

Aplicación de los algoritmos genéticos para el diseño de un controlador PID adaptativo

Application of genetic algorithms for the design of an adaptive PID controller

ILBER A. RUGE

Ingeniero Electrónico, Docente Electrónica Digital de la Universidad de Cundinamarca, investigador Grupo GITEINCO, Msc(c) Ingeniería de Control Industrial de la Universidad de Ibagué, iruge@unicundi.edu.co, ilberruge@yahoo.es.

MIGUEL A. ALVIS

Ingeniero Electrónico e Investigador del grupo GITEINCO de la Universidad de Cundinamarca. Correo electrónico: miguel.alviz@gmail.com.

Clasificación del artículo: investigación (conciencias)

Fecha de recepción: 22 de octubre de 2009

Fecha de aceptación: 29 de enero de 2010

Palabras clave: optimizar, control adaptativo, algoritmos genéticos, función error operadores genéticos

Key words: optimize, adaptable control, genetic algorithms, function error, genetic operators

RESUMEN

Los principios metodológicos del control adaptativo fueron introducidos en los años setenta y se culminaron en la última década, con excelentes aplicaciones prácticas, que pusieron de relieve la capacidad de esta tecnología de optimizar el control de las plantas industriales. Un ejemplo de esto son los algoritmos genéticos aplicados a controladores PID que pueden trabajar en un proceso de optimización, desconociendo la dinámica de la planta del sistema, ya que sólo busca converger mediante operadores genéticos a una óptima solución basándose sólo en una función de error.

ABSTRACT

The methodological principles of the control adaptativo were introduced in the years seventy culminating in the last decade with practical excellent applications that put of relief the capacity of this technology of optimizing the control of the industrial plants. An example of this is the genetic algorithms applied controllers PID that can work in a process of optimization ignoring the dynamics of the plant of the system since single search to converge by means of genetic operators to a good solution being based alone in an error function.

1. Introducción

En la selección natural los individuos más fuertes sobreviven a los cambios que se generan en su entorno, el hombre como individuo, siempre ha tratado de adaptarse a los cambios que son inherentes a él, creando soluciones a los problemas cotidianos; primero, lo hizo utilizando herramientas rudimentarias, pero conforme pasaba el tiempo sus herramientas se hacían complejas, ya que las nuevas necesidades así lo exigían.

Esta misma lógica la retomaron los sistemas de control, los cuales fueron automatizando sus procesos mediante el control adaptativo. Esta nueva forma de realizar control tiene diferentes aplicaciones como redes neuronales lógica difusa y los algoritmos genéticos, entre otros. Estos últimos toman una población de individuos, los cuales se evalúan para obtener un sobreviviente que cumpla con todas las características necesarias a la solución.

2. Estado del arte

El desarrollo de controladores adaptables utilizando Algoritmos Genéticos es un método no muy utilizado, debido al desconocimiento del uso de las características de los AG en este tipo de aplicaciones. Algunos de los trabajos realizados al respecto y que fueron tomados como base para la realización de este proyecto están referenciados por [1], [2], [3] y [4]:

3. Controlador pid

El controlador PID [5] es el denominado controlador de tres términos El primer término Kp tendrá el efecto de reducir el tiempo de elevación. Un control integral Ki que tendrá el efecto de eliminar el error de estado estacionario, pero puede empeorar la respuesta transitoria. Y un control derivativo Kd que tendrá el efecto de incrementar la estabilidad del sistema, reduciendo el sobrepico, y mejorando la respuesta transitoria. Los efectos de cada uno de los controladores Kp , Kd , y Ki en un sistema a lazo cerrado.

Este controlador está definido por la función de transferencia mostrada en la ecuación 1.

$$G = kp + \frac{ki}{s} + kds \quad (1)$$

La representación de la ecuación anterior en el dominio del tiempo, está dada por la ecuación 2.

$$U(t) = k \left\{ et + \frac{1}{Ti} \int_0^t et dt + Td \frac{\partial et}{\partial t} \right\} \quad (2)$$

4. Algoritmos genéticos

Los algoritmos genéticos [6] constituyen una técnica de búsqueda fundamentada en el proceso de evolución natural en la cual los individuos más adaptados tienen mayores probabilidades de sobrevivir y de transferir su material genético a las siguientes generaciones. 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.

Dichos procesos están dados por operadores genéticos que definen la estructura de un algoritmo genético.

4.1. Operadores genéticos

4.1.1. Fitness

Es la responsable de evaluar a los individuos y asignarle valores altos de adaptación; ya que es una función completamente dependiente del problema, puede determinar qué tan buenos o malos serán los resultados y la convergencia o no del método.

4.1.2. Función objetivo

la función objetivo proporciona una medida de desempeño del sistema asociado a cierto individuo en la población.

4.1.3. Operador de selección:

el proceso de selección busca escoger ciertos individuos de la población, quienes darán origen a las futuras generaciones. Por lo general, la selección depende del valor de la función de evaluación de cada individuo.

4.1.4. Operador de cruce

Consiste en mezclar la información genética de dos individuos a fin de generar nuevos individuos.

4.1.4. Operador de mutación

Consiste en alterar las características genéticas de

un individuo, con el objeto de aumentar la probabilidad de exploración del espacio de búsqueda y disminuir el riesgo de estancamiento del algoritmo en óptimos locales.

4.2. Estructura General de los AG.

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 la función 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.

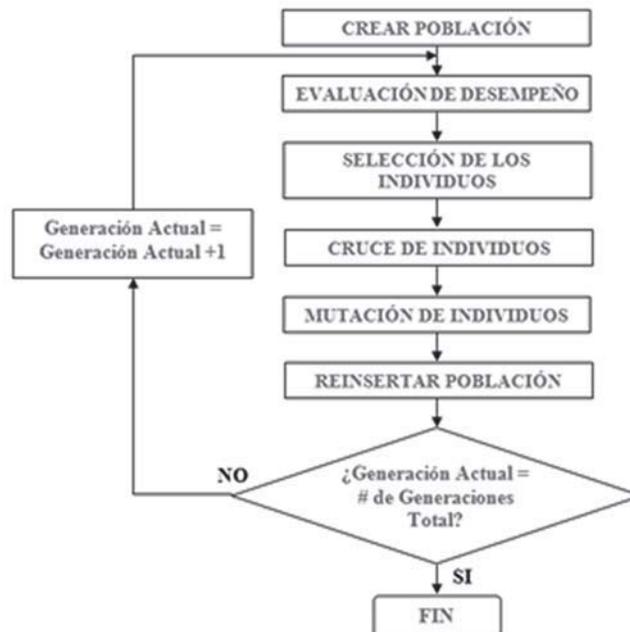
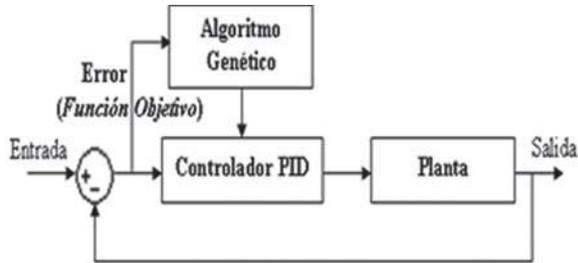


Figura 1. Diagrama de flujo de algoritmo genético simple

4.3. Algoritmo genético aplicado a un PID



A continuación se muestra el código fuente en Matlab [4] obtenido para aplicar los operadores genéticos del algoritmo al controlador PID adaptable.

Figura 2. Algoritmos genéticos aplicados a controladores PID.

```

%% - - - INICIO DEL ALGORITMO GENÉTICO - - -
>> NIND = 40; % NÚMERO INDIVIDUOS POBLACIÓN
>> MAXGEN = 100; % NÚMERO DE GENERACIONES
>> NVAR = 3; % NÚMERO DE VARIABLES
>> PRECI = 20; % PRECISIÓN BINARIA
>> FieldD=[rep([PRECI],[1,NVAR]); [0.01;10], [0.01;10], [0.01;2]; rep([0; 0;
1; 1], [1,NVAR])];
>> Chrom = crtbp(NIND,NVAR*PRECI); % POBLACIÓN INICIAL
>> Best = []; % MEJOR DE LA POBLACIÓN ACTUAL
>> gen = 1; % CONTADOR DE GENERACIONES
% EVALUACIÓN DE LA FUNCIÓN OBJETIVO
>> pobPID = bs2rv(Chrom,FieldD);
>> objV = objfunPID(pobPID);
>> [minimo,ubi] = min(objV);
>> Best = [Best; minimo, VP(ubi), VI(ubi), VD(ubi)];
% LOOP GENERACIONAL
>> while gen < MAXGEN;
% ASIGNACIÓN DESEMPEÑO
>> FitnV = ranking(objV);
% SELECCIÓN DE INDIVIDUOS
>> SelCh = select('sus', Chrom, FitnV);
% CRUCE
>> SelCh = recomb('xovsp',SelCh,0.7);
% MUTACIÓN
>> SelCh = mut(SelCh);
% EVALUACIÓN DE LOS DESCENDIENTES
>> pobPID = bs2rv(SelCh, FieldD);
>> objV = objfunPID(pobPID);
>> [minimo,ubi] = min(objV);

```

```

>> Best = [Best; minimo, VP(ubi), VI(ubi), VD(ubi)];
% REINSERTAR DESCENDIENTES A POBLACIÓN ACTUAL
>> [Chrom objV] = reins(Chrom, SelCh, 1, 1, objV, ObjVSel);
>> gen = gen+1;
>> end

% RECUPERA EL MEJOR VALOR DE LA MATRIZ BEST Y GRAFICA ESTE VALOR DE PID
>> Best1 = Best(:,1);
>> [m,i] = min(Best1);
>> Best(i,1);
>> Kp = Best(i,2);
>> Ki = Best(i,3);
>> Kd = Best(i,4);
    
```

Para el algoritmo definido anteriormente, la función objetivo está dada por la ecuación 3.

$$objfunPID = \int error(t) dt \quad (3)$$

$$G(s) = \frac{k \cdot e^{-t_d \cdot s}}{\tau s + 1} \quad (4)$$

siguiente manera:

5. Pruebas y resultados

Con el ánimo de evaluar el desempeño del algoritmo y teniendo en cuenta cambio en los parámetros del sistema bajo evaluación, se realizaron pruebas para dos tipos de sistemas: Sistema de Primer Orden y de Segundo Orden.

5.1. Pruebas y resultados aplicados a un sistema de primer orden

El modelo clásico para la representación de un sistema de primer orden son los sistemas térmicos [5]. La función de transferencia del modelo seleccionado para las respectivas pruebas del algoritmo diseñado está dado por la ecuación 4.

Los parámetros del sistema están definidos de la

Tabla 1. Parámetros de la planta G(s)

Parámetros de la planta		
Parámetro	Símbolo	Valor
Ganancia del sistema	k	3
Tiempo de retardo	t_d	0seg
Constante de tiempo	τ	4seg

Por ejemplo, para $k = 3$, tiempo de retardo $t_d = 0$ seg. y constante de tiempo $\tau = 4$ seg, la planta queda definida en la ecuación 5.

$$G(s) = \frac{3}{4s + 1} \quad (5)$$

Ante una entrada escalón unitario, el sistema pre-

senta una respuesta tal como se muestra en la figura 4.

De acuerdo con los resultados mostrados en la fi-

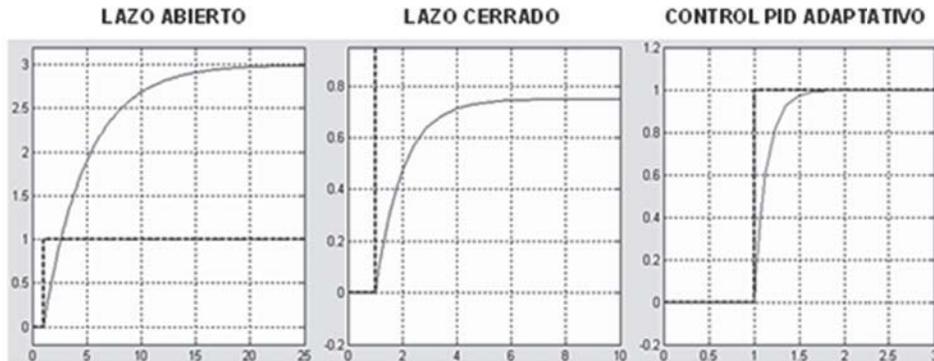


Figura 4. Respuesta del sistema G(s) controlado ante una entrada escalón unitario

gura 4 se puede observar que el error en estado estacionario en lazo cerrado es grande. Por otro lado, la acción del controlador PID obtenido a través del algoritmo genético elimina dicho error en estado estacionario y además mejora significativamente el tiempo de establecimiento.

ponde a la dinámica de un sistema motorbrazo de una unidad de disco duro convencional [7]. Los parámetros de dicho sistema son:

- Giros: *6000 giros /minuto*
- Distancia cabezal-disco: *< 100 nm*
- Precisión de posición: *< 100 μm*
- Pista a pista: *< 10 ms*

5.2 Pruebas y resultados aplicados a un sistema de segundo orden

Los parámetros físicos que se deben conocer de la

El modelo de segundo orden seleccionado para la realización de las pruebas del algoritmo, corres-

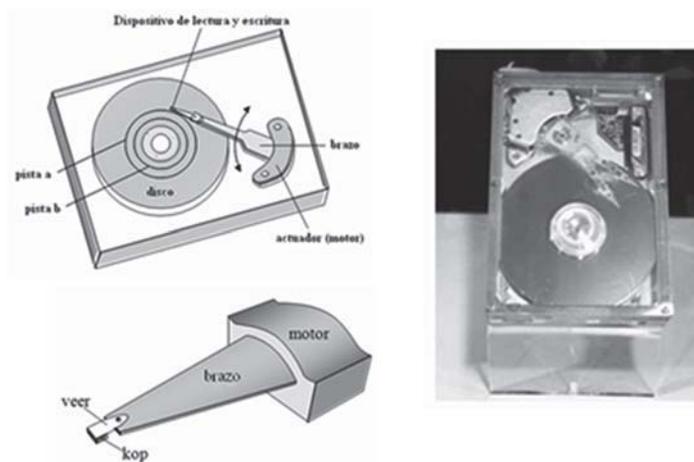


Figura 5. Partes de un disco duro, Motor y brazo

dinámica del sistema para realizar su análisis, son los mostrados en la tabla 2.

La función de transferencia del sistema queda de-

Tabla 2. Parámetros para la dinámica de una unidad de disco duro

Typical Parameters for Disk Drive			
Parameter	Symbol	Typical Value	
Mass of arm+head	m	25	g
Inertia of arm+head	J	100	$\text{g}(\text{cm})^2$
Friction	b	20	$(\text{mN})(\text{cm})\text{s}$
Amplifier	K_a	10-1000	
Field resistance	R	1	Ω
Motor constant	K_t	5	$(\text{mN})(\text{cm})/\text{A}$
Field inductance	L	1	mH

finido por:

$$G(s) = \frac{25}{s(1 + 0.05s)} \quad (6)$$

La respuesta a la señal escalón del sistema $G(s)$ en lazo abierto, lazo cerrado y el sistema controlado mediante AG, como se muestra en la figura 6.



Figura 6. Respuesta del sistema controlado mediante PID adaptativo

De acuerdo con los resultados mostrados en la figura 6, se puede observar que el sistema en lazo cerrado está controlado, pero, muestra un sobrepico de aproximadamente el 20% y un tiempo de establecimiento de aproximadamente. 0.5s. Ahora, el sistema controlado mediante el PID obtenido

por el AG muestra que el sistema es controlado y que además la respuesta al escalón presenta un sobrepico de aproximadamente un 10% y un tiempo de establecimiento de 0.2s, lo cual indica un óptimo rendimiento del algoritmo diseñado.

Tabla 3. Tiempo de ejecución del AG sin simulink

DURACION DEL ALGORITMO GENETICO SIN SIMULINK		
Numero de generaciones	Numero de individuos	Tiempo de ejecución (seg)
100	20	0.185484
500	20	0.999374
1000	20	1.844187
100	40	0.326361
500	40	1.348266
1000	40	2.482890
100	60	0.548797
500	60	1.479810
1000	60	2.664453

Tabla 4. Tiempo de ejecución del AG con simulink

DURACION DEL ALGORITMO GENETICO SIN SIMULINK		
Numero de generaciones	Numero de individuos	Tiempo de ejecución (seg)
100	20	0.185484
500	20	0.999374
1000	20	1.844187
100	40	0.326361
500	40	1.348266
1000	40	2.482890
100	60	0.548797
500	60	1.479810
1000	60	2.664453

6. Cálculo del costo computacional del algoritmo

Dado que el algoritmo genético desarrollado evalúa el desempeño de la población haciendo uso de Simulink, se utiliza los comandos TIC y TOC de Matlab para determinar cuánto es el tiempo que tarda en ejecutarse dicho algoritmo. Se llevaron a cabo pruebas variando el número de generaciones y el número de individuos de la población; de igual manera, se realizaron pruebas contando el tiempo de ejecución sin utilizar simulink (ver tabla 3) y con simulink (ver tabla 4) para observar el tiempo que tarda el algoritmo en evaluar los individuos de la población.

7. Conclusiones

Para obtener un funcionamiento óptimo del algoritmo genético se recomienda un número mayor en el número de generaciones, pues entre más generaciones sean evaluadas se obtendrá un mejor resultado en el desempeño de los individuos. Inicialmente, se tomó como función objetivo minimizar el valor de error $e(t)$ del sistema, pero, al realizar pruebas con sistemas o plantas oscilatorias, el algoritmo asignaba un buen desempeño a individuos que llevaban el sistema de control a la inestabilidad. Es por esto que se optó por tomar como función objetivo la integral del error como valor de función objetivo de los individuos, obteniendo así un controlador PID de buen desempeño.

Referencias

- [1] J. Fiffe M. *Empleo de los algoritmos genéticos para el ajuste en línea de un regulador PID convencional*. Disponible en internet <http://proton.ucting.udg.mx/posgrado/cursos/idc/pdf/neuronales/A014.pdf>
- [2] S. Aguilar, *Implementación del Algoritmo PID adaptativo para controlar temperatura a través del computador usando algoritmos genéticos*. Universidad de Carabobo, Unidad de Automatización Industrial. Disponible en Internet <http://servicio.cid.uc.edu.ve/ingenieria/revista/a7n2/7-2-2.pdf>
- [3] A. Caro C. *Control multivariable para un helicóptero de dos grados de libertad utilizando algoritmos genéticos*. Pontificia Universidad Javeriana, Facultad de Ingeniería. Disponible en Internet <http://www.javeriana.edu.co/biblos/tesis/ingenieria/tesis96.pdf>
- [4] I. Ruge, «*Diseño de un regulador adaptable basado en Algoritmos Genéticos.*» Revista Colombiana de Tecnologías de Avanzada. Disponible en http://www.unipamplona.edu.co/unipamplona/hermesoft/portalIG/home_18/recursos/01_general/documentos/16052008/rev_tec_avan_art3_vol2_num6.pdf
- [5] R. Dorf, «*Sistemas de Control Moderno*». 10ª edición. Ed. Prentice Hall.
- [6] D. Goldberg, “*Genetic Algorithms in Search, Optimization, and Machine Learning.*” Ed. Addison-Wesley.
- [7] R. De Keyser, *Adaptive and Predictive Control. Master Course Industrial Control Engineering*. Universidad de Ibagué - Colombia. octubre de 2008.