CONTROL EN LINEA CON ALGORITMOS GENETICOS Y RECOCIDO SIMULADO

On line control with genetic algorithm and simulated annealing

RESUMEN

Se presenta una técnica de control inteligente que utiliza una ley de control PID, en la cual las ganancias primero se calculan fuera de línea con un algoritmo genético (GA) de coma flotante y luego el sistema trabaja en línea de manera adaptativa ajustando las ganancias a través de un algoritmo de recocido simulado (SA), este ultimo es mucho mas rápido que el algoritmo genético y por esto se selecciono para el control en tiempo real.

PALABRAS CLAVES: Control Inteligente, algoritmos genéticos, recocido simulado.

ABSTRACT

An intelligent control method is presented, it used a PID law, first a set of gains is calculated by a floating point genetic algorithm in off line mode , then the system work on line in adaptive mode with gains change trough a simulated annealing algorithm, it is more fast than a genetic algorithm and them is selected for real time control.

KEYWORDS: *intelligent control, genetic algorithms, simulated annealing.*

JOSÉ GABRIEL HOYOS G. *
Ingeniero Electricista M.Sc U.T.P.
Profesor Asistente
Universidad del Quindío
josegrabrielh@uniquindio.edu.co

JAIBER EVELIO CARDONA. *
Ingeniero Electrónico
M.Sc Univalle
Profesor Auxiliar
Universidad del Quindío
jaibercardona@uniquindio.edu.co

RAMIRO ARANGO
Licenciado en Física U.T.P.
Profesor Asistente
Universidad del Quindío
ramy@uniquindio.edu.co

*** GAMA: GRUPO DE AUTOMATIZACIÓN Y MÁQUINAS DE APRENDIZAJE.**

1. INTRODUCCIÓN

El mayor problema al utilizar algoritmos genéticos [2] en control, es el gran tiempo de cálculo, ya que es necesario poblaciones grandes y probar cada individuo de la población en varias muestras en la planta para poder obtener un valor de costo o "fitness". Un método que mejora el problema anterior es el uso de la técnica de recocido simulado, la cual puede trabajar con pequeñas poblaciones [3].

En [6] se presento una técnica de control en la cual se calculo de forma previa las ganancias de realimentación de estado para un péndulo invertido rotacional, en este articulo las ganancias de un PID se calculan fuera de línea con un algoritmo genético (GA) de coma flotante de manera similar a [6] y luego se continúan ajustando a través de un algoritmo de recocido simulado (SA), a diferencia de [7] donde se utilizo un algoritmo de pre-entrenamiento fuera de línea, en este se pretende que la adaptación de las ganancias del PID se den en tiempo de ejecución.

Primero se presentan brevemente la teoría de algoritmos genéticos y de recocido simulado, luego se presenta los resultados obtenidos al simular el sistema (figura 1) en Matlab y con una planta real usando el programa Builder c++ y una tarjeta de adquisición, por ultimo se presentan las conclusiones.

2. CONTENIDO

2.1 Algoritmos genéticos

Basados en la naturaleza, simulan procesos que se dan en los seres vivos como el cruce, la mutación, selección. La idea es que se comienza con una población aleatoria de individuos, se prueba cada individuo con una función de costo o "fitness" y los mejores se cruzan, también para provocar la aparición de nuevos individuos, algunos individuos mutan o se transforman. [1][4][5].

En este articulo el algoritmo genético se utiliza para el calculo fuera de línea de las constantes del control PID, para ello se requiere el modelo de la planta, el cual se utiliza para probar cada uno de los *P* individuos, al final de n muestras se obtiene un individuo o constantes que dan un control PID optimo.

La función de costo o "fitness" utilizada es:

$$se = \sum_{k=1}^{m} e^2(k) \quad (1)$$

$$f_{costo} = \frac{1}{se} \quad (2)$$

donde $e(k)$ es el error y *m* es el número de muestras.

Fecha de Recepción: 30 Mayo de 2007
Fecha de Aceptación: 20 Agosto de 2007

2.2 RECOCIDO SIMULADO:

Es una metodología de optimización basado en procesos termodinámicos (recocido), la idea es que a partir de puntos existentes se crean vecinos (versiones perturbadas de los puntos), a los cuales se les prueba su función de costo o "fitness" y si esta es de mejor calidad que la del punto existente, el vecino entra a reemplazar al punto existente. Por medio del enfriamiento se puede lograr que vecinos de mala calidad que se pueden presentar en alto porcentaje en la parte inicial del proceso, se vayan eliminando y se reduzca la probabilidad de su aparición.

Generalmente los algoritmos de recocido se ejecutan dentro de un lazo donde su ciclo finaliza cuando se cumple alguna meta deseada, esto no se puede aplicar a los sistemas de tiempo real, donde el tiempo limitado impide que la estrategia de control tenga lazos indefinidos, por lo que solo se ejecuta el algoritmo mostrado en la figura 2.0, el cual se desarrolla durante un tiempo determinado.

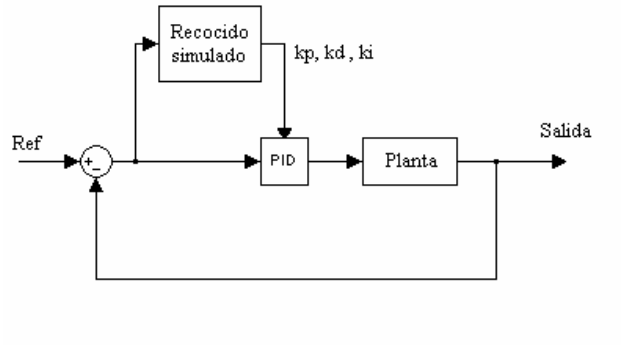

Figura 1. Esquema del sistema implementado.

La probabilidad ***prob*** de que se acepte un vecino de mala calidad se logra con la ecuación [7] [8]:

$$prob = \frac{\exp(fitness - fitness\_vecino)}{Temp} \qquad (3)$$

valor el cual se compara con un dato aleatorio entre 0 y 1.

La simulación de la disminución de la temperatura se logra multiplicando la temperatura por una constante de enfriamiento β menor a la unidad.

$$Temp = \beta Temp \qquad (4)$$

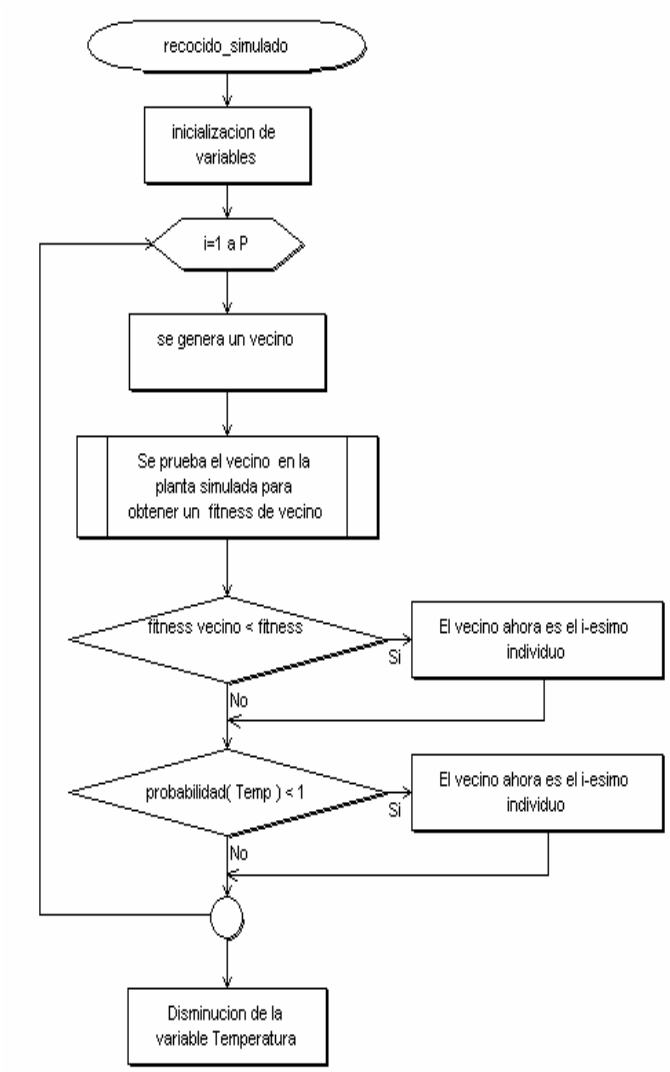

Figura 2. Algoritmo de recocido, el cual adapta las constantes del control PID.

2.3 RESULTADOS

Se implementaron simulaciones en Matlab del sistema en lazo cerrado de la figura 1, donde el bloque recocido simulado es como se muestra en la figura 2, se utilizo como planta un integrador:

$$G(s) = \frac{1}{s} \qquad (5)$$

que en la muestra 1800 (100ms de muestreo) se convierte en un planta de primer orden (ecuación 6), esto con el fin de probar la efectividad del recocido simulado:

$$G(s) = \frac{1}{s + 0.2} \qquad (6)$$

En la figura 3, se muestra el instante donde se cambia la planta (muestra 1800), la señal continua es la que se obtuvo del PID calculado por el algoritmo genético, con

una población de 100 individuos y 800 generaciones aplicado sobre el integrador (ecuación 5).

Para el recocido se utilizo una población de 10 individuos, los cuales se tomaron de la población resultado del algoritmo genético, una temperatura inicial de 1000 y un coeficiente de enfriamiento β de 0.95.

A cada individuo se le permite controlar la planta durante 10 muestras para poder calcular un "fitness" con las ecuaciones 1 y 2. Al ser 10 individuos, es necesario 100 muestras para poder llamar la función recocido (Figura 2).

En la figura 4, se muestran las últimas 1000 muestras y como el recocido mejora el tiempo de respuesta (línea punteada).

Para obtener la figura 5, se utilizo la ecuación:

$$se400 = \sum_{k'=k}^{k+400} e^2(k') \qquad (7)$$

La cual calcula la sumatoria del error cada 400 muestras, La grafica es para 16,000 muestras, pero la respuesta básicamente mejora a partir de las 8000 muestras (figura 5).

En la figura 6, se muestra la grafica para las primeras muestras del control de velocidad de un motor dc usando el programa Builder c++, para un tiempo de muestreo de 50ms.

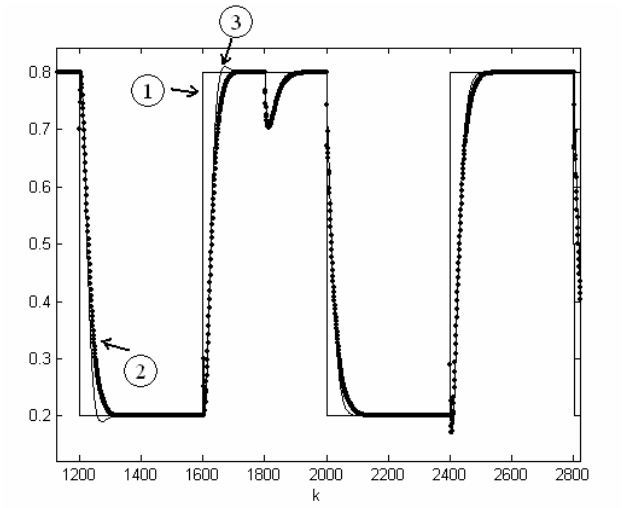

Figura 3. Señal de referencia (***1***), control pid con SA (***2***), control pid normal (***3***) ante una variación de la planta en la muestra 1800.

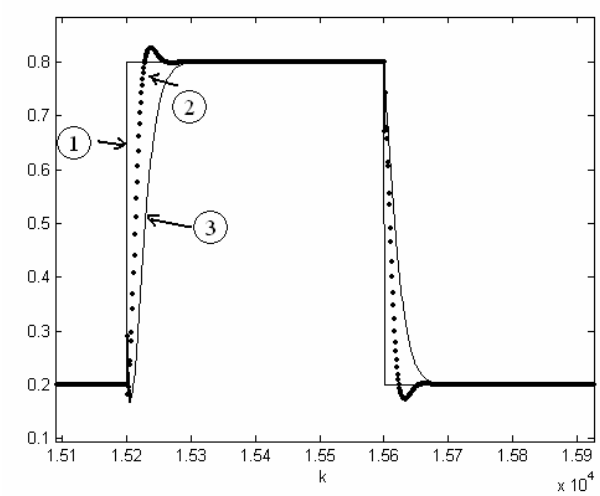

Figura 4. Señal de referencia (***1***), control pid con SA (***2***), control pid normal (***3***) para las últimas muestras, la línea punteada es la señal del PID con SA.

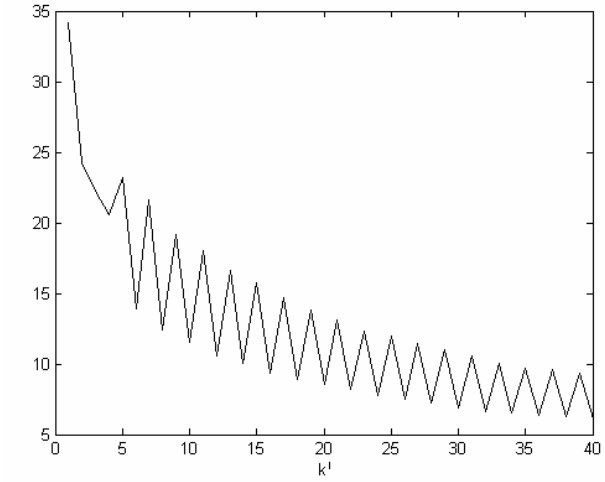

Figura 5. Sumatoria del error (7) tomado cada 400 muestras.

3. CONCLUSIONES Y RECOMENDACIONES

La técnica de recocido simulado logra mejorar la respuesta del sistema, dando tiempos de respuesta más cortos.

El recocido simulado también se probó solo o sea sin pre-entrenamiento con genéticos, pero no funciono bien y se notaban muchas oscilaciones.

A futuro se piensa agregarle un sistema que identifique la planta, y así no es necesario el modelo de la planta para poder calcular el 'fitness' del vecino.

También se continúa con pruebas para un sistema de bola balancín, ya que al momento de entrega de este artículo no se han logrado resultados aceptables.

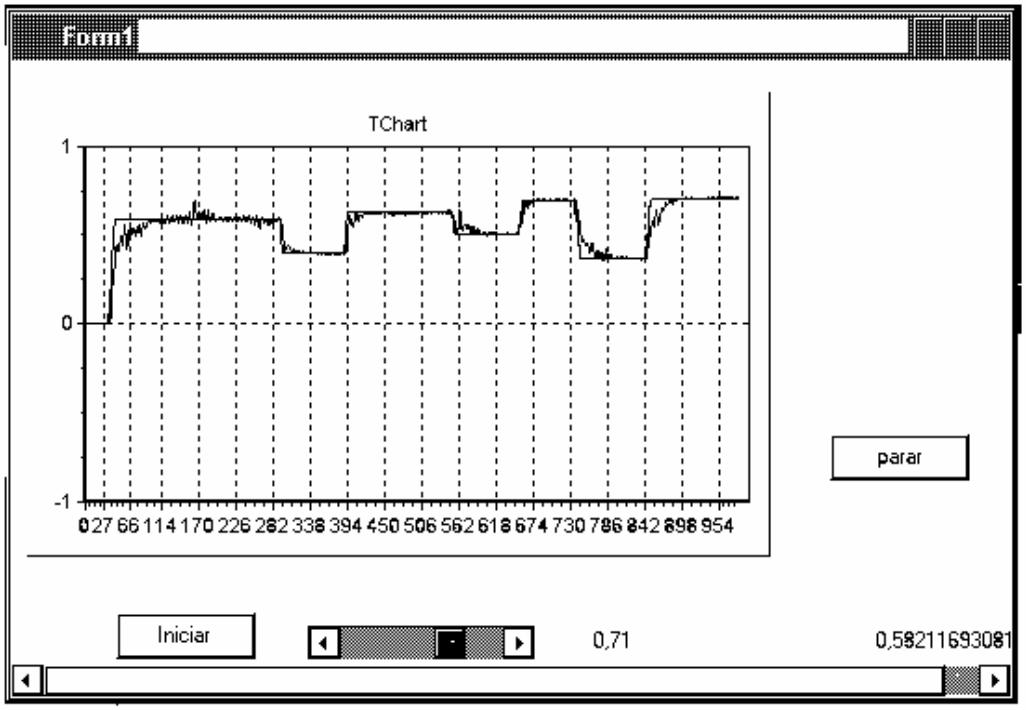

Figura 6. Graficas para el control de la velocidad de un motor dc para las primeras muestras.

4. BIBLIOGRAFÍA

[1] RESTREPO Héctor F., PEÑA Carlos A., PEREZ Andrés., Hacia el desarrollo de nuevas máquinas computacionales, Energía y Computación, Univalle, octubre del 2000.

[2] DELGADO Alberto, Inteligencia Artificial y Minirobots, ECOE ediciones, Bogota-Colombia 1998.

[3] CHIABERGE M., MERELO J.J. y otros, A Comparison of Neural Networks, Linear Controllers, Genetic Algorithms and Simulated Annealing for Real Time Control, Dipartimento di Elettronica, Politecnico di Torino - Italia y Departamento de Electrónica y Tecnología de Computadores, Universidad de Granada – España, 2004.

[4] HERNÁNDEZ Maria del C., Los algoritmos genéticos en el ajuste óptimo de reguladores, Energía y Computación, Univalle, Vol. VI, No 1, ed. 12, 1997.

[5] SALAZAR Harold, y otros, Entrenamiento de una red neuronal artificial usando el algoritmo de simulated annealing, Sci. Tech., Año X, No.24, UTP, mayo 2004.

[6] HOYOS José Gabriel, IBARGUEN Francisco, CARDONA Jaiber, Control de un Péndulo Invertido Rotacional por Realimentación de Espacio de Estados Generado a Través de Algoritmos Genéticos, Sci. Tech., Año XII, No.32, UTP, diciembre del 2006.

[7] LI, Yun y otros, Performance Based Linear Control System Design by Genetic Evolution with Simulated Annealing, Centre for systems and control, University of Glasgow, United Kingdom, 1995.

[8] FRANCO John F., TABARES Pompilio, Aplicación del Simulated Annealing al Problema de las N Reinas, Sci. Tech., Año XI, No 29, UTP, Diciembre de 2005.