

# Algoritmo genético para sintonización de pid basado en la integral del error absoluto

## *Genetic Algorithm for Tuning PID Integral Based on the absolute error*

Carlos Pillajo <sup>1</sup>, Paul Bonilla <sup>1</sup>, Roberto Hincapié<sup>2</sup>

<sup>1</sup> Carrera de Electrónica

Universidad Politécnica Salesiana, [cpillajo@ups.edu.ec](mailto:cpillajo@ups.edu.ec), [pbonilla@est.ups.edu.ec](mailto:pbonilla@est.ups.edu.ec)

<sup>2</sup> Carrera de Telecomunicaciones

Universidad Pontificia Bolivariana, [Roberto.hincapie@upb.edu.co](mailto:Roberto.hincapie@upb.edu.co)

Recibido: 01-07-2016. Aceptado después de revisión: 28-08-2016

---

**Resumen:** A pesar de la rápida evolución en los sistemas de control de procesos, el controlador PID (Proporcional-Integral-Derivativo) es el más utilizado en la industria, debido a que sus funciones simples y robustas son suficientes para trabajar en una gran variedad de procesos. Sin embargo, resulta complejo sintonizar de manera adecuada las ganancias del PID, debido a que los procesos no son sistemas lineales, ni estacionarios, de modo que sus características cambian con el transcurso del tiempo. El proceso de sintonización no es trivial, ya que en gran medida depende del proceso a ser controlado, y no existe un método que sea apropiado en todas las aplicaciones. Ante esto surgen los sistemas de control adaptable, donde el controlador realiza una auto-sintonización. Se han diseñado estrategias muy sofisticadas alrededor del PID clásico, entre estas se cuentan las técnicas de inteligencia artificial. En este trabajo se diseña un algoritmo genético capaz de sintonizar controladores PID basado en el criterio de la integral del error absoluto (IAE), dicho algoritmo se simulara y se comparara con herramientas de matlab para tal efecto.

**Palabras claves:** Algoritmo Genético, Integral del error absoluto, Sintonización PID, PID

**Abstract:** Despite the rapid development in systems of process control, the PID controller (Proportional-Integral-Derivative) is the most widely used in the industry, because its simple and robust features are enough to work in a variety of processes. However, it is properly tuned complex gains PID, because the processes are nonlinear systems, or stationary, so that its characteristics change over time. The tuning process is not trivial, since it largely depends on the process to be controlled, and there is no method that is appropriate for all applications. Given this adaptive control systems, where the controller performs a self-tuning arise. Are designed very sophisticated about classical PID strategies, these techniques have artificial intelligence. In this paper a genetic algorithm capable of tuning PID controllers based on the criterion of the integral absolute error (IAE), the algorithm is simulated and compared with matlab tools designed for this purpose..

**Keywords:** Genetic Algorithm, integral absolute error, PID tuning, PID.

## 1. Introducción

La variedad de procesos y sistemas que se desean controlar es muy basta, sería muy ambicioso pretender crear un controlador que presente un desempeño aceptable en todos los casos. Sin embargo, en la mayoría de los casos, tal como se menciona en (Aström and Hägglund, 2001) la retroalimentación es un concepto que por sí mismo ha revolucionado la manera en cómo se realiza el control. Y a pesar de que gran parte de esta revolución se debe a la sola idea de la retroalimentación, en ese mismo artículo también se señala que el controlador PID es por mucho la forma más utilizada de control por retroalimentación. Esto se debe a las características que este tipo de controlador presenta. El controlador PID quizá no es el más adecuado para algún proceso en particular, sin embargo, sí es el controlador que prefieren las industrias[1],[2],[3].

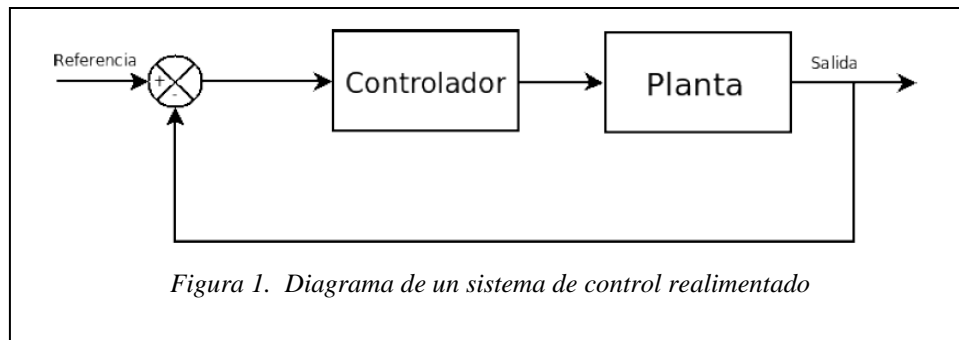
Uno de los aspectos más importantes a considerar del controlador PID es la estructura del mismo, la cual tiene que ver con cómo es que se procesa el error, de antemano se sabe que todo control PID debe ser capaz de manejar las ganancias proporcional, derivativa e integral, sin embargo, existen diversas maneras de hacerlo, es decir, existen diversas estructuras de PID, que se pueden utilizar y que cumplen con estas características.

Sin importar cual estructura se haya elegido para el controlador, se debe considerar que éste debe ser sintonizado, es decir, se debe elegir valores para cada una de las ganancias del controlador, esto es lo que se conoce como sintonización. Este proceso de sintonización es necesario que se realice con cada nueva planta o cuando el desempeño no es satisfactorio, esta característica y necesidad del controlador es lo que lo hace tan versátil. Para resolver el problema de la sintonización de procesos o sistemas que son muy cambiantes, se han propuesto diversas técnicas. Actualmente se estudian de manera seria las técnicas adaptables dentro de éstas, existe un tipo de ellas, las basadas en heurística, las cuales según se ha estudiado presentan resultados satisfactorios y un alto grado de importancia, sin embargo, tienen problemas debido a la cantidad de cálculos que se requiere realizar para llevar a cabo este tipo de métodos. Los algoritmos genéticos son una de dichas técnicas heurísticas. La idea general de esta técnica es encontrar una solución a un problema a través de la optimización de un criterio específico, esto se hace utilizando un espacio de búsqueda que se define en un principio y que va evolucionando conforme el algoritmo avanza. Esta idea de evolución toma conceptos de la misma evolución de las especies en la naturaleza. Es importante remarcar que esta técnica presupone que no se tiene la menor idea de cual podrá ser la mejor solución al problema al iniciar el algoritmo[4].

Así pues, si se tiene un sistema o proceso que cambia con el tiempo y un controlador PID al cual se le deben sintonizar sus ganancias para ser eficiente ante las nuevas condiciones del problema, entonces es posible aplicar un algoritmo genético que se encargue de la sintonización (adaptación) de dichas ganancias, es decir, que sintonice al PID. Ya se han reportado artículos donde se utiliza un algoritmo genético para la sintonización de las ganancias de controladores de tipo PID. Los resultados obtenidos en la mayoría de las ocasiones han sido satisfactorios o al menos prometedores. Es importante remarcar que a pesar de que los resultados son satisfactorios, la mayoría de ellos provienen de simulaciones. Uno de los aspectos que de alguna manera son complejos en el controlador PID es la sintonización del mismo, es decir, cuánto deben valer cada una de las ganancias (proporcional, derivativa e integral) para que el controlador tenga el desempeño esperado.

### 1.1 El controlador PID

Cuando se habla de control automático es casi imposible no pensar en la idea de contar con la retroalimentación.



En dicho esquema se tiene que la entrada al controlador es una señal que se conoce como de error, esta señal es la diferencia que existe entre la referencia y la señal medida a la salida de la planta, esta señal ayuda a que de alguna manera el sistema sea consciente de la respuesta, es decir, que si hay discrepancia entre lo deseado y lo obtenido de manera real, el controlador trate de cambiar esta circunstancia.[5]

A pesar de que elegir un controlador u otro no es una decisión trivial y evidentemente no tiene una respuesta única, existe un controlador que se utiliza ampliamente en el control automático, a saber, el controlador PID, por sus siglas en inglés (Proportional Integral Derivative). Este controlador utiliza la señal de error del sistema y la procesa de tres maneras diferentes y muy útiles además de intuitivas, dichas maneras son: lo amplifica o atenúa, lo deriva y lo integra, esto es muy abstracto y matemático, sin embargo si se trata de interpretar el significado de cada una de estas acciones realizadas se puede decir que el estado presente se ve afectado por la parte amplificadora, de alguna manera se tiene idea del pasado, esto mediante la acción integral e incluso se puede decir que de alguna forma se puede predecir lo que sucederá, esto es gracias a la acción derivativa del control.

De manera formal el controlador PID se puede expresar como:

$$g_c(t) = K_p e(t) + K_d \frac{de(t)}{dt} + K_i \int e(t) dt \quad \text{Ecuación (1)}$$

esto es en su representación temporal y tomando la señal del error como  $e(t)$ , sin embargo, en control es muy común utilizar la representación en  $s$ , es decir, utilizando la transformada de Laplace, esta representación es como sigue:

$$G_c(s) = K_p E(s) + K_d s E(s) + \frac{K_i E(s)}{s} = (K_p + s K_d + \frac{K_i}{s}) E(s) \quad \text{Ecuación (2)}$$

### 1.2 Métodos de sintonización

Seleccionar estas ganancias no es trivial, ya que se puede decir que esto hace que para cada uno de las plantas distintas el PID se pueda utilizar de forma satisfactoria. Así pues, seleccionar las ganancias o sintonizar, como comúnmente se le conoce, es una tarea fundamental para el buen desempeño del controlador PID.

#### 1.2.1 Métodos Clásicos

*Ziegler-Nichols.*- Son dos los métodos propuestos por Ziegler-Nichols y se utilizan de manera muy extensa, ya sea en su forma original o con algunas modificaciones. Estos métodos se basan en la determinación de algunas características de la dinámica de la planta a controlar y posteriormente los parámetros del controlador se expresan en

fórmulas que relacionan los parámetros del controlador con los obtenidos de manera experimental.

*Otros.-* Existe varios métodos distintos al método de propuesto por Ziegler-Nichols, sin embargo, utilizando éste y sus derivados se obtienen resultados aceptables en la mayoría de los casos. A continuación se enunciará brevemente otros métodos que no son derivados del propuesto por Ziegler y Nichols: [6]

Primero se encuentran los métodos en los cuales se trata de dar forma al lazo, es decir, se trata de cambiar el valor de la función de transferencia en un punto o serie de puntos mediante los valores otorgados a las ganancias del controlador.

También existen métodos, en los cuales se encuentra el modelo de la planta y mediante procedimientos analíticos se trata de dar valores a los parámetros del PID para que cuando se combine en cascada con la planta y se realice la retroalimentación, el sistema tenga los polos y ceros de tal forma que el resultado sea el deseado, un ejemplo de estos métodos es de Haalman[3].

Además de los métodos mencionados con anterioridad existen los que se basan en la optimización. Dichos métodos tratan de optimizar el criterio de diseño más importante, esto lo logra generando una o varias funciones o desigualdades, en las cuales involucren los parámetros del PID y el criterio de mayor interés en el diseño.

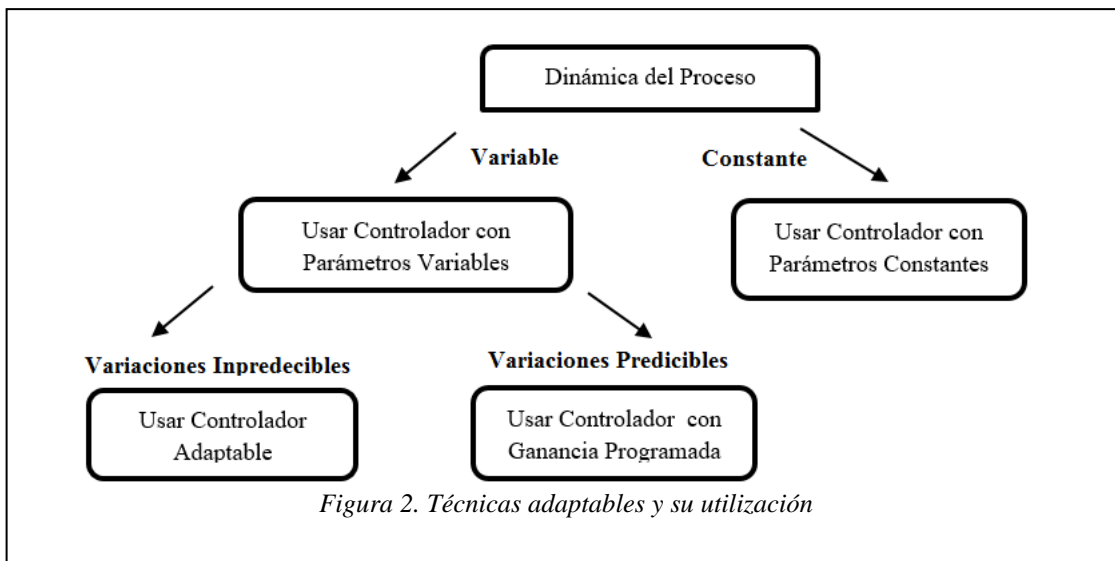
### 1.2.2. Métodos Adaptables.

Todos los métodos anteriores de forma tácita utilizan el hecho de que la planta o su modelo, bajo ninguna circunstancia, cambiarán con el tiempo, lo cual en la realidad se sabe que no es verdad. Tan sólo es simple hecho de que todo los elementos constitutivos de un sistema tienen un desgaste y tiempo de vida hacen que los parámetros de la planta cambien y por lo tanto la respuesta que pueden tener en un determinado momento a una cierta entrada no sería la misma que presentarán a la misma entrada en otro momento. Bajo estas condiciones los métodos presentados con anterioridad serían válidos mientras el proceso o la planta no cambien, esto es un problema más que se tiene que resolver[7]. En la Figura 2 se muestra que cuando se tiene una planta con dinámica variables es conveniente utilizar un controlador con parámetros variables, y si este es el caso, se tienen dos posibilidades, la primera es que dichas variaciones de alguna manera se puede predecir, en tal situación lo más conveniente es utilizar un método de asignación de ganancias, pero si no se pueden predecir las variaciones, entonces, es más conveniente utilizar un controlador adaptable. Así pues si se piensa en el peor de los casos, se tendría que la planta tiene una dinámica variable y que dichas variaciones no se pueden predecir. Lo primero es que se debe delimitar la definición de sistema adaptable al campo del control, donde se dice que un sistema de control adaptable es un sistema que tiene la capacidad de adaptar los parámetros o la estructura de una parte del sistema de control, a saber, el controlador, esto cada que existan cambios en los parámetros o la estructura de la planta, dichos cambios se deben dar de tal forma que el sistema en conjunto mantenga un comportamiento óptimo de acuerdo a un criterio seleccionado. En la misma referencia se señala que la adaptación a los cambios que se puedan dar en la planta puede ocurrir principalmente de tres maneras:

- Realizando un cambio adecuado en los parámetros del controlador
- Alterando la estructura del controlador
- Generando una señal auxiliar de entrada

Es importante señalar que la retroalimentación de alguna manera compensa los cambios que se puedan presentar, sin embargo por sí sola no se considera como un sistema de

control adaptable, ya que la manera en la cual procesa el error es fija, es decir, no se adapta a los cambios de manera dinámica[2].



## 2. Sintonización de controladores PID utilizando Algoritmos Genéticos

En el trabajo que presenta (Yasue et al., 1999) se pone de manifiesto la posibilidad de utilizar los algoritmos genéticos como una herramienta para la sintonización de un controlador PID. En este trabajo se propone un esquema de control PID basado en algoritmos genéticos y leyes de control de varianza mínima generalizada. El propósito de este trabajo es utilizar este esquema de control en sistemas con parámetros desconocidos o variables. Al final de este trabajo se presentan simulaciones de control con diversos ejemplos de plantas en donde se aplicará los controladores PID autoajustables, y la que se propuso en ese trabajo resultó ser la de mejor desempeño.

A pesar de que los algoritmos genéticos son una herramienta poderosa, innovadora y atractiva para resolver problemas, en los trabajos previamente mencionados en esta sección, se analizan desde un punto de vista más teórico y a nivel de simulación. Algunos se implementan usando una computadora personal como plataforma de desarrollo, quizá un sistema embebido sería una opción más adecuada como plataforma de implementación.

### 2.1. Metodología

En primera instancia se presenta el controlador que se utilizará, es decir, la estructura del controlador PID que se utiliza. Posteriormente se explica en que consiste un algoritmo genético, cuáles son sus operadores y finalmente utilizando estos conceptos se desarrolla el algoritmo genético que para la sintonización del PID.

#### 2.1.1 PID Digital

El hecho de que se utilice una implementación de naturaleza digital implica que se deben tomar en cuenta diversos aspectos que más que ligados al concepto del controlador PID, se encuentran ligados a la naturaleza digital. Entre estos aspectos se encuentran el tiempo de muestreo, filtrado anti-alias, desratización y efectos de cuantización, entre las versiones digitales del controlador PID se encuentran la llamada de posición y la incremental o de velocidad. A continuación se presentan cada una de ellas

Forma posicional:

$$u(t) = K_p \left[ e(t) + \frac{T_c}{T_i} \sum_{i=0}^t e(i) + \frac{T_d}{T_c} (e(t) - e(t-1)) \right] \quad \text{Ecuación (3)}$$

Forma de velocidad:

$$\Delta u(t) = (u(t) - u(t-1)) \quad \text{Ecuación (4)}$$

$$u(t) = u(t-1) + \Delta u(t) \quad \text{Ecuación (5)}$$

$$\Delta u(t) = K_p \left[ (e(t) - e(t-1)) + \frac{T_c}{T_i} e(t) + \frac{T_d}{T_c} (e(t) - 2e(t-1) + e(t-2)) \right] \quad \text{Ecuación (6)}$$

## 2.2. Algoritmos Genéticos

La computación evolutiva se inspira en el principio de selección natural para resolver problemas de búsqueda y optimización. Entre las principales técnicas de computación evolutiva se encuentran los algoritmos genéticos propuestos por Holland, la programación genética, estrategias evolutivas presentadas por Recheberg y la programación evolutiva a cargo de Fogel. Todos estos paradigmas utilizan el concepto de selección natural, sin embargo, cada uno de ellos los aplica de manera distinta.

De forma general la computación evolutiva ofrece ventajas en muchos problemas de optimización. Algunas de las ventajas más notorias son: simplicidad de aproximación al problema, respuesta robusta ante cambio de circunstancias en el problema, paralelismo, posibilidad de realizar un híbrido entre técnicas evolutivas y técnicas convencionales o incluso con otras técnicas de la inteligencia artificial como las redes neuronales y el control difuso u otros y quizá como se señala en (S.N.Sivanandam and S.N.Deepa, 2008) la característica que distingue a este tipo de técnicas es que ayuda a resolver el problema de cómo resolver los problemas[8].

Cabe señalar que de todas estas alternativas la más representativa es la de los algoritmos genéticos. Los cuales son una herramienta que pertenece a la rama de la inteligencia artificial, que probado ser útil en la solución de problemas donde se requiere optimizar una función. Desde el punto de vista del control se puede pensar en esta optimización como una minimización de una función, dicha función por lo regular es el error entre la referencia del sistema y la salida del mismo.

### 2.2.1. Etapas del algoritmo genético.

*Codificación* .- Antes de comenzar a describir cada uno de los operadores de los algoritmos genéticos es pertinente mencionar que cada individuo de la población puede representarse o codificarse de dos maneras distintas, una es a través de cadenas de bits y la otra es mediante números reales. Históricamente primero se utilizaron las cadenas de bits como la codificación a elegir, sin embargo, se ha probado que la codificación a través de número reales es la más eficiente[9].

*Creación de la primera generación*.- Este el primer paso de un algoritmo genético, el cual tiene que ver con el espacio de búsqueda de la solución. Esta definición del espacio de búsqueda debe ser aleatoria. Cada elemento de esta generación se conoce como un individuo. Cada uno de estos individuos es un candidato a la solución más apta para el problema. Existen dos cuestiones importantes a la hora de crear la primera generación, la primera es que tan aleatoria va a ser la creación y la segunda es el tamaño de la población.

*Función de aptitud.*- Para saber si un elemento de la generación actual cumple con los requerimientos para considerarse la solución buscada se debe contar con un mecanismo, el cual se conoce como función de costo o aptitud. Así pues, cada individuo se somete a la función de aptitud y con ello se logra saber si es un buen candidato para participar en la procreación de nuevas posibles soluciones o incluso se puede dar el caso, que es lo que se busca al final, de que sea tan apto el individuo que sea la solución buscada.

La función de aptitud no es trivial, de hecho es un punto crítico para el éxito que puede alcanzar el algoritmo genético, es decir, cada problema debe definir una función de aptitud que verdaderamente ponga de manifiesto la aptitud del individuo en cuestión para el problema que se desea resolver.

*Selección.*- Una vez que se han asignado aptitudes a cada uno de los individuos, se procede a la selección de los más aptos. Dicha selección se puede llevar a cabo de diversas formas una de ellas es seleccionar a la mitad de la generación que haya mostrado ser la más apta individuo a individuo, a esta forma se le conoce como elitismo. A pesar de que esto parece ser a primera vista lo más adecuado, algunas veces acarrea problemas de convergencia prematura, por lo cual se utilizan métodos alternativos como la ruleta de selección o selección aleatoria.

*Creación de la siguiente generación y criterio de finalización.*- Antes de crear la nueva generación, primero se revisa si es necesaria dicha creación, es decir, se evalúa el criterio de finalización del algoritmo, en el mejor de los casos, uno de los individuos cuenta con las características pertinentes para ser la solución que se desea, con lo cual el algoritmo termina. A pesar de que es muy posible que se encuentre un individuo que cumpla con los requerimientos necesarios para ser la solución, también existe la posibilidad de que no se encuentre este individuo, así que en muchas ocasiones se plantea como criterio de finalización del algoritmo un cierto número de generaciones, así pues, si se encuentra el individuo solución o se alcanza el límite de generaciones el algoritmo ha finalizado, de otra manera se prosigue con los operadores que se describirán a continuación.

*Recombinación.*- Si es necesario crear una nueva generación, se cuentan con dos operadores principales, los cuales son el de recombinación y el de mutación.

El primero se hace referencia a como a partir de soluciones aptas se pueden generar soluciones más aptas, o al menos en teoría eso es lo que se busca.

Existe varias aproximaciones para este operador, una de las más utilizadas es utilizar lo que se conoce como dos padres, individuos de la generación anterior, y a partir de ellos crear dos hijos, esa creación de hijos es lo que se conoce como recombinación.

Utilizando codificación en números reales el operador de recombinación que se utiliza en este trabajo es el que se conoce como operador de recombinación aritmética que es básicamente una combinación lineal de los padres, y tiene la forma que sigue.

$$h_1 = rP_1 + (1 - r)P_2 \quad \text{Ecuación (7)}$$

$$h_2 = (1 - r)P_1 + rP_2 \quad \text{Ecuación (8)}$$

Donde  $P_x$  son los padres y  $h_x$  son los hijos generados.

*Mutación.*- Este operador es opcional, pero en muchos casos es muy deseable contar con él, ya que en algunos casos permite que no se presente una convergencia prematura.

### **2.3. Algoritmo genético para la sintonización del controlador PID digital**

En las secciones previas se han expuesto el controlador PID digital y los algoritmos genéticos respectivamente, en esta sección se retoman los conceptos de esas secciones para en base a ellos poder desarrollar un algoritmo genético dedicado a la sintonización de un PID[10].

*Consideraciones para el desarrollo del algoritmo genético.*- Ya teniendo en mente que el objetivo del algoritmo genético es sintonizar un controlador PID es conveniente que desde un principio se tome en cuenta que la plataforma elegida es digital. A pesar de que desde un principio sería posible tomar en cuenta algunas restricciones más impuestas por la plataforma final, como por ejemplo el tipo de aritmética que se utilizará o la cantidad de recursos (memoria) con de las cuales se dispondrá para la implementación final, esto no se realiza así en primera instancia, esto con la finalidad de aprovechar la oportunidad para realizar la comparación en desempeño del algoritmo genético implementado en software y con su contraparte en hardware.

*Representación y codificación de cada individuo.*- Debido a que sin importar la estructura del PID que se elija, incluso sin tomar en cuenta si fuese analógico o digital, lo que no varía es que se compone de tres ganancias las cuales son proporcional ( $K_p$ ), derivativa ( $K_d$ ) y la integral ( $K_i$ ). Es por eso que se la codificación de los individuo se deben tomar en cuenta las tres. Así pues cada individuo tendrá la forma de  $[K_p K_i K_d]$

Ahora gracias a que la plataforma final de implementación es digital suena tentador que la codificación del individuo se lleve a cabo por cadenas de bits en vez de que sea de forma real, esto es debido a que las operaciones con cadenas de bits son más sencillas, sin embargo en este caso se optó por realizar una codificación real, ya que no se contará con la limitaciones fuertes para realizar operaciones aritméticas y además que los resultados en trabajos relacionados sugieres que los resultados alcanzados de esta manera son mejores.

*Creación de la primera generación .*- Es importante señalar que la creación de la primera generación de alguna manera establece el espacio de búsqueda, claro que dependiendo del algoritmo genético en cuestión y de la manera en que evolucione, las soluciones pueden, y en algunos casos es deseable, que vayan más allá de este espacio de búsqueda. También se debe mencionar que por lo regular esta primer generación se lleva a cabo utilizando métodos aleatorios o pseudo-aleatorios, sin embargo si te tiene una idea aunque sea somera de por donde puede estar la solución, es conveniente que el espacio de búsqueda se centre en ese lugar.

Así pues en este caso se utiliza un método pseudo-aleatorio para la creación de la primera generación y además se cuenta con un mecanismo para poder delimitar en que rango se generen los individuos.

*Función de aptitud.*- Este es un punto muy crítico en la adaptación del algoritmo genético. De manera común se prueban los algoritmos genéticos con funciones matemáticas conocidas y bien estudiadas, en esos casos la elección de la función de aptitud no sólo no es retardadora sino que es directa, es decir, se utiliza la función matemática como función de aptitud o costo, sin embargo en este caso debido a que lo que se requiere es minimizar el error de la referencia con respecto a la salida pues la elección se complica bastante ya que la respuesta del sistema y por ende el error no solo depende del controlador, el cual es bien conocido, sino también de la planta en cuestión. En este caso se optó por tomar la función de aptitud como la integral del valor absoluto del error (IAE), esto es cuando se



pone el controlador PID en cascada con la planta y en un esquema de retroalimentación típico

*Selección.*- Para el operador de selección se optó por utilizar elitismo, el cual consiste en escoger a los individuos más aptos, en este caso se selecciona a la mitad de la población más apta. Para esto se ordena los individuos según su aptitud y se procede a tomar los más aptos.

*Recombinación y Mutación.*- El operador de recombinación que se utilizó es una extrapolación y el de mutación se utilizó el de no uniformidad.

*Siguiente generación y criterio de finalización.*- El criterio de finalización que se propone es por número de generaciones, es decir, que al cabo de un cierto número de generaciones presumiblemente se tiene a la mejor solución, en este caso, las ganancias adecuadas para el controlador, y claro que mientras no se alcance este número de generaciones lo que sucede es que la nueva generación está conformada por los padres de esta generación y los hijos de estos.

#### *Versión en software del algoritmo genético para sintonizar controladores PID*

En primera instancia se debe mencionar que con propósitos de experimentación y validación del algoritmo, en un principio se utilizó un lenguaje de scripts, Matlab, para el desarrollo de la versión en software. Esto tiene varias implicaciones. Primero como matlab estar orientado a matrices y tiene muchas funciones disponibles, las cuales en ningún momento se dudó utilizarse, el desarrollo de las versiones preliminares y primer versión fue más sencillo. Segundo, precisamente el hecho de que tenga tantas funciones disponibles permitió probar diferentes ideas de manera rápida para ver cuál daba mejor resultado. Y tercero, la velocidad de ejecución es más lenta que si se hubiese optado por otro lenguaje de programación.

### **3. Resultados**

En este trabajo se presentó un algoritmo heurístico que sintoniza un controlador PID para el control de plantas cuya función de transferencia es de la forma:

$$G(s) = \frac{Kp}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad \text{Ecuación (9)}$$

El Algoritmo Genético (AG) este algoritmo presenta un desempeño satisfactorio para sintonizar controladores PID en línea, además de encontrar soluciones satisfactorias estas también encuentran la solución en pocas interacciones. Por otra parte, el algoritmo Genético es flexible debido a que pueden adaptarse a diversos problemas de control.

Se simuló la sintonización del PID con varios ejercicios, dando resultados convincentes, a continuación se analiza los resultados para el ejercicio N1.

Ejercicio N1: Dada una planta de segundo orden sin tiempo muerto (ecuación 10), determinar los parámetros de un controlador PID para que la respuesta del sistema tenga un Mp no mayor al 15% y un Ta 2% de 5 segundos.

$$G(s) = \frac{1.5}{(10s+1) \cdot (5s+1)} \quad \text{Ecuación (10)}$$

Donde:  $\tau_{o1}=10$  ;  $\tau_{o2}=5$  ;  $a=0.5$  ;  $b= 2$  ;  $K_p=1.5$  ;  $z=0.8$  ;  $w=9$

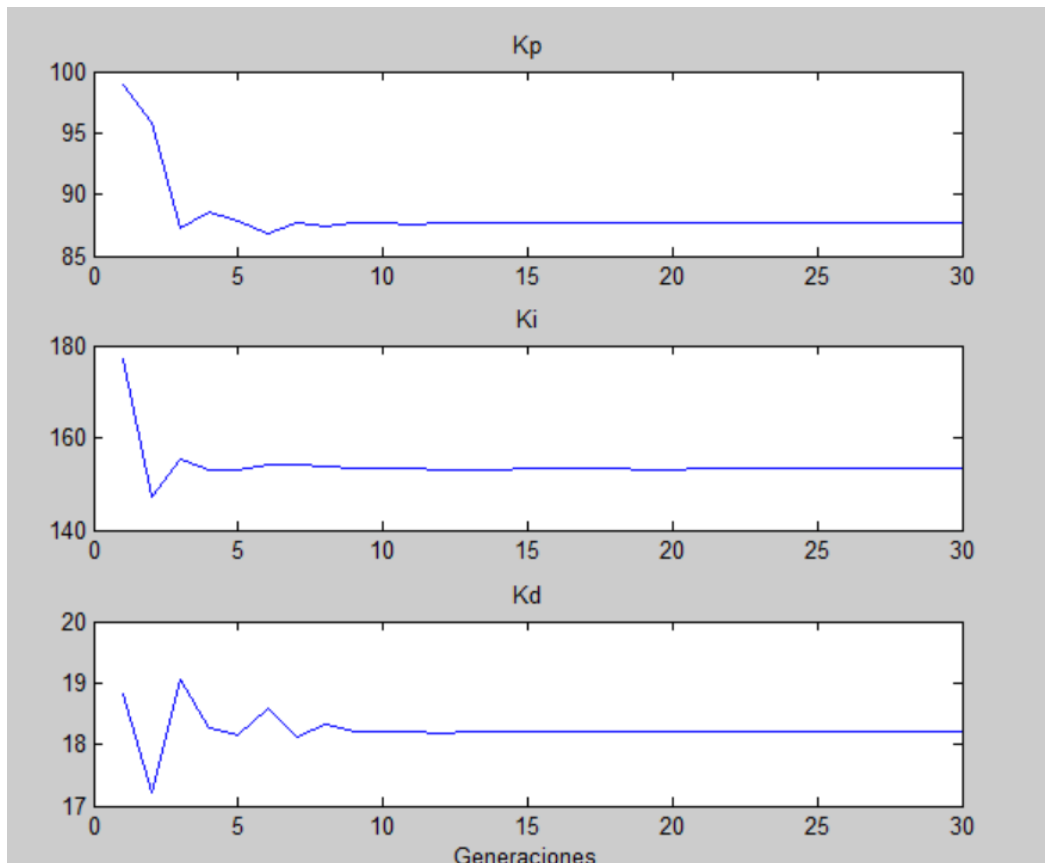
<b>Valores obtenidos con el método de ubicación de polos y ceros para la sintonización de controlador PID</b>	<b>Valores obtenidos con el algoritmo genético para la sintonización de controlador PID</b>		
	<b>1er</b>	<b>2da</b>	<b>3ra</b>
Kcp= 53,90	Kcp= 84,41	53,28	88,16
Tip= 0,67	Tip= 8,25	1,71	195,15
Tdp= 0,12	Tdp= 15,94	0,30	11,35
<b>Criterio de la integral de valor absoluto del error</b>	<b>Criterio de la integral de valor absoluto del error</b>		
IAE= 0,03	IAE= 0,00	0,01	0,00
MP= 14,90	MP= 14,90	14,90	14,90
TMp= 0,25	TMp= 0,25	0,25	0,25
Ts= 0,56	Ts= 0,56	0,56	0,56

*Tabla 1 Valores obtenidos en las respectivas simulaciones*

Se realizó la sintonización del PID con el método de ubicación de polos y ceros, el cual es un procedimiento matemático que obedece a fórmulas que garanticen la estabilidad y la efectiva respuesta del sistema en su parte dinámica.

También se realizó la sintonización del mismo ejercicio utilizando el algoritmo genético, con un criterio de función de costo de la integral del error absoluto IAE, como puede constatar en la tabla 1, anterior se sintonizó el algoritmo genético con diferentes rangos para las variables a encontrar como son:  $K_p, T_i, T_d$ , obteniendo cada vez mejores resultados en cuanto al IAE respecto al método de polos y ceros.

Los algoritmos heurísticos tal como el algoritmo genético tienen condiciones para terminar el proceso de búsqueda, estas condiciones dependen del proceso a optimizar o por el número de iteraciones. En este caso se eligió un número límite de iteraciones tomando en cuenta la convergencia del algoritmo, este número se estableció de acuerdo a las pruebas que se realizaron. En la figura 3, se presenta la evolución de las ganancias del controlador PID. El número de iteraciones máximo es de 30 pero resulta que el algoritmo genético converge a partir de la décima iteración.



#### 4. CONCLUSION

Los algoritmos heurísticos tal como el algoritmo genético tienen condiciones para terminar el proceso de búsqueda, estas condiciones dependen del proceso a optimizar o por el número de iteraciones. En este caso se eligió un número límite de iteraciones tomando en cuenta la convergencia del algoritmo, este número se estableció de acuerdo a las pruebas que se realizaron.

#### 5. REFERENCIAS

- [1] T. Álamo, “Diseño del Controlador PID,” *Univ. Sevilla, Dep. Ing. Sist. y Automática*, pp. 1–37, 2007.
- [2] D. Cavada Hernández, “Autosintonización de lazos de control PID en controladores programables,” phdthesis, Universidad Autónoma de Nuevo León, 1998.
- [3] C. Industrial and A. Wind-up, “Optimización de señal de control en reguladores PID con arquitectura antireset Wind-Up,” no. 30, pp. 24–31, 2011.
- [4] S. Ortiz Santos, “Sintonización de un controlador PID basado en un algoritmo heurístico para el control de un Ball and Beam,” phdthesis, 2015.
- [5] F. De Ingeniería, E. Tacconi, R. Mantz, and J. Solsona, “Versión electrónica editada por : Tania Salazar y Ana Roquez. Año 2005,” 2005.
- [6] M. Pinto, “Universidad de pamplona facultad de ingenierias y arquitectura

departamento de ingeniería electrónica, eléctrica, telecomunicaciones y sistemas,” 2006.

- [7] C. Autom, “Universidad Autónoma de Querétaro Facultad de Ingeniería Maestría en Instrumentación y Control Automático México,” 2011.
- [8] L. Hernandez-Corredor, J. Dorado, A. Quintero-Moreno, I. Ortiz, A. Buzòn, B. R. A. Jaimes, J. C. F. Galindo, and J. A. G. Camperos, “Desarrollo de una herramienta en Matlab para Sintonización de Controladores PID, utilizando algoritmos genéticos basado en técnicas de optimización multiobjetivo,” *Rev. del Sist. Cienc. Tecnol. e Innovación*, vol. 1, no. 1, pp. 80–103, 2014.
- [9] C. R. Luna Ortiz, “Diseño e Implementación de un algoritmo genético en FPGA para sintonización de controladores PID,” phdthesis, 2014.
- [10] L. J. Marín, “Método de ubicación de polos y ceros para la sintonización de controladores PID,” *Proy. Eléctrico (Bachillerato), Esc. Ing. Eléctrica, Univ. Costa Rica*, 2005.