ISSN: 1692-7257 Volumen 2 Número 6 año - 2005

Revista Colombiana de
Tecnologías de Avanzada

DESIGN OF A ADAPTIVE CONTROLLER BASED IN GENETIC ALGORITHMS

DISEÑO DE UN REGULADOR ADAPTABLE BASADO EN ALGORITMOS GENÉTICOS:

Ing. Ilber Adonayt Ruge

Universidad Pedagógica y Tecnológica de Colombia
Escuela de Ingeniería Electrónica
ilberruge@hotmail.com , ilberruge@yahoo.es
Sogamoso, Boyacá

Abstract: The real world is surrounded of several processes that due to its nature, it is difficult to determine a mathematical model that allows to predict its behavior. The theory of conventional control is based in the dependability that one can have of the pattern, for this way to be able to design a control strategy that can fulfill the wanted requirements. This is the main reason that motive to design a strategy of adaptive control based on AG that omits the modeling of the process or plant (Black Box) and respond strictly to the dynamics of the process, always maintaining under control.

Resumen: El mundo real está rodeado de varios procesos que debido a su naturaleza, son difíciles de determinar en un modelo matemático que permita predecir su comportamiento. La teoría del control convencional se basa en las formalidades que pueda tener cada patrón, y así poder diseñar una estrategia de control que pueda satisfacer los requerimientos necesarios. Esta es la razón principal que motiva el diseño de una estrategia de control adaptable, basado en AG que omite el modelo del proceso de la planta (caja negra) y responde estrictamente a las dinámicas del proceso, siempre manteniéndola bajo control.

Keywords: Evolutionary programming, AG, Selection, Crossover, Mutation, PID, FPGA.

1. INTRODUCCIÓN

La teoría de la evolución (que no es tal teoría, sino una serie de hechos probados), fue descrita por Charles Darwin en el Beagle, en el libro sobre el origen de las especies por medio de la selección natural. Darwin pensaba que los rasgos de un ser vivo eran como un fluido, y que los "fluidos" de los dos padres se mezclaban en la descendencia; esto provocaba el problema de que al cabo de cierto tiempo, un población tendría los mismos rasgos intermedios.

Muchos problemas de optimización en el mundo de la ingeniería, son muy complejos y requieren de técnicas de optimización convencionales muy rígidos. Desde el año 1960 se ha incrementado el interés por IMITAR formas de vida humana para resolver problemas de optimización fuertes.

2. ALGORITMOS GENETICOS

2.1. Fundamentación de Algoritmos Genéticos.

Recientemente, los algoritmos genéticos (AG) han recibido una atención especial, debido a su gran potencial como una TECNICA DE OPTIMIZACION para resolver problemas complejos, y son ampliamente usados en áreas de ingeniería industrial.

2.*2 Estructura General de los AG.*

Los algoritmos genéticos difieren de las técnicas de búsqueda convencional, en que éstas comienzan con una población inicial de soluciones aleatorias (Población). Cada individuo en la población es llamado cromosoma, el cual representa una solución al problema. Un cromosoma es una cadena de símbolos por lo general binaria (no necesariamente). Los cromosomas evolucionan a través de iteraciones sucesivas, llamada también generación. Durante cada generación los nuevos cromosomas son evaluados, usando algunas medidas de aptitud. La creación de la próxima generación son nuevos cromosomas llamados descendientes. Estos son formados por:

- Combinación de dos cromosomas de la generación actual usando un operador de cruce (croosover).
- Modificando un cromosoma usando un operador de mutación (mutation).

Una nueva generación es formada:

- Seleccionando, de acuerdo al valor de aptitud, algunos de los padres e hijos.
- Rechazando los de valor de aptitud no deseada, pero manteniendo el tamaño de la población constante.

Los mejores cromosomas tienen una alta probabilidad de ser seleccionado. Después de diferentes generaciones el algoritmo converge al mejor cromosoma, el cual representa la solución ÓPTIMA.

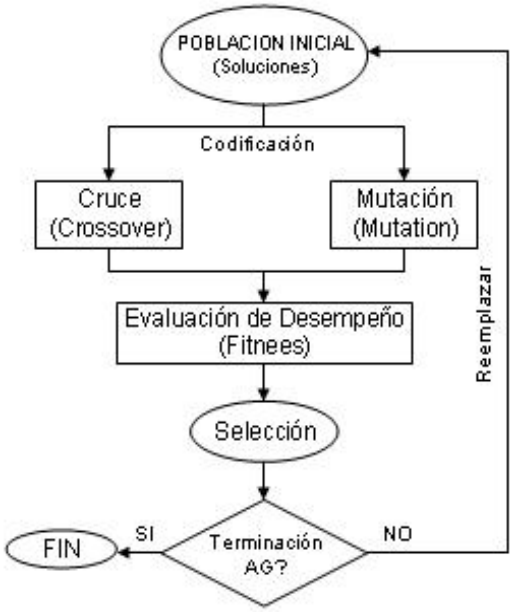

Fig. 1. Diagrama de Flujo de un Algoritmo Genético Simple.

2.*3 Terminación del Algoritmo Genético.*

Debido a que un Algoritmo Genético es un método de búsqueda estocástico, es difícil especificar formalmente un criterio de convergencia. Como el desempeño de una población puede permanecer estática por un número de generaciones antes de que un individuo superior sea encontrado, la aplicación de criterios de terminación convencional es algo problemático. Una práctica común es terminar el Algoritmo Genético después de un número preestablecido de generaciones y entonces la prueba de calidad de los mejores miembros de la población determina la mejor solución al problema. Si la solución encontrada no es aceptable, el Algoritmo Genético deberá comenzar y realizar un refrescamiento de las posibles soluciones.

2.*4 Exploración y Explotación.*

Si la función objetivo que se desea minimizar (ó maximizar) es definida por (1), la población inicial queda determinada así:

$$Ras(x) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1 + \cos 2\pi x_2) \quad (1)$$

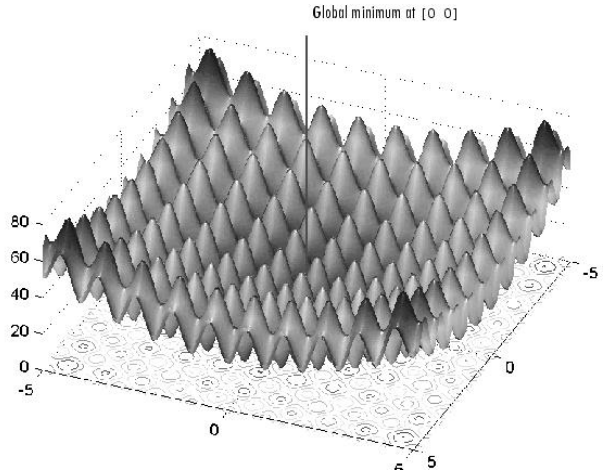

Fig. 2. Espacio de búsqueda determinado por (1).

Explotación: Función asumida por el operador genético cruce (croosover), y consiste en la búsqueda de la mejor solución local; sin que esto implique que sea el óptimo global.

Exploración: Función asumida por el operador genético mutación (mutation), y consiste en buscar regiones alternativas dentro del espacio de búsqueda que contengan posibles soluciones globales.

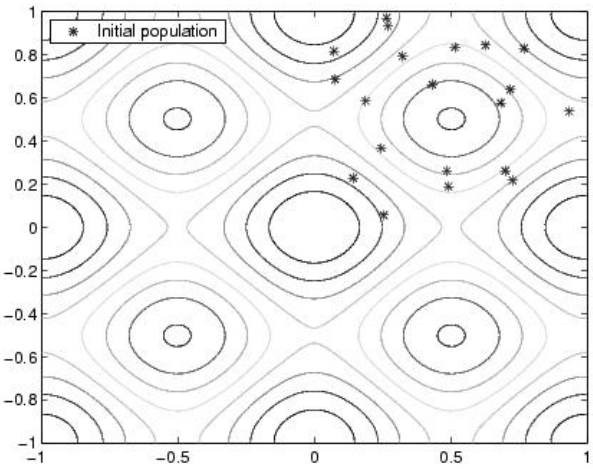

Fig. 3. Desempeño de los operadores cruce y mutación dentro del espacio de búsqueda.

3. CONTROL ADAPTATIVO BASADO EN ALGORITMOS GENETICOS

Dos procedimientos básicos para el diseño de controladores adaptativos son:

- Identificación de características de la planta y toma de decisión.
- Control adaptativo con Modelo de Referencia.

Cada uno de estos métodos requiere determinar ciertas características de la planta que permita obtener un modelo aproximado del proceso, y de acuerdo a éste, diseñar la estrategia correspondiente o deseada.

El Control Adaptativo basado en Algoritmos Genéticos aquí propuesto, busca explorar una alternativa de control basado en la optimización de la señal de error del proceso en lazo cerrado, que si se asume como **función objetivo** permite determinar un modelo de controlador adecuado sin necesidad de tener un amplio conocimiento de las características dinámicas de la planta (Black Box). Es así como se reduce considerablemente la fundamentación matemática que trae consigo el diseño de los controladores adaptativos mencionados anteriormente, ya que la confiabilidad del controlador obtenido por AG depende de la estrategia que se adopte y de los operadores genéticos seleccionados.

Una descripción global de la síntesis del controlador adaptativo basado en AG es:

*Crear población inicial***:** Se compone de un número de cromosomas (Nind) cuyos genes (real, entero, binario ó híbrido) corresponden a valores P_i, I_i y D_i del controlador base. El conjunto de cromosomas determina el espacio de búsqueda que el AG utiliza para encontrar la solución más óptima.

$$Chrom = \begin{bmatrix} P_i & I_i & D_i \end{bmatrix} \quad (2)$$

*Operador Cruce***:** Es el operador genético principal. Su función corresponde a la exploración del espacio de búsqueda.

Operador Mutación: Es el operador secundario dentro del AG. Su función corresponde a la explotación del espacio de búsqueda.

Operador Selección: Su función principal es evaluar el desempeño de cada uno de los cromosomas de la población inicial, y de acuerdo al resultado asigna probabilidades de supervivencia.

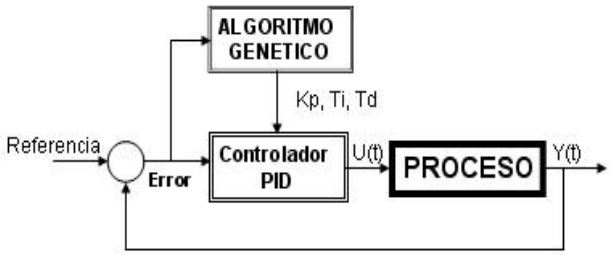

Fig. 4. Esquema del control adaptativo basado en algoritmos genéticos.

Si se toma como modelo base un controlador PID discreto, la síntesis para la función objetivo es:

$$U(t) = K\{e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{\partial e(t)}{\partial t}\} \quad (3)$$

Usando el método de aproximación por operador integral de Tustin para discretizar (3):

$$U_k = K\left[e_k + \frac{T}{T_i}\frac{e_0 + e_k}{2} + \frac{T}{T_i}\sum_{i=1}^{k-1} e_k + \frac{T_d}{T}(e_k - e_{k-1})\right] \quad (4)$$

Simplificando (4) se tiene :

$$U_k = U_{k-1} + (q_0 e_k + q_1 e_{k-1} + q_2 e_{k-2}) \quad (5)$$

Para hacer que la función objetivo sea más selectiva, se calcula el error de acuerdo al siguiente criterio:

$$e_k = abs(ref - y)^2 \quad (6)$$

donde *ref* es el valor de referencia del sistema en lazo cerrado y *y* el valor de la señal de salida.

El código fuente tanto de la función objetivo como del algoritmo genético general es:

function ObjVal = objfunPID(Chrom, Ref);

Error(3)=Error(2); % $e_{k-2} = e_{k-1}$

Error(2)=Error(1); % $e_{k-1} = e_k$

U(2)=U(1); % $U_{k-1} = U_k$

% Cálculo de Error

Error(1)= abs(Ref-Y)^2; % e_k

% Evaluación del Controlador PID Discreto
U(1)=U(2)+(Chrom(1).*Error(1)+Chrom(2).*Error (2)+… Chrom(3).*Error(3));

ObjVal = Error(1);

% End of function

% Inicio Algoritmo Genético (PID.m)

```
Ref = 2;              % Señal de Referencia
NIND = 40;            % Número individuos población
MAXGEN = 100;         % Número de generaciones
NVAR = 20;            % Número de variables
PRECI = 20;           % Precisión binaria

FieldD=[rep([PRECI],[1,NVAR]);          rep([-
512;512],...
[1, NVAR]); rep([1; 0; 1; 1], [1, NVAR])];

% Población Inicial
  Chrom = crtbp(NIND, NVAR*PRECI);

% Mejor de la población actual
  Best = NaN*ones(MAXGEN,1);
  gen = 0;            % Contador de generaciones

% Evaluar población inicial
  ObjV = objfunPID (bs2rv(Chrom, FieldD));

  Best(gen+1) = min(ObjV);

% Loop generacional
  while gen < MAXGEN,

% Asignación desempeño
      FitnV = ranking(ObjV);

% Selección de individuos
      SelCh = select('sus', Chrom, FitnV, GGAP);

% cruce
      SelCh = recombin('xovsp',SelCh,0.7);

% mutación
      SelCh = mut(SelCh);

% Evaluar descendientes
      ObjVSel = objfunPID(bs2rv(SelCh, FieldD));
```

```
% Reinsertar descendientes a población actual
[Chrom ObjV]=reins(Chrom, SelCh, 1, 1, ObjV, ObjVSel);

% Incrementar contador
      gen = gen+1;

Best(gen+1) = min(ObjV);
  end
```

Para realizar las pruebas funcionales del AG, se optó por utilizar un modelo simulink de la planta junto con el MCode de Xilinx Blockset (System Generator), el cual enmascara el código PID.m y permite realizar aplicaciones más versátiles. Los resultados de las pruebas se muestran a continuación:

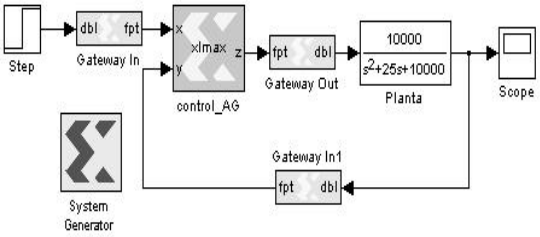

Fig. 5. Modelo simulink de controlador adaptativo basado en AG.

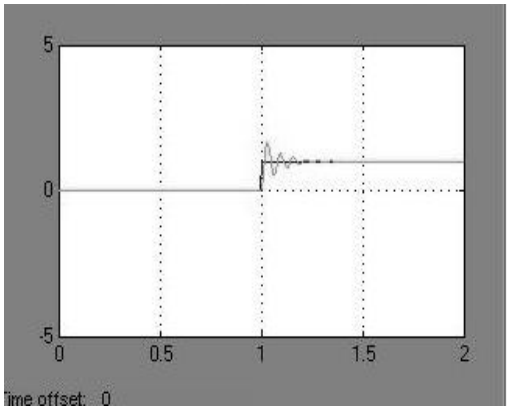

Fig. 6. Respuesta de la planta modelo en lazo abierto.

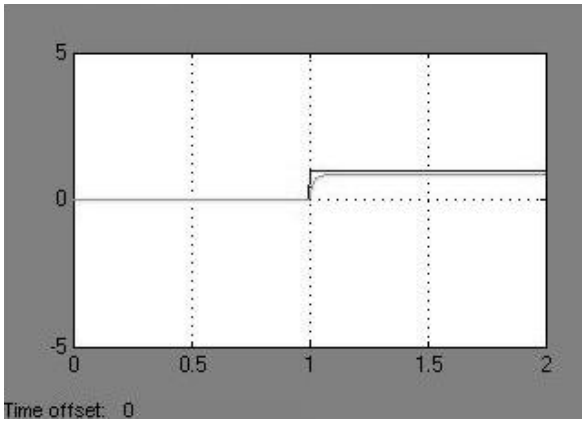

Fig. 7. Respuesta de la planta en lazo cerrado (controlador adaptativo AG).

Para verificar la adaptabilidad del sistema, se modificaron los parámetros de la planta modelo obteniéndose los siguientes resultados:

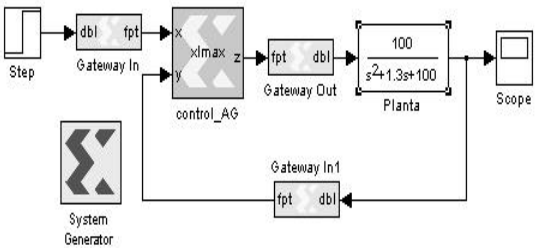

Fig. 8. Modelo simulink de controlador adaptativo basado en AG variando parámetros de planta modelo.

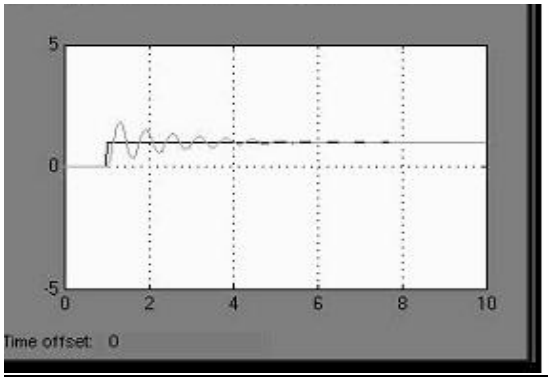

Fig. 9. Respuesta de la planta modelo en lazo abierto.

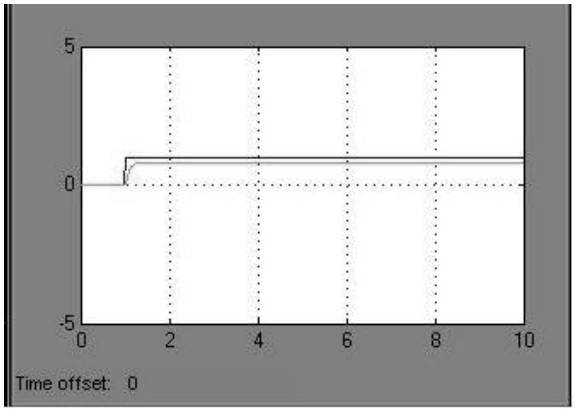

Fig. 10. Respuesta de la planta en lazo cerrado (acción del controlador adaptativo AG).

La siguiente tabla muestra los parámetros PID dados por el algoritmo para cada uno de los casos expuestos anteriormente:

Tabla 1. Resultado de parámetros PID dados por el AG.

Función de Transferencia planta	**p**	**I**	**D**	**error**
$\frac{100}{s^2+1.3s+100}$	4.21	0.022	0.52	4.85e-3
$\frac{10000}{s^2+25s+10000}$	5.32	0.55	0.10	3.32e-2

4. CONTROL ADAPTATIVO SOBRE PLATAFORMA FPGA

El sistema anteriormente descrito se ha desarrollado con el fin de poder ser utilizado en aplicaciones reales, por lo cuál será conveniente implementar el algoritmo sobre alguna plataforma que interactúe con el mundo real. Se propone trabajar sobre dispositivos FPGA, ya que gracias a su capacidad de procesamiento paralelo y velocidad de procesamiento, permite implementar funciones complejas y además, es posible realizar modificaciones de último minuto sin que esto implique grandes alteraciones en el hardware y software. Actualmente se esta evaluando la plataforma de desarrollo **Pegasus** de la compañía **DIGILENT Inc**, centrada en una FPGA Spartan2 cuyas características de operación son:

- Xilinx Spartan 2 FPGA con 50Kgates y opera a frecuencia de 200MHz.
- XCF01S Xilinx Plataforma ROM Flash.
- Una colección de dispositivos I/O incluyendo 8
- Leds, 4 displays siete segmentos, 4 pulsadores y 8 interruptores.
- Un oscilador 50MHz y un socket para oscilador secundario.
- Puertos PS/2 y VGA.
- 96 pines I/O distribuidos en tres conectores de 40 pines.
- Todos los pines I/O protección contra corto circuito.
- Un puerto de programación JTAG.

La siguiente figura muestra el diagrama esquemático de la plataforma de desarrollo anteriormente descrita:

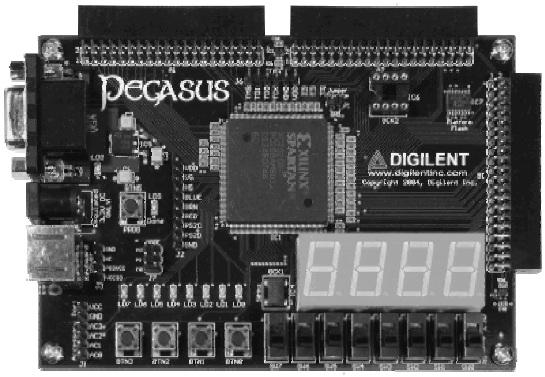

Fig. 9. Plataforma de desarrollo Spartan 2 Pegasus.

El software de desarrollo es el ISE 6.3i de la compañía Xilinx, el cual contiene todas las herramientas necesarias para el máximo aprovechamiento de la FPGA. Esta herramienta también se encuentra bajo evaluación.

5. CONCLUSIONES

1. El no tener que realizar un modelamiento del proceso que se desea controlar (Black Box), hace del control adaptativo una herramienta altamente aplicable en el área industrial, aún más si dichos controladores son basados en procesos de inteligencia artificial como redes

neuronales, lógica difusa ó algoritmos genéticos.

2. Los Algoritmos Genéticos son un mecanismo de optimización probabilístico, ya que parten de una población de posibles soluciones y no de una en particular. A diferencia de otros procesos de optimización, el propósito general es hallar el mínimo ó máximo de una función objetivo.

3. El control adaptativo basado en AG debido a su naturaleza estocástica, ofrece un universo de posibles soluciones. Por tal razón, el diseñador debe poner a consideración si la solución dada por el algoritmo es adecuada o no, de lo contrario se deberá escoger un criterio de terminación del AG más estricto.

REFERENCIAS

Chipperfield, A. y Fonseca, C. Genetic Algorithms TOOLBOX User's Guide, Versión 1.2, University of Sheffield.

Duarte, G. UNGenético: Una librería en C++ de Algoritmos Genéticos con Codificación Híbrida, Universidad Nacional de Colombia.

Mitsuo, G. Genetic Algorithms and Engineering Design, A Wiley-interscience Publication.

Ogata, K. (1987). *Discrete-Time Control Systems*. Prentice-Hall, Englewood Cliffs, NJ.

Xcell Journal, (2004). Xilinx, Issue 49, pp. 75-77.

Xcell Journal, (2004). Xilinx, Issue 50, pp. 101-105.

Xcell Journal, (2004). Xilinx, Issue 51, pp. 51-59.

www.digilentinc.com/tutorials.html

www.xilinx.com/documentation.html