un experimento en el sistema de control de proceso T5554 de la marca AMATROL (Figura 2).

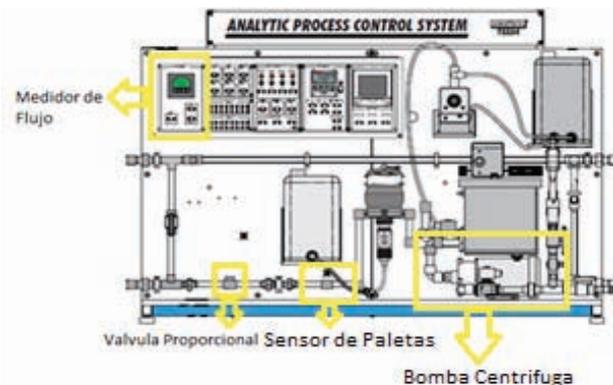


Figura 2. Sistema de Control de Proceso T5554 [12]

El sistema de control de proceso T5554 es un banco de pruebas en el cual se controlan las variables de proceso pH y flujo. Dicho sistema cuenta con sensores y actuadores de tipo industrial para controlar dichas variables, sin embargo, este artículo muestra los resultados obtenidos controlando la variable de proceso "flujo".

El flujo de agua impulsado por una bomba centrífuga es controlado a través de una válvula proporcional que funciona de 4 a 20mA y medido con un sensor de paletas. Dicho sensor envía la señal a un medidor de flujo el cual muestra el valor del flujo actual y envía una señal de corriente que varía linealmente con respecto al flujo (4mA = 0GPM y 20mA = 2GPM).

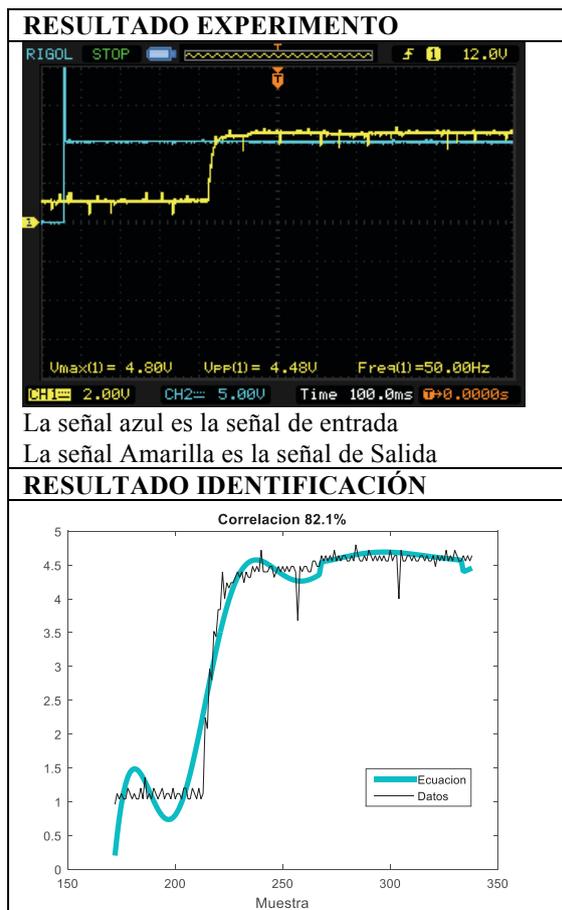
El experimento consiste en aplicar una señal a la válvula proporcional. Dicha señal emula una señal paso que varía

el caudal desde 0GPM hasta un 1GPM. Los datos son registrados con ayuda de un osciloscopio marca RIGOL DS1052-E, cuyos valores registrados son guardados en un formato de texto plano con el fin de ser procesados posteriormente.

La expresión matemática (1), que representa el comportamiento del sistema, obtuvo un coeficiente de correlación del 82,1%, como puede observarse en la Tabla 1, con un tiempo de muestreo de 10m.

$$H(z) = \frac{1665 - 3329z^{-1} + 1664z^{-2}}{1 - 2.09z^{-1} - 2.82z^{-2} - 0.91z^{-3}} \quad (1)$$

Tabla 1. Identificación del sistema



Para realizar la caracterización del PID, se utilizó la expresión (2) [13]. Esta expresión permite emular las características de un bloque PID de la marca Siemens, usado en los PLC S7-300 CPU 313C.

$$up = P + I * T_s \left(\frac{1}{z-1} \right) + D \left(\frac{N}{1 + N * T_s \left(\frac{1}{z-1} \right)} \right)$$

Donde:

- up* salida del controlador
- P* ganancia controlador
- I* tiempo integral controlador
- D* tiempo derivativo controlador
- N* Coeficiente de filtrado
- T_s* tiempo de muestreo

La finalidad de emular el bloque de PID es estimar el comportamiento del sistema una vez esté implementado el PLC. Para estimar el comportamiento del sistema, se realizó una aplicación la cual muestra la respuesta de H(z) y up(t) a la señal paso (Ver figura 3).

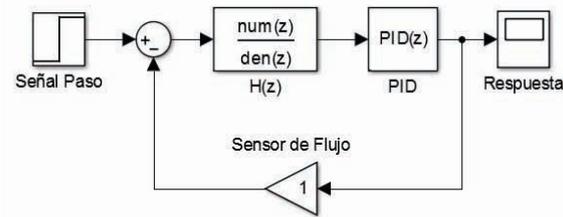


Figura 3. Diagrama de Bloques

Los valores de las constantes P, I y D son calculadas con el algoritmo diseñado. N es un valor de 100 porque la salida no se ve afectada drásticamente, como lo hace en el caso de 0, porque la salida es 0 y en 1000 la salida tiende a infinito. El tiempo de muestreo es de 100m, ya que éste fue el tiempo por división medido con el osciloscopio, en el cual se pudo visualizar mejor el comportamiento de la señal.

3.2. Diseño del algoritmo

El algoritmo se encarga de encontrar combinaciones de constantes P, I y D, seleccionando las constantes que permitan al proceso alcanzar el valor de consigna o valor esperado. La población inicial (Ver figura 4) se genera a partir de un arreglo matricial, en el cual están contenidos algunos valores aleatorios de constantes P, I y D. Estas constantes se evalúan asignando dichos valores al controlador PID y calculando el error estacionario promedio como indicador de fitness (siguiendo el esquema de la figura 3), teniendo en cuenta que la restricción del sistema es el tiempo de estabilización.

K	I	D	Fitness
1	32	12	0.1
20.3	51	65	10
2.3	468	46	0.4
5.6	90	80	0.2
-	-	-	-
-	-	-	-
n	n	n	n

→ Población

→ Individuo

Figura 4. Ejemplo de Población

Para realizar la selección de los individuos, se utilizó un operador de selección por ruleta, el cual asigna a cada individuo de la población cierta probabilidad. Seguidamente, se crea un nuevo arreglo el cual será llenado

con individuos seleccionados de manera aleatoria, con individuos que se encuentren dentro de la población.

La probabilidad se calcula mediante una función de normalización Z, aplicado a los diferentes valores de fitness de cada individuo, seguidamente se calcula un número aleatorio. Dicho valor es buscado dentro de los valores normalizados y se selecciona el individuo que tenga un valor de probabilidad cercano a ese valor permitiendo así que los individuos puedan operarse de un error cercano a cero tengan una mayor probabilidad de ser seleccionados.

Luego se propone un operador de cruce, basado en un sistema de dos puntos. Este operador selecciona dos individuos (si la probabilidad de cruce aleatoria es mayor a un umbral fijo en 0.5) de la población al azar e intercambia dos posiciones al azar como se muestra en la figura 5.

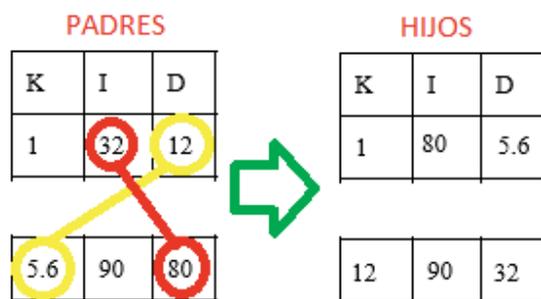


Figura 5. Ejemplo de cruce

A continuación, una mutación gaussiana selecciona un individuo de la población de manera aleatoria y cambia uno de sus valores al azar siguiendo una distribución normal (Figura 6).

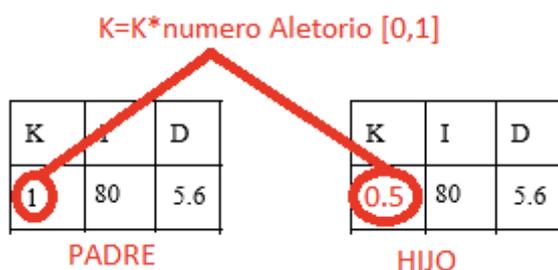


Figura 6. Ejemplo de mutación

El umbral de la mutación se hace más pequeño (siguiendo la función exponencial) a medida que el algoritmo genera poblaciones. Por último, se realiza un reemplazo poblacional utilizando la técnica de STATE STADE, que consiste en recalcular el fitness cada individuo de la nueva población y compararla con cada individuo de la población actual, seleccionando los que tengan el fitness más alto. Como resultante, se tendrá un arreglo con los mejores individuos. Dicho arreglo se convertirá en la población actual, cuyo evento se denominará reemplazo generacional. En síntesis, el algoritmo utilizado se muestra en la tabla 2.

Tabla 2. Algoritmo usado para sintonizar el PID

Algoritmo 1. Descripción del Algoritmo Evolutivo

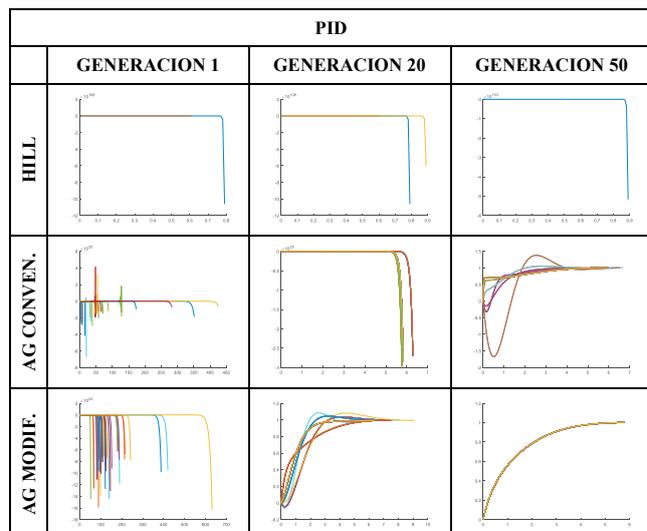
```

Programa PID ()
  Iniciar Variables;
  P0=Población Inicial U~ [0, D]; //D=Dimensiones del problema
  Mientras! =Condición de Parada
    f=evaluar fitness (Restricciones, P0);
    Padres=Selección (f, P0);
    Hijos=Cruce (Probabilidad de Cruce, f, P0);
    Si N~U [0,1]<Probabilidad de Mutación
      Hijos=Hijos-H*Sigma;
      //H~ Distribución Normal
    P0=Selección (Padres, Hijos);
    sigma=sigma*e^(-1/Amortiguación);
    Amortiguación= Amortiguación+1;
    Si Amortiguación>100, Amortiguación=1;
  fin Mientras
fin PID
  
```

4 Experimentos y Resultados

El algoritmo implementado fue comparado utilizando las estrategias evolutivas de ascenso a la colina y un algoritmo genético convencional durante la misma cantidad de iteraciones, cuyos resultados se muestran en la tabla 3.

Tabla 3. Respuestas del controlador



La primera estrategia de ascenso a la colina (Hill Climbing [14]) es una técnica de optimización matemática basado en un algoritmo de búsqueda de valores óptimos locales. Comienza con una solución arbitraria a un problema, intentando encontrar una mejor solución variando aleatoriamente un único elemento de la solución. Si la variación produce una solución más óptima, otra variación se le realiza a la nueva solución, iterando hasta que no se encuentren mejores.

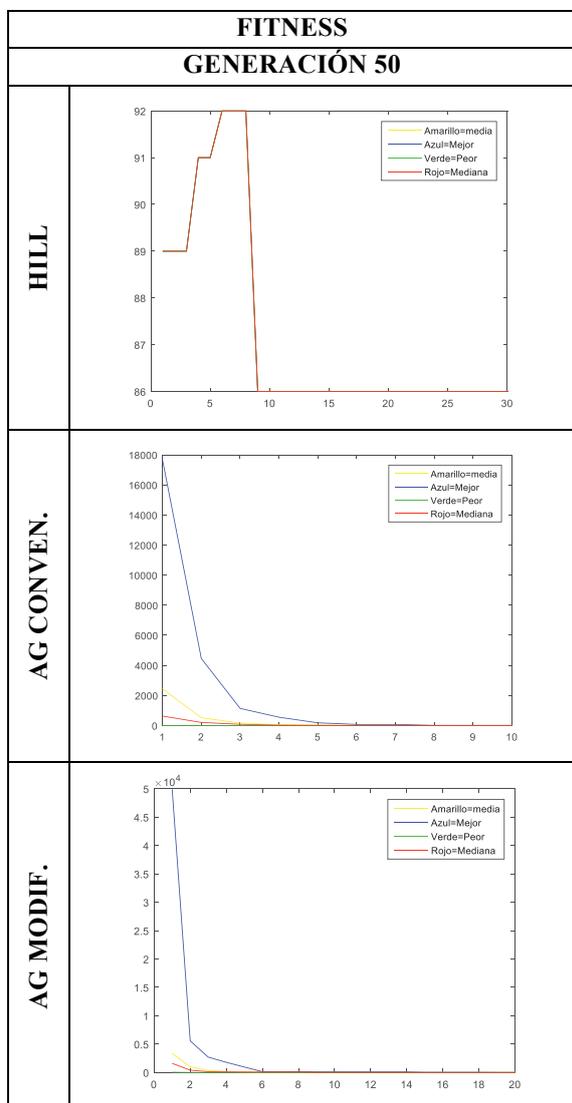
La segunda estrategia es un algoritmo genético convencional [15] (AG CONVEN), parte de un conjunto de soluciones a un problema llamado fenotipo, y un conjunto de individuos que forman una población, codificando la información de cada solución en una cadena, denominada cromosoma. Estos cromosomas

evolucionan mediante iteraciones, a las cuales se llama generación y son evaluados con una medida de aptitud o fitness. Los nuevos cromosomas generados varían aplicando operadores genéticos, siendo estos operadores de selección, cruce, mutación y reemplazo.

Por último, en la Tabla 3, se encuentra AG MODIF. El cuál es el algoritmo de elaboración propia. Como se puede observar en dicha tabla, este algoritmo logro llegar al valor de consigna en la generación 50, lo cual no fue posible con el algoritmo de ascenso a la colina y con el algoritmo genético convencional, quizás se hubiera logrado en una cantidad mayor de generaciones.

En la tabla 4, se muestra el comportamiento generacional de la función de fitness, donde está el comportamiento del mejor individuo (Error de estado estacionario bajo), el peor (Error de estado estacionario Alto) y la Mediana (Error de estado estacionario promedio en la población).

Tabla 4. Función de fitness



Los valores estadísticos numéricos de la última generación se muestran en la tabla 5, en la cual se puede observar que tan alejada se encuentra la solución propuesta por los diferentes algoritmos de la solución ideal (Tomando como ideal un valor cercano a 0).

Tabla 5. Estadística

GENERACIÓN 50					
	MEJOR	PEOR	MEDIA	MEDIANA	DESV. EST.
HILL	86	86	86	86	0
AG CONV	21.3997	3.671e+04	27.5138	27.3009	8.5668e+03
AG MODIF	2.8335	104.6696	27.8174	9.0355	1.1098e+04

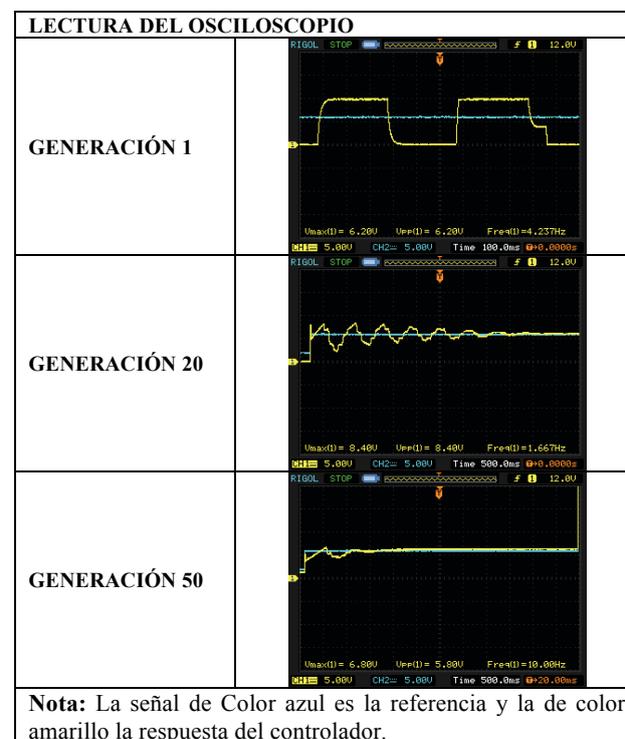
Por último, en la tabla 6, se encuentran los mejores individuos de la última generación en cada caso, con los cuales se espera que el PID se sintonice y controle la variable de proceso.

Tabla 6. Mejores constantes

ALGORITMO GENÉTICO MODIFICADO			
	Kp	Ki	Kd
GENERACION 1	0.2529	-0.7913	-0.3256
GENERACION 20	0.4673	0.8715	-0.1918
GENERACION 50	0.5062	9.1417	0.0641

Una vez terminadas las simulaciones, se tomó el mejor individuo del algoritmo de elaboración propia, en cada una de las generaciones mostradas en la tabla 7.

Tabla 7. Validación de la simulación



Nota: La señal de Color azul es la referencia y la de color amarillo la respuesta del controlador.

Seguidamente el PLC Siemens S7-300 CPU 313C, se conectó (Figura 7) al sistema de control de proceso a través de una entrada analógica graduada en el rango de 4-20mA, y así leer la señal proporcionada por el transmisor de flujo.

La válvula proporcional está sujeta a las variaciones proporcionadas por el PLC, en su salida analógica graduada en el rango de 4-20mA. Se configura el PLC en estos rangos debido a que la instrumentación dispuesta en el sistema de control de proceso trabaja en dicho rango.

Por último, se realizó un programa en lenguaje LADDER, en el cual se incorporó en la estructura de PID cuyas constantes modificaron con los valores de cada individuo,

cuya respuesta a las variaciones fueron registradas por el osciloscopio RIGOL y se muestra en la tabla 7.

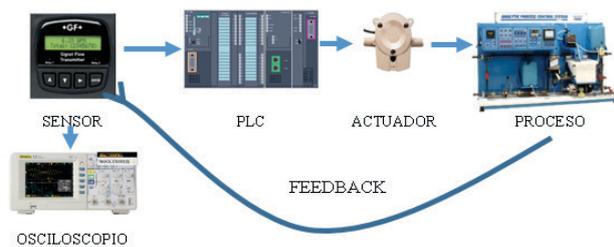


Figura 7. Diagrama del montaje realizado

5 Conclusiones

- La estrategia de búsqueda por ascenso a la colina resultó no ser eficiente para encontrar la configuración más adecuada de constantes para sintonizar el controlador PID, porque es difícil que se actualice el punto dentro del espacio de búsqueda lo que ocasiona que converja muy rápidamente y quede atrapada en un óptimo local.
- Las gráficas del algoritmo modificado y el PID implementado en la generación uno, no se parecen debido a que la representación hecha por la simulación no tiene valores de saturación, hace que tienda a infinito y al no estar dentro del espacio de búsqueda, el PID lo traduce como una oscilación.
- Esta técnica es un algoritmo bioinspirado que permite la sintonización de PID Siemens, cuya ventaja es que el usuario no necesita un conocimiento profundo del sistema de control o basarse en algún método de sintonización por ensayo y error para controlar el sistema.

6 Trabajos futuros

- Este algoritmo se puede implementar dentro de un PLC realizando las respectivas adecuaciones para que sea autorregulable y así el sistema sufra variaciones bruscas este siempre encontrara la mejor combinación de variables posible o como estrategia de autotuning propia del PLC.
- Se pueden implementar algoritmos de memoria para no estar ejecutando siempre el algoritmo genético, sino que tenga presentes cuál es o ha sido la mejor combinación de variables, además de alguna estrategia de coevolución ya que todas las constantes se pueden modificar por separado.

Referencias bibliográficas

- [1]. Minorsky, Directional stability of automatically steered bodies, Journal of the American Society of Naval Engineering, 1922.
- [2]. Bestune, PID Controller Auto-Tuning Software, [Último acceso: 12 01| 2015]. <http://bestune.50megs.com/>
- [3]. M. Alfaro, Método de identificación de modelos de orden reducido de tres puntos, Costa Rica, 2007.
- [4]. J. Luna, Control Automático, F.R. Tucumán – Universidad Tecnológica Nacional.
- [5]. M. Solórzano, Ecuaciones para la sintonización de controladores, Ciudad Universitaria Rodrigo Facio, 2007.
- [6]. G. Lemus, Metodología para la implementación de controlador difuso tipo Takagi-Sugeno en PLC s7-300, Medellín: Tecnura, 2011.
- [7]. L. García, Panel gráfico para diagnóstico de fallas en Forja, Santiago de Querétaro, 2011.
- [8]. J. H. Holland, Adaptation in natural and artificial systems, Cambridge, MA: MIT, 1992.
- [9]. J. L. Guzmán, Laboratorio Remoto Para El Control De Una Maqueta De Invernadero, Universidad de Almería. Dpto. de Lenguajes y Computación., 2001.
- [10]. G. E. Monte, Controladores Inteligentes Aplicados A Robótica Móvil, Unidad Académica Confluencia. UTN, 2005.
- [11]. S. Neshmachnow, Algoritmos Genéticos Paralelos Y Su Aplicación Al Diseño De Redes De Comunicaciones Confiables, Montevideo: Pedeciba, 2004.
- [12]. C. Igel, Covariance Matrix Adaptation for Multi-Objective Optimization, Evolutionary Computation Volume 15, Number 1, 2007.
- [13]. N. C. Díaz, Desarrollo de un código de simulación por monte carlo para la calibración de sistemas de detección en espectrometría de radiación gamma, Badajoz, 2009.
- [14]. H. John, Hill-Climbing theories of learning, University of California, USA, 1982.
- [15]. T. Piedad, Introducción a los algoritmos genéticos y sus aplicaciones, Paseo de los Artilleros s/n Madrid 28032, Madrid.