

**DEVELOPMENT OF A TOOL FOR TUNING IN MATLAB PID CONTROLLERS
USING GENETIC ALGORITHMS BASED OPTIMIZATION TECHNIQUES
MULTIOBJECTIVE**

**DESARROLLO DE UNA HERRAMIENTA EN MATLAB PARA
SINTONIZACIÓN DE CONTROLADORES PID UTILIZANDO ALGORITMOS
GENÉTICOS BASADO EN TÉCNICAS DE OPTIMIZACIÓN MULTIOBJETIVO**

**Est. Brayan Rene Acevedo Jaimes*, Est. Juan Camilo Fonseca Galindo*
PhD. Antonio Faustino Muñoz Moner****

***Universidad Francisco de Paula Santander.**

Grupo de Investigación en Automatización y Control, GIAC.
San José de Cúcuta, Norte de Santander, Colombia. Tel.: (+57) 5776655, Ext. 202.
E-mail: payo_aj29@hotmail.com, juankmilofg@gmail.com

****Universidad Autónoma de Bucaramanga - Grupo de Control & Mecatrónica.**
Calle 48, No. 39-234, Bucaramanga, Tel.: (+57) 6436111, Ext. 457.
Email: amunozm@unab.edu.co

Abstract: This document shows the versatility and efficiency by the development of a tool in Matlab for tuning proportional, integral and derivative controller (PID) using genetic algorithms (GA), based techniques for multiobjective optimization (MOP) based on Pareto fronts, calculating optimally constant proportional gain, integral gain and derivative gain (KP, KI, KD) for error minimization, mitigation of the maximum overshoot and settling time reduction in a given plant. Performance that is implementing genetic algorithms provide solutions to multiple targets in PID controllers, the tuning of existing PID controllers SISOTOOL Matlab compares different control systems closed loop formed by a transfer function is simulated, the controller and feedback loop. In these systems the behavior that drivers have to pass on a stair at the entrance to the ground is analyzed.

Keywords: Genetic algorithms, optimization, PID, Matlab, controller tuning.

Resumen: Este trabajo muestra la versatilidad y eficiencia que presenta el desarrollo de una herramienta en Matlab para sintonización de controladores proporcional, integral y derivativo (PID) utilizando algoritmos genéticos (AG) basado en técnicas de optimización multiobjetivo (MOP) fundamentado en frentes de Pareto; calculando de manera óptima las constantes de ganancia proporcional, ganancia integral y ganancia derivativa (KP, KI, KD) para minimización del error, atenuación del sobrepico máximo y reducción del tiempo de establecimiento en una planta determinada. Se compara el desempeño que tiene la implementación de algoritmos genéticos en dar soluciones a múltiples objetivos en controladores PID, con la sintonización de controladores PID existente en Sisotool de Matlab, se simuló diferentes sistemas de control de lazo cerrado conformados por una función de transferencia, su controlador y lazo de realimentación. En estos sistemas se analiza el comportamiento que presentan los controladores al aplicarle un Step a la entrada de la planta.

Palabras clave: Algoritmos genéticos, optimización, PID, Matlab, sintonización.

1. INTRODUCCIÓN

Los controladores PID en la actualidad son una de las soluciones más prácticas, confiables y robustas para el control de procesos industriales; A pesar de que la mayoría de controladores PID funcionan moderadamente bien aplicando reglas de sintonía sencillas, existen procesos que necesitan de un ajuste más preciso para garantizar su óptimo desempeño, de modo que una correcta sintonización de los parámetros de ganancia del controlador conlleva un buen funcionamiento del mismo.

De modo que en este documento se implementa una herramienta que da solución a un problema de optimización multiobjetivo de controladores PID en función de tres acciones básicas de control: acción proporcional, integral y derivativa, representando el modelo de la planta por una función de transferencia, su controlador y lazo de realimentación; teniendo como funciones objetivo el error, sobrepico máximo y tiempo de establecimiento. Para mejorar la optimización de controladores PID y obtener las constantes óptimas de ganancia (KP, KI y KD) del controlador se utilizó algoritmos genéticos, el cual es un algoritmo basado en las leyes de selección natural. Además se maneja un problema de optimización multiobjetivo usando frentes de Pareto para encontrar a los individuos óptimos que satisfagan a las funciones objetivos propuestas.

2. MARCO TEORICO

Uno de los mayores retos de los seres humanos es poder controlar las diferentes tareas que se realizan en la industria. Desde prender y apagar un dispositivo con la ayuda de un botón, hasta controlar aeronaves no tripuladas (J. Camilo y M. Alejandra, 2013). Los controladores han generado muchos beneficios para la apropiada operación de procesos o sistemas; por lo cual son de gran interés, tanto en sus funcionalidades, como en el mejoramiento continuo del desempeño de procesos industriales (M. Claudia, 2006).

2.1 controladores PID

El diseño de controladores se realiza en función del conocimiento del proceso, es decir, a partir del

modelo del proceso, del esquema de control y de las restricciones que se le imponen al mismo. A diferencia de ello la sintonización de los

controladores se realiza sin que se disponga de dicha información; Todo controlador PID está gobernado por los valores de ciertas constantes que ponderan las ganancias en el controlador, estas se denominan: Ganancia Proporcional, Ganancia Integral y Ganancia Derivativa (I. A. Ruge and M. A. Alvis, 2009).

El primer término, Ganancia proporcional KP tendrá el efecto de reducir el tiempo de elevación. Una Ganancia integral KI tendrá el efecto de eliminar el error en estado estacionario, pero puede empeorar la respuesta transitoria. Una Ganancia derivativa KD tiene el efecto de incrementar la estabilidad al sistema, reducir el sobrepico y mejorar la respuesta transitoria.

En la ecuación 1 se aprecia la expresión matemática que modela un controlador PID en el dominio de Laplace (K. Ogata, 1993). En la figura 1 se observa el diagrama de bloques de un control PID a una planta. Como entrada al sistema se tiene la referencia $r(t)$ y como salida del sistema se tiene $y(t)$, el controlador en este caso se denomina $G_{pid}(s)$ y el sistema o planta que se desea controlar se encuentra como $G_p(s)$ (J. Camilo y M. Alejandra, 2013).

$$G_{PID}(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (1)$$

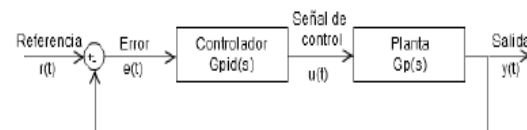


Fig. 1. Diagrama de bloques control PID.

2.2 Algoritmos Genéticos

Los algoritmos genéticos fueron introducidos por John H. Holland (John Holland, 1992) a principios de los 60s y son, por mucho, la técnica evolutiva más popular en la actualidad.

Los algoritmos genéticos resultan ser un tipo de algoritmo matemático altamente paralelo que transforma un conjunto de objetos matemáticos individuales con respecto al tiempo usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto, y tras haberse presentado de forma natural una serie de operaciones genéticas entre las que destaca la recombinación sexual (John Koza, 1992), (G. Toscano, 2001).

La idea fundamental de los algoritmos genéticos consiste en encontrar una solución aceptable a un problema por medio del mejoramiento de un conjunto de individuos, cuya función de evaluación corresponde a una solución del problema. Esta optimización se realiza mediante procesos selectivos y de intercambio de información genética (John Holland, 1992).

Para poder aplicar el algoritmo genético se requiere de los siguientes componentes básicos:

- Una **representación** de las soluciones potenciales del problema. La representación es la cadena binaria para la codificación de las soluciones del problema, a la cadena se le llama **cromosoma**. A cada posición de la cadena se le denomina **gen** y al valor dentro de esta posición se le llama **alelo**.
- Una forma de crear una **población inicial** de posibles soluciones, normalmente un proceso aleatorio.
- Una función de evaluación (**Fitness**) que juegue el papel del ambiente, clasificando las soluciones en términos de su aptitud; esta función puede determinar qué tan buenos o malos serán los resultados y la convergencia o no del método.
- Un mecanismo de **selección** que permita seleccionar a los individuos de acuerdo a su aptitud. En los algoritmos genéticos se puede llevar el proceso de selección de diversas maneras, ya sea determinística o probabilísticamente. Las técnicas de selección pueden ser de selección proporcional (John Holland, 1992); Selección mediante Torneo (A. Wetzel, 1983) o Selección de estado uniforme (Darrel Whitley, 1989).
- Operadores genéticos como **cruza**, **mutación** y **elitismo** que alteren la composición de los hijos que se producirán para las siguientes generaciones (Gunter Rudolph, 1994).

Existen tres tipos principales de cruza; Cruza de un punto, se selecciona un punto de manera aleatoria dentro del cromosoma de cada padre y a partir de este se intercambian los materiales genéticos para dar origen a nuevos individuos (John Holland, 1992); Cruza de dos puntos (A. De Jong, 1975); Cruza uniforme, cruza de n puntos (Gilbert, 1989), (Lawrence, 1991).

2.2.1 Optimización multiobjetivo

El problema de encontrar un vector de variables de decisión que satisfaga las restricciones y optimice una función vectorial cuyos elementos representen las funciones objetivo. Estas funciones forman una descripción matemática de criterios de desempeño que están usualmente en conflicto entre sí. Por lo tanto, el termino optimizar significa encontrar aquella solución que daría un valor aceptable al diseñador en todas las funciones objetivo (A. Osyczka, 1984).

La optimización multiobjetivo maneja los siguientes conceptos:

- **Variables de decisión**, que resultan ser un conjunto de n parámetros cuyos valores dan una solución que puede o no ser válida a un problema de optimización (G. Toscano, 2001).
- **Restricciones**, generalmente en todos los problemas de ingeniería; ellas delimitan el problema y validan las soluciones. Por lo tanto, se puede decir que las restricciones dibujan el contorno de la región donde se encuentra el conjunto factible del problema. Las restricciones son funciones de las variables de decisión y pueden ser tanto de igualdad como de desigualdad (G. Toscano, 2001).
- **Funciones objetivo**, estas forman el criterio de evaluación para saber qué tan buena es una solución; al igual que las restricciones, son funciones de las variables de decisión. En la optimización multiobjetivo existen dos o más funciones objetivos ($f_1(x)$, $f_2(x)$, ..., $f_k(x)$) en cada problema (G. Toscano, 2001).
- **Problema de optimización objetivo**, En estos problemas el objetivo de la optimización es encontrar un vector de decisión o individuo que minimice los resultados evaluados por las funciones objetivos y además cumpla con las restricciones de igualdad o desigualdad impuestas, a esto se le conoce como un conjunto factible en la población (G. Toscano, 2001).
- **Dominancia de Pareto**, Para dos vectores de decisión (x^*) , $(y^*) \in X$ (G. Toscano, 2001).

$\bar{x}^* < \bar{y}^*$ \bar{x}^* domina a \bar{y}^*	Si $f_i(\bar{x}^*) < f_i(\bar{y}^*)$ para toda $i = 1, \dots, k$
$\bar{x}^* \leq \bar{y}^*$ \bar{x}^* domina débilmente a \bar{y}^*	Si $f_i(\bar{x}^*) \leq f_i(\bar{y}^*)$ para toda $i = 1, \dots, k$
$\bar{x}^* \sim \bar{y}^*$ \bar{x}^* es indiferente a \bar{y}^*	Si $f_i(\bar{x}^*) \sim f_i(\bar{y}^*)$ para toda $i = 1, \dots, k$

Es importante notar que aunque la dominancia se da en el espacio de las variables de decisión, la comparación se da en el resultado de la evaluación de las funciones objetivo es decir, en el espacio de las funciones objetivo.

- **Óptimo de Pareto**, Podemos decir que un vector de decisión (x^*) que es miembro del conjunto factible, es óptimo de Pareto si y solo si no existe otro vector de decisión (y^*) que pertenezca al conjunto factible y que lo domine (G. Toscano, 2001).

2.2.2 Estructura general de los AG

En la figura 2 se observa el diagrama básico de un algoritmo genético.

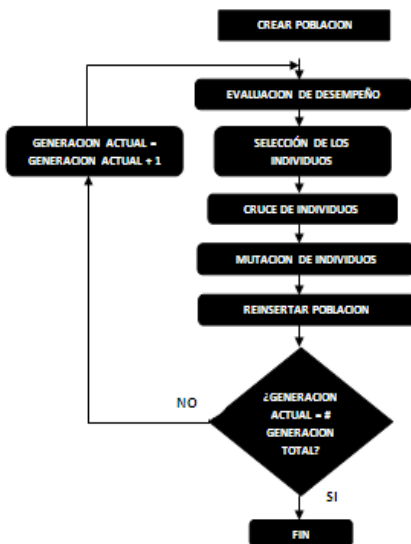


Fig. 2. Diagrama de flujo de algoritmo genético simple.

Se comienza con una población inicial de soluciones aleatorias (población). Cada individuo en la población es llamado cromosoma, el cual representa una solución al problema. Los cromosomas evolucionan a través de iteraciones sucesivas, llamada también generación; durante cada generación los nuevos cromosomas son evaluados, usando la medida de aptitud. Luego, mediante los operadores de cruce y mutación, se seleccionan los cromosomas con mayor

probabilidad de supervivencia, a fin de realizar la exploración y la explotación de las funciones objetivo. Por último, se reinsertan a la población actual, después de terminar el número de generaciones se selecciona al individuo con mayor desempeño, el cual representa la solución más óptima (I. A. Ruge and M. A. Alvis, 2009).

2.3 Aplicación de algoritmos genéticos en controladores PID

En su mayoría, los algoritmos evolutivos han sido empleados para ajustar los parámetros de controladores PID, para mejorar el rendimiento de controladores y así obtener un diseño multidisciplinar, que involucre tanto al diseño de control, como el diseño mecánico y electrónico. A continuación, se muestra en la figura 3 el diagrama de bloques que se tiene para aplicar los algoritmos genéticos al controlador PID.

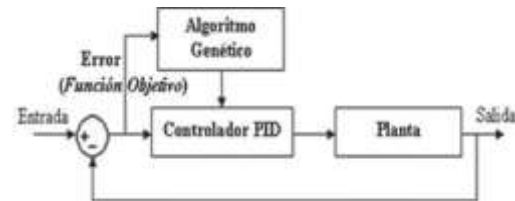


Fig. 3. Algoritmos genéticos aplicados a controladores PID.

3. METODOLOGÍA E IMPLEMENTACIÓN

El algoritmo que se aplicó fue desarrollado e implementado en Matlab 2013 utilizado en un computador portátil Lenovo Z460 con procesador Intel® Core™ i5 M460 a 2,53 GHz con 8 GB de memoria RAM, Windows 8, 64 bits. Se desarrolló una herramienta en Matlab en donde se tomaron como funciones objetivos: el error, el sobrepico máximo y como restricción el tiempo de establecimiento, de manera que el problema de optimización multiobjetivo resuelto se define así:

$$f_1(K_p, K_i, K_d) = \int_0^T [e(t)]^2 dt = \text{Error (ISE)}$$

$$f_2(K_p, K_i, K_d) = M_p = \text{sobrepico máximo}$$

$$R(K_p, K_i, K_d) = t_s = \text{tiempo de establecimiento}$$

Luego del proceso de optimización, al evaluar cada función objetivo en cada iteración la herramienta selecciona y reordena los valores óptimos de las constantes K_p , K_i y K_d para todas las soluciones

obtenidas en cada aumento de generación, de modo que al terminar el número máximo de generaciones tendremos los mejores resultados de las constantes a lo largo de todas las generaciones evaluadas.

Los Algoritmos 1 y 2 muestran el pseudo-código del proceso de optimización multiobjetivo y de la obtención de los valores provenientes de las funciones objetivo, respectivamente.

Algoritmo 1 Pseudo-código general.

1. número de individuos en la población (tamaño de la población)=30.
2. número máximo de iteraciones (número de generaciones)=50.
3. Establecer dimensiones del espacio de búsqueda.
4. Inicializar función de transferencia (numerador y denominador).
5. Definir función para obtener el valor de las funciones objetivo. (Algoritmo 2).
6. Aplicar Algoritmo de Optimización usando Algoritmos Genéticos y optimización multiobjetivo en base a frentes de pareto (script en Matlab).
7. Obtener los mejores resultados.

Algoritmo 2 Pseudo-código para obtener funciones objetivo.

1. Obtener los parámetros de la función de transferencia.
2. Obtener las ganancias para evaluar (KP, KI y KD).
3. Definir la función de transferencia resultante de multiplicar el controlador PID (Ec. 1) y la función de transferencia de la planta.
4. Realizar el lazo de retroalimentación (Feedback).
5. Obtener la respuesta al paso (Step) de la función de transferencia resultante del Feedback.
6. Obtener los valores de error, sobrepico máximo y tiempo de establecimiento con la respuesta al paso unitario.
7. Ordenar del más óptimo al menos óptimo los valores obtenidos por la evaluación.

La población inicial opera con una codificación de tipo numérico real, dando como ventaja la omisión de procesos de codificación y decodificación que son tenidos en cuenta para pasar a los individuos de la población al espacio de evaluación definido por la función objetivo; trabajando así con los coeficientes de las constantes en forma directa. En la figura 4 se muestra el formato de la población inicial.

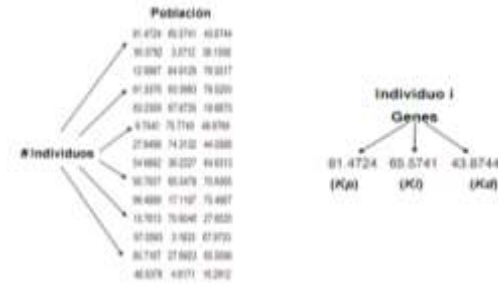


Fig. 4. Población inicial.

En este algoritmo se usó selección por torneo, utilizando dominancia como método de comparación, la idea básica de este tipo de selección consiste en seleccionar con base en comparaciones directas de los individuos de la población. Se escoge un número de individuos para someterlos a competición, generalmente compiten de a dos individuos, después comparamos con base en su aptitud, de modo que el ganador es el individuo más apto de la competencia; así se garantiza que el mejor individuo será seleccionado un mayor número de veces P que el resto de individuos. En la figura 5 se observa el proceso de selección utilizado en la herramienta.

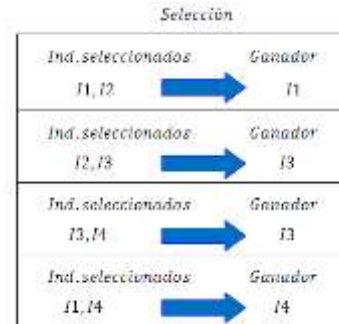


Fig. 5. Selección.

El operador de cruce trabaja sobre dos individuos los cuales son denominados individuos progenitores o padres, este operador produce generalmente otros dos individuos que serán nuevos puntos en el espacio de búsqueda del algoritmo; El operador de cruce implementado en el algoritmo es un cruce de tipo uniforme, y se basa en el uso de una máscara llamada máscara de cruce, la obtención de esta se realiza de forma aleatoria dando la misma probabilidad a que en cada una de las posiciones de la máscara haya un 1 o un 0. En la figura 6 se observa los dos tipos de cruce, cruce de un punto (izquierda) y cruce en dos puntos (derecha).

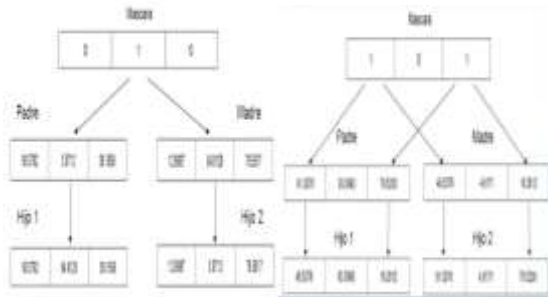


Fig. 6. Cruce uniforme.

Además, se usó una mutación de tipo uniforme la cual nos garantiza un porcentaje de mutación constante a lo largo de todo el proceso evolutivo; en la mutación se habla de un progenitor y un descendiente, además de una máscara aleatoria de tipo binario quien es la encargada de seleccionar cual será en bit del gen correspondiente a mutar; de este modo el operador de mutación ayuda a renovar la población y lo hace de forma arbitraria pues el resultado de la descendencia, nada tiene que ver con su progenitor. En la figura 7 se muestra el proceso de mutación implementado.

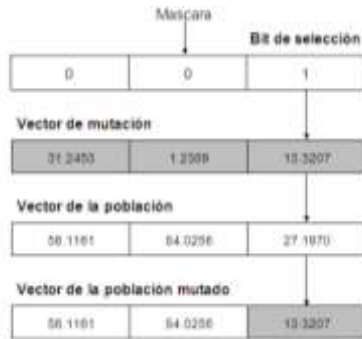


Figura 7. Mutación.

después de haber hecho mutación, se busca obtener un conjunto de individuos aptos para la solución del problema, de este modo la población se somete a un proceso de optimización multiobjetivo basado en frentes de Pareto. El proceso de optimización multiobjetivo se enfoca en la búsqueda de individuos que satisfagan las restricciones impuestas y al mismo tiempo sobresalgan en la evaluación de las funciones objetivo, estos criterios de desempeño por lo general están en conflicto entre sí. En la figura 8 se muestra una gráfica de cómo se encuentran distribuidos los individuos en el frente de Pareto.

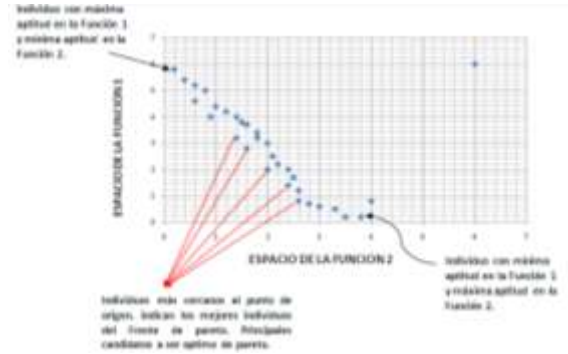


Fig. 8. Frente de Pareto.

4. SIMULACIÓN Y RESULTADOS

Para todas las pruebas desarrolladas, se utilizó una población de 30 individuos y 20 iteraciones, que corresponden en cada caso a 600 evaluaciones de cada una de las funciones objetivo. Para el algoritmo de optimización la población inicial generada es de tipo aleatorio y se operó con un espacio de búsqueda de $0 \leq K_p \leq 100$, $0 \leq K_i \leq 50$ y $0 \leq K_d \leq 50$.

Se realizaron pruebas de la herramienta con plantas extraídas del libro K. Ogata, "Ingeniería de Control Moderna", en cada una de ellas se obtuvieron las gráficas de la planta en lazo cerrado sin controlador, y gráficas de la planta en lazo cerrado con el controlador obtenido de la herramienta PIDag y se compararon resultados con la herramienta Sisotool de Matlab.

4.1 función de prueba 1

$$\frac{5s + 2}{s^3 + 5s^2 + 8s + 4}$$

Proporcional	Integrativo	Derivativo
37.6692	18.9251	13.0447

Tabla: 1. Resultados para la función de prueba

	Sobrepico máximo	Tiempo establecimiento	Error
Planta	0.4506	5.7288	0.67
Algoritmo G.	0.9995	1.14	2.33e-7
Sisotool	1.22	2.5	0.05

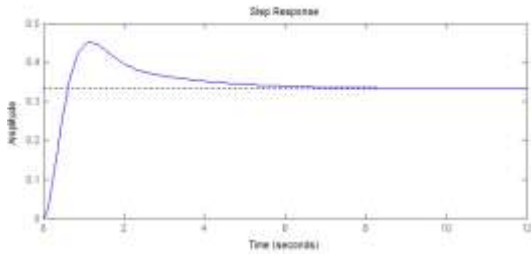


Fig. 9. Respuesta de la planta 1 sin controlador ante un escalón unitario.

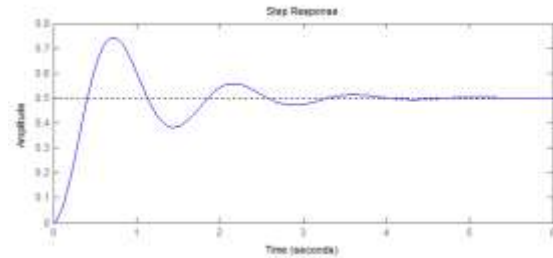


Fig. 12. Respuesta de la planta 2 sin controlador ante un escalón unitario.

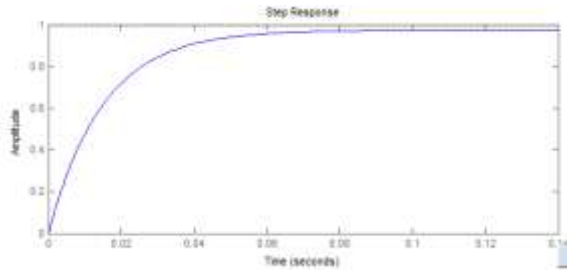


Fig. 10. Respuesta de la planta 1 con controlador sintonizado por PIDag ante un escalón unitario.

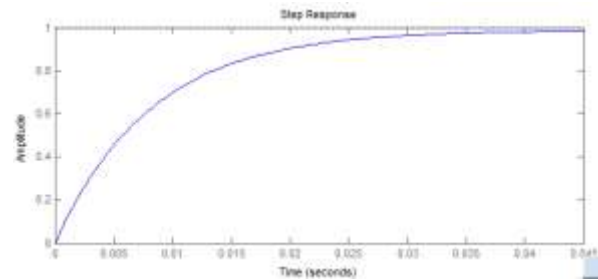


Fig. 13. Respuesta de la planta 2 con controlador sintonizado por PIDag ante un escalón unitario.

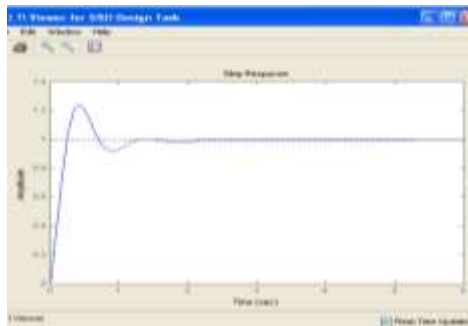


Fig. 11. Respuesta de la planta 1 con controlador sintonizado por Sisotool ante un escalón unitario.

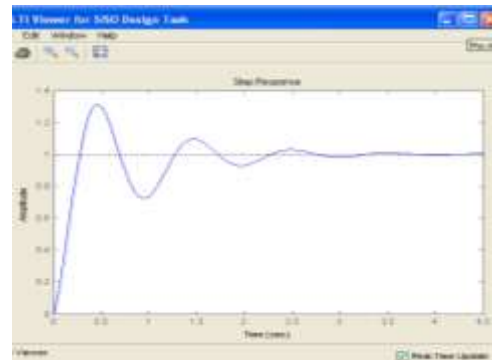


Fig. 14. Respuesta de la planta 2 con controlador sintonizado por Sisotool ante un escalón unitario.

4.2 función de prueba 2

$$\frac{10}{s^2 + 2s + 10}$$

Proporcional	Integrativo	Derivativo
8.4047	17.1467	12.0918

Tabla: 2. Resultados para la función de prueba.

	Sobrepico máximo	Tiempo establecimiento	Error
Planta	0.7421	3.7795	0.5
Algoritmo G.	1.0137	1.9484	3.013e-7
Sisotool	1.32	4	0.004

5. CONCLUSIONES

Se analizó la sintonización de los controladores correspondientes a cada una de las plantas simuladas tomando como criterios de evaluación el error, el sobrepico máximo y tiempo de establecimiento obtenidos en la señal de salida como resultados a la aplicación de un escalón unitario (step) a la entrada de cada una de las plantas, cuando se utiliza la herramienta PIDag, con el fin de minimizar la magnitud del error, reducir el sobrepico máximo y obtener tiempos de establecimientos más cortos, las tablas comparativas y las graficas de las señales tomadas a la salida de la planta, mostraron que la herramienta PIDag en comparación con el Sisotool

de Matlab obtiene mejores resultados en los tres criterios de evaluación tomados para la sintonización de controladores PID.

Se demostró que la aplicación de un método que se sustenta en principios de selección natural para la optimización de los valores de ganancia de un controlador PID, permite dar a conocer una nueva forma de solución no convencional a problemas de control; además de encontrar una variedad de valores óptimos para estos parámetros, dando como opción al programador de elegir el que más se acomode a sus requerimientos de diseño.

Se evidencio que la eficacia de la implementación de Algoritmo Genéticos aplicados a la sintonización de controladores PID radica en la capacidad de exploración activa que tenga el algoritmo, la manera en que seleccionan los individuos de la población, tener una cuidadosa selección de los individuos a recombinar y mutar, además de una buena interpretación de las funciones objetivos.

Al desarrollar una herramienta de sintonización de controladores PID usando algoritmos genéticos en donde se toman más de un criterio de evaluación es imperativo y necesario implementar un proceso de optimización multiobjetivo fundamentado en frentes de Pareto, en donde se demuestra que la asignación de restricciones y funciones objetivos permiten garantizar una mejor exploración del espacio de búsqueda y escoger los individuos mas óptimos, que satisfagan las restricciones impuestas y al mismo tiempo sobresalgan en la evaluación de las funciones objetivos en un espacio de búsqueda global sin que el algoritmo converja a un óptimo local.

REFERENCIAS

- J. Camilo Castro P. and M. Alejandra Guzmán P., (2013). "Optimización Multi-objetivo de un Controlador PID Aplicando Algoritmos Bio-inspirados", CIIMA, vol. 2, No.20, pp 191-197.
- M. Claudia Pinto F., (2006). "Sintonización de controladores PID utilizando algoritmos evolutivos", Universidad de Pamplona, facultad de Ingenierías y Arquitectura, Pamplona Colombia, mayo.
- I. A. Ruge and M. A. Alvis, (2009). "Aplicación de los algoritmos genéticos para el diseño de un controlador PID adaptativo", Tecnura, Vol. 13, No 23, pp 82-90.
- K. Ogata, (1993). "Ingeniería de Control Moderna", Segunda Edición, Traducido por: B. A. Fabian-Kernel, Ed. Prentice-Hall, México.
- DiStefano, Stubberud and Williams, (1992). "Retroatimentación y Sistemas de Control", Segunda Edición, Traducido por: R. Gomez Cruz, Ed. Mc. Graw Hill, Bogotá.
- John H. Holland, (1992). "Adaptation in Natural Artificial Systems." MIT Press, Cambridge, Massachusetts, second edition.
- John R. Koza. "Genetic Programming: On the Programming of Computers by Means of Natural Selection." The MIT Press, Cambridge, Massachusetts, 1992.
- A. Wetzel, (1983). "Evaluation of Effectiveness of genetic algorithms in combinational optimization.", University of Pittsburgh, Pittsburgh (unpublished).
- Darrel Whitley, (1989). "The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best. In Proceedings of the Third International Conference on Genetic Algorithms", pages 116-121, San Mateo, California, July, Morgan Kaufmann Publishers.
- Gunter Rudolph, (1994). "Convergence analysis of canonical genetic algorithms." IEEE Transactions on Neural Networks, 5(1):96-101, January.
- A. K. De Jong, (1975). "An Analysis of the Behavior of a Class of Genetic Adaptive Systems." PhD thesis, University of Michigan, Michigan.
- Gilbert Syswerda, (1989). "Uniform Crossover in Genetic Algorithms." In J. David Schaffer, editor, Proceedings of the Third International Conference on Genetic Algorithms, pages 2-9, San Mateo, California, Morgan Kaufmann.
- Lawrence Davis, (1991). "Handbook of Genetic Algorithms." Van Nostrand Reinhold, New York.
- Van Rensburg, P.J., Shaw, LS. y Van Wyk J. D., (1998). "Adaptive PID-control using a Genetic Algorithm". 1998 Second International Conference on Knowledge-Based Intelligent Electronic Systems." Editors, L.C. Jain and R.K. Jain. Adelaide, Australia. 21 - 23 April.
- K. Deb et al. (2002), "A fast an elitist Multiobjective Genetic Algorithm: NSGA II", IEEE Transaction on Evolutionary Computation, Vol. 6, No 2.
- G. Toscano P., (2001). "Optimización Multiobjetivo usando un Micro Algoritmo Genético", Universidad Veracruzana - Lania, Septiembre.

- A. Osyczka, (1984). Multicriterion Optimization in engineering with FORTRAN programs. Ellis Horwood Limited.
- A.F. Muñoz M., (2005). WSEAS, Advance Control of Applied Artificial Cloning. WSEAS, Transactions On Systems, Vol. 4, Issue 7, July.

- A.F. Muñoz M., (2003). Tecnologías de control inteligente: Redes neuronales, algoritmos genéticos y de clonación artificial. Vol 1, No. 1.