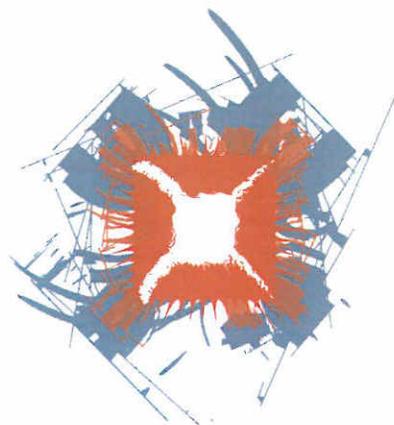


THOMSON

CEDI 2005

I CONGRESO ESPAÑOL DE INFORMÁTICA
GRANADA DEL 13 AL 16 DE SEPTIEMBRE



V Jornadas de Computación Reconfigurable y Aplicaciones

[JCRA'2005]

EDITORES

Sergio Cuenca Asensi • Francisco José Pelayo Valle

```
echo "Parte de PHP <br>";  
for ($i 0; $i<10;$i++)  
{  
echo "linea". $i<10;$i."<br>";  
}
```

```
<?php  
echo  
for (  
{  
echo
```

CONGRESOS

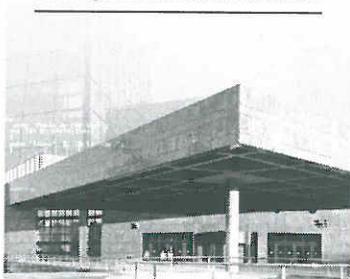
PALACIO

Nuevos retos científicos y tecnológicos en Ingeniería Informática

I CONGRESO ESPAÑOL
DE INFORMÁTICA

CEDI 2005

Nuevos retos
científicos y tecnológicos
en Ingeniería Informática



ACTAS DE LAS
**V Jornadas de Computación
Reconfigurable y Aplicaciones**
[JCRA'2005]

EDITORES

Sergio Cuenca Asensi
Francisco José Pelayo Valle

JORNADAS ORGANIZADAS POR

Dpto. de Tecnología Informática y Computación, Universidad de Alicante
Dpto. de Arquitectura y Tecnología de Computadores, Universidad de Granada

ENTIDADES COLABORADORAS

Vicerrectorado de Investigación de la Universidad de Granada
Consejería de Innovación, Ciencia y Empresa; Junta de Andalucía

THOMSON



TM



Actas de las V Jornadas de Computación
Reconfigurable y Aplicaciones [JCRA'2005]

© Los Autores



Editores de la serie de Actas del CEDI

Rafael Molina Soriano
Antonio Díaz García
Alberto Prieto Espinosa

Editores de las Actas de las presentes Jornadas

Sergio Cuenca Asensi
Francisco José Pelayo Valle

Diseño de Cubiertas



www.dixi-e.com

Impresión

THOMSON

COPYRIGHT© 2005 International
Thomson Editores Spain
Paraninfo, S.A.
Magallanes 25 · 28015 Madrid España
Tel: 91 446 33 50 · Fax: 91 445 62 18
clientes@paraninfo.es

Impreso en España

Printed in Spain

ISBN: 84-9732-439-0
Depósito legal: SE-4072-2005 European Union
Printed by Publidisa

Reservados todos los derechos para todos los países de lengua española. De conformidad con lo dispuesto en el artículo 270 del código penal vigente, podrán ser castigados con penas de multa y privación de libertad quienes reprodujeran o plagiaran, en todo o en parte, una obra literaria, artística o científica fijada en cualquier tipo de soporte sin la preceptiva autorización.

Ninguna parte de esta publicación, incluido el diseño de la cubierta, puede ser reproducida, almacenada o transmitida de ninguna forma, ni por ningún medio, sea éste electrónico, químico, electro-óptico, grabación, fotocopia o cualquier otro, sin la previa autorización escrita por parte de los autores.

Coprocesador para Operaciones en un Campo Finito.....	283
Giuseppe Forte, Jean-Pierre Deschamps	
<i>Universitat Rovira i Virgili</i>	
Optimización de Algoritmos de Cifrado en Soft Core Processors.....	291
Francisco Gómez Arribas, Iván González	
<i>Universidad Autónoma de Madrid</i>	
Control y robótica.....	297
Vehículo-Robot para Aplicaciones de Control y Visión Artificial.....	299
Luis Carranza González, Elisenda Roca, Angel Rodríguez-Vázquez	
<i>IMSE-CNM-CSIC, Sevilla</i>	
Sistema de Transporte autónomo con GPS.....	305
Ángel Grediaga, Pablo Baeza Nadal, Sergio Cuenca Asensi, Francisco Ibarra Picó, Bernardo Ledesma	
<i>Universidad Alicante</i>	
Sistema evolutivo de control de un brazo robot: Optimización de constantes dinámicas.....	311
Rodrigo Agís Melero, Eduardo Ros Vidal, Francisco José Pelayo Valle, Richard Carrillo Sánchez, Eva Ortigosa, Rafael Rodríguez	
<i>Universidad de Granada</i>	
Implementación hardware basada en FPGAs para el control de movimiento de un tren digital mediante técnicas de procesamiento de imágenes.....	317
Juan Suardíaz, Basil M. Al-Hadithi, Andrés Iborra García	
<i>Univ. Politécnica de Cartagena & Univ. Alfonso X</i>	

Sistema evolutivo de control de un brazo robot: Optimización de constantes dinámicas.

Rodrigo Agis, Eduardo Ros, F.J. Pelayo, Richard Carrillo, Eva Ortigosa, Rafael Rodríguez, Javier Díaz, Sonia Mota

Dept. Arq. y Tec. de Computadores, E.T.S.I. Informática, U. Granada, 18071, España
{ragis, eros, fpelayo, rcarrillo, eva, rrodriguez, jdiaz, smota}@atc.ugr.es.

Resumen

El estudio de esquemas de control que permitan a un sistema mantenerse estable frente a perturbaciones externas ha dado como resultado la creación de reguladores muy sofisticados. Entre ellos, el más utilizado es el controlador PID (Proporcional Integrador Derivativo). Ciertamente el funcionamiento de los reguladores PID, una vez ajustadas sus constantes proporcional, integral y derivativa, es bastante bueno, sin embargo el ajuste de estas constantes no es una tarea sencilla en la práctica, sobre todo cuando el elemento actuador es complejo. En este artículo presentamos la implementación real de un sistema evolutivo de optimización de constantes para un controlador PID basado en algoritmos genéticos. El objetivo principal, a medio plazo, consiste en estudiar la versatilidad y viabilidad del aprendizaje automático de la dinámica interna en robots complejos. Para ello se ha utilizado un brazo robot con motores analógicos con articulaciones no rígidas y un sistema completo de control implementado en un solo dispositivo FPGA.

1. Introducción

En robótica existen multitud de esquemas de control adaptativo sin embargo, la mayoría de ellos precalculan las ganancias o parámetros de ajuste en una fase previa de diseño con modelos simulados. Los parámetros quedan fijados a intervalos de forma que se utilicen unos u otros (los más apropiados) en función de un criterio de selección, por ejemplo; el punto del espacio de trabajo donde se mueva el robot. La implementación más convencional es la de una

tabla en la cual las entradas son intervalos discretizados de las variables, que determinan la pertenencia a una región u otra del espacio de trabajo, mientras que las salidas son los valores de las ganancias en estos intervalos. Las variables más empleadas para ajustar las ganancias en el control de robots manipuladores son las variables articulares y la carga. Nótese que el procedimiento para ajustar los parámetros de los controladores PID de las articulaciones puede ser muy laborioso puesto que, en general, se necesitan ajustar $3n$ parámetros K_p , K_i , K_d , siendo n el número de articulaciones. Por consiguiente, el número de ensayos que es necesario realizar depende de la resolución que se desee. En cualquier caso si no se dispone del modelo dinámico del sistema robótico por su enorme complejidad, es necesario realizar una gran cantidad de pruebas. Una posible solución consiste en ajustar las ganancias (parámetros del PID) de acuerdo a una ley de adaptación que intenta minimizar (normalmente el error cometido) de acuerdo con un modelo de referencia [1]. El modelo de referencia es una aproximación que describe como se desea que se comporte el sistema mientras que la técnica de ajuste modela las constantes del controlador PID de acuerdo a un objetivo (función de optimización). Véase Figura 1.

La idea de esta experiencia consiste en intentar ajustar las constantes del controlador PID para que sean óptimas para una trayectoria predefinida, haciendo que el error cometido y el consumo de energía sean los mínimos posibles (control adaptativo dinámico). Para el ajuste de estas constantes se ha utilizado un algoritmo genético simple, el cual codifica mediante individuos las constantes para los controladores PID. Véase Figura 2. Normalmente, los robots

convencionales sólo utilizan modelos cinemáticos ya que pueden realizar tareas de control en ciclo cerrado (sensor-actuador). Pero esto significa un consumo enorme de energía que hace esta aproximación poco viable para robots autónomos. Además, si las constantes inerciales de los elementos a manipular son significativas con respecto a la fuerza que son capaces de realizar, el control será muy deficiente.

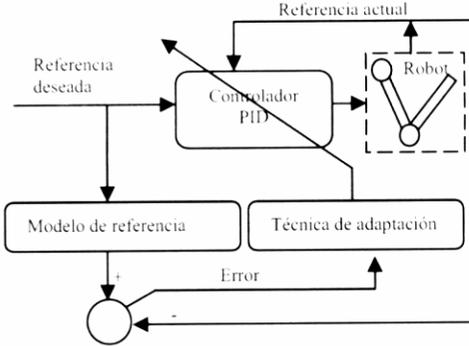


Figura 1. Control por modelo de referencia.

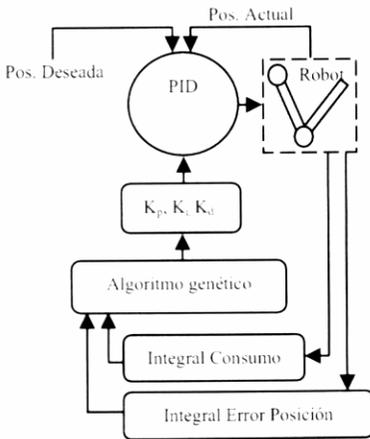


Figura 2. Optimización de constantes dinámicas mediante algoritmos genéticos.

2. Plataforma de experimentación

Para realizar este estudio se utilizó una plataforma robótica experimental formada por un brazo robótico de articulaciones con inercia (no rígidas) de dos grados de libertad.

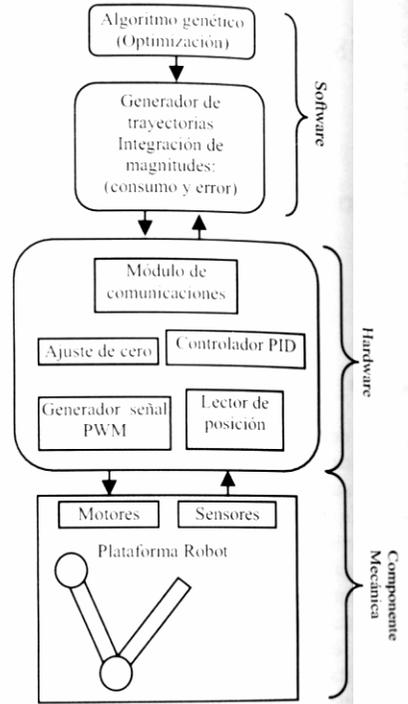


Figura 3. Arquitectura hardware software del sistema.

Módulos	% Slices
Comunicaciones	5.64
Lector de posición	0.97
Controlador PID	0.99
Generador PWM	3.78
TOTAL	11.38

Tabla 1. Porcentaje de ocupación realizada con compilaciones parciales (aislamiento parcial de módulos). Software de desarrollo de síntesis DK-2 [2] e ISE-5.li [3]. N° de slices de la FPGA Virtex II: 5120.

El sistema dispone de propioceptores de posición ópticos "encoders" así como un sistema de visión (localizador de coordenadas del extremo del brazo). Todo el procesamiento visual así como el sistema de control de motores, es local y han sido implementados en un dispositivo FPGA. Además, el sistema dispone de un conjunto de librerías software que permiten la configuración y control del robot, mediante un PC [4]. Nuestro sistema se compone de módulos implementados en software (actualmente en un PC) y módulos implementados en hardware (FPGA), véase Figura 3. En un

futuro, los módulos de optimización software se pueden transferir a un procesador embebido para tener un sistema independiente. La tabla 1 muestra el porcentaje de ocupación del dispositivo FPGA para la parte hardware de control. Observar que el consumo total de "slices" es muy bajo, permitiendo ser implementado en dispositivos FPGA de bajo coste.

3. Algoritmo genético

La funcionalidad del algoritmo genético consiste en generar las constantes K_p , K_i y K_d del controlador PID y mejorarlas, en cada iteración, intentando minimizar el consumo y el error de posición integrado de una trayectoria. Por tanto las funciones objetivo a minimizar son el *error de posición*, y el *consumo*. Esto trae consigo un problema a la hora de asignar una valoración concreta a un individuo "fitness" porque una solución con consumo próximo a cero y error medio pequeño puede ser etiquetada como óptima y sin embargo el robot no se mueve. La solución pasa por establecer una política multiobjetivo para el algoritmo o bien definir la función objetivo filtrando estas inconveniencias. Nosotros hemos adoptado esta última opción. La población está formada por un conjunto de individuos constituidos por los tres parámetros K_p , K_i , K_d , que conforman los cromosomas. Los individuos se cruzan entre sí en función de sus cualidades "fitness" e intercambia su material genético. Hemos implementado un algoritmo genético sencillo [5]. Al tener que realizar "movimientos reales" en cada iteración para cada individuo, la evolución del sistema es lenta. El cruce se realiza generando un valor aleatorio que indicará que trozo de material genético se intercambia. Por ejemplo; Suponer que un individuo es el 0101 y el siguiente mejor es el 00010. Ahora suponer que el valor aleatorio de cruce es el 0100 entonces daría como resultado dos nuevos individuos, que son 0001 y el 0110. Utilizando este esquema de cruce es obvio que solamente los $n/2$ individuos mejores se cruzan dando lugar a otros $n/2$ nuevos individuos y los $n/2$ peores de la lista inicial, se desechan. Para comenzar la competición se genera una población de individuos (en este caso 8) con valores de cromosomas aleatorios pero dentro de un rango máximo admisible. Esto es; max para K_p = 1024, max para K_i 50, Max para K_d = 1024.

Posteriormente se siguen los siguientes pasos que se repiten un número de veces predefinido (generaciones):

1. Evaluar las cualidades de cada individuo (selección natural). Esto consiste en configurar el controlador PID con los cromosomas de cada individuo y evaluar la trayectoria previamente definida midiendo el consumo y error totales. Esta evaluación supone la activación del robot para realizar la trayectoria.
2. Ordenar los individuos según la puntuación obtenida (función de minimización de error de posición y consumo). Esto se consigue ordenando los individuos de mayor a menor "fitness" utilizando el algoritmo "Quick Short" [6] y permitiendo reproducirse solamente a los primeros de la lista.
3. Permitir a cada uno de los individuos reproducirse (intercambien material genético) de acuerdo con su puntuación "fitness". Además se permite con una probabilidad (P) que alguno de los bits de un gen se vea alterado espontáneamente.

4. Proceso de evaluación de individuos

El proceso de evaluación parte con una trayectoria predefinida formada por un conjunto de posiciones deseadas obtenidas mediante programación gestual. Seguidamente se ajustan las constantes del PID con los valores del individuo a evaluar y se comienza el envío de coordenadas objetivo hacia el robot a intervalos de 10 ms. En cada envío se lee del robot el consumo y el error de posición de la coordenada enviada en el instante anterior (inicialmente 0).

Para calcular el *consumo*, Ecuación 1, y el *error*, Ecuación 2, se calculan las integrales discretas a lo largo de toda la trayectoria.

$$Err = \int_0^{kT} c(t)dt \approx \sum_{j=1}^{k-1} \frac{c(j) \cdot c(j-1)}{2} \quad (1)$$

$$Con = \int_0^{kT} e(t)dt \approx \sum_{j=1}^{k-1} \frac{|e(j)| + |e(j-1)|}{2} \quad (2)$$

Con estas dos magnitudes, el algoritmo genético evalúa la calidad de la solución y establece la clasificación entre individuos peores y mejores de acuerdo a la función de minimización.

5. Funciones de evaluación de error y consumo

Para minimizar el error y el consumo se han evaluado tres funciones distintas (A, B, C) que se ilustran seguidamente.

5.1. Función de evaluación "fitness" (A)

La primera función de evaluación consta de dos partes ponderadas (Ecuación 3): La integral del error Err y la integral del consumo Con . Estas dos partes se ponderan y suman para dar una valoración final a un individuo que el algoritmo intentará hacer máxima (en este caso un individuo es mejor cuanto menor es el valor del error y el consumo).

Mediante el ajuste de los valores de ponderación w se puede dar mayor o menor importancia al error o al consumo de la trayectoria.

$$F = 1 - (w * Con + (1 - w) * Err) \quad (3)$$

La utilización de esta función de evaluación conlleva implícitamente la posibilidad de soluciones óptimas en lo referente al proceso de búsqueda del algoritmo genético, pero inútiles en la práctica. Porque es posible dar como buena una solución con un consumo muy próximo a cero y un error aceptable (suponer que el brazo se encuentra en un punto medio de una trayectoria circular). Esta solución haría que el robot no se moviese porque no hay un consumo suficiente. Por consiguiente este tipo de soluciones no son buenas ya que interesa que la valoración de la solución quede comprendida lo más cerca posible del máximo absoluto pero compensada entre ambos parámetros (error y consumo). Es decir, la solución óptima estará en un intervalo que establezca un compromiso entre un consumo y un error admisible del plano de soluciones posibles.

5.2. Función de evaluación "fitness" (B)

En esta función de evaluación se intenta solucionar el problema comentado anteriormente mediante la definición de un escalón en la frontera de valores admisibles. Este escalón establece un umbral mínimo para el consumo. Todas las

soluciones que tengan un consumo inferior a cierto umbral serán penalizadas. La elección del umbral conlleva un compromiso puesto que si la trayectoria predefinida se puede realizar con un consumo menor el algoritmo intentará alcanzar dicho consumo y penalizará una solución que a priori era mejor. El valor óptimo del umbral de consumo depende por tanto de la trayectoria predefinida y de la dinámica del robot (suponer que coge algún peso). Como estos parámetros son a priori desconocidos, una solución práctica consiste en ajustar el umbral de consumo a un valor lo suficientemente pequeño como para permitir que el robot se mueva sin carga (sin coger pesos) y, de este modo, no caer en el problema descrito.

A continuación se muestra una posible implementación muy sencilla del algoritmo de evaluación.

```

si consumo < 250 mA
entonces fitness = 0.0;
si no
entonces fitness = F(Con) + F(Err);

```

$$F(Con) = e^{\left(\frac{-6 * Con(n)}{2 * pos_size} * \log(0.5) + 0.75\right)} \quad (4)$$

$$F(Err) = e^{\left(\frac{-6 * Err(n)}{4096 * pos_size} + \log(0.5)\right)} \quad (5)$$

La función fitness descrita en las ecuaciones 4 y 5 son de tipo exponencial. De este modo se obtiene una mayor velocidad de convergencia hacia la solución.

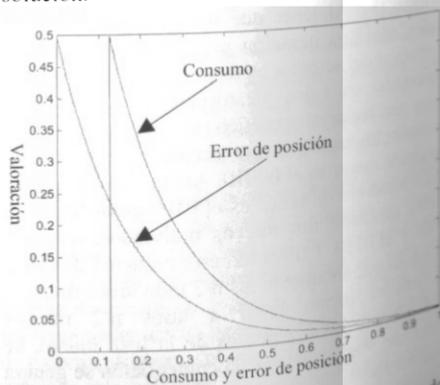


Figura 4. Funciones de evaluación; consumo y error de posición normalizado [0, 1].

Esto es crítico ya que el tiempo de evaluación es largo. Además la forma exponencial permite una mayor flexibilidad en la valoración de los individuos en etapas iniciales del algoritmo. De esta forma se puede elegir secuencias de soluciones a priori poco prometedoras pero buenas a largo plazo. En etapas finales, cuando el conjunto de las soluciones son más o menos parecidos, la discriminación entre individuos es mucho mayor (gracias a la pendiente exponencial del plano de soluciones) permitiendo una tendencia más elitista del algoritmo. Observar que tanto en $F(Err)$ como en $F(Con)$ ecuaciones 4 y 5, se realiza una operación de escalado para que los valores introducidos en la función exponencial estén comprendidos entre 0 y 1. Este escalado se hace para el caso de la integral del consumo dividiendo por $2 * n^n$ de puntos de la trayectoria. El valor 2 es porque el consumo máximo es de 2 amperios.

Para el caso del error se divide por $4096 * n^n$ de puntos porque el error máximo posible es de 4096 ppr (Pasos Por Revolución) [4]. La ponderación para el consumo y el error esta integrada implícitamente en las propias funciones de valoración. Ambas funciones dan como resultado valores comprendidos en el intervalo [0, 0.5] de forma que al sumarlas darán valoraciones entre 0 y 1, véase Figura 4.

5.3. Función de evaluación "fitness" (C)

La función "fitnes" para este caso es igual a la descrita en el apartado 5.2 solamente que ahora la valoración de un individuo para el consumo se hace igual a 0.5 si está comprendido entre 250 y 500 mA. Esta característica da mayor flexibilidad permitiendo mantener buenos individuos que producen trayectorias con consumos medios dejando la elección final del mejor individuo a la otra componente de la función, el error de posición.

La modificación del algoritmo se ilustra en las siguientes líneas de código:

```

81 250 mA <= consumo <= 500 mA
82 entonces fitness = 0.5;
83 no
84 entonces fitness = (Con)+F(Err);
    
```

6. Resultados experimentales

Los experimentos realizados con las diferentes funciones de evaluación fueron tres y se corresponden con las funciones mostradas en la figura 5 y 6.

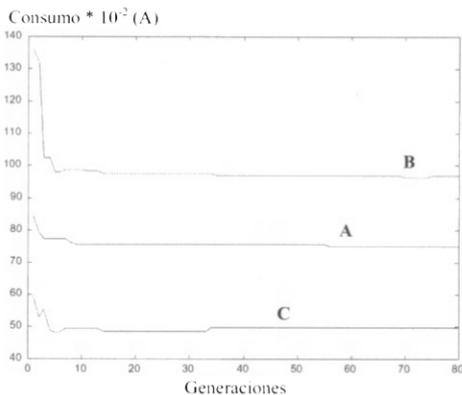


Figura 5. Curvas de consumo obtenidas con las funciones de evaluación A, B, C.

Los parámetros de configuración utilizados para todos los experimentos fueron:

- N° individuos: 8
- N° de generaciones: 80
- Trayectoria "dibujar un 9" de 149 puntos
- Periodo de muestreo 10 ms

Obsérvese que el número de individuos e iteraciones es muy reducido ya que el proceso de evolución es muy lento. Las curvas de consumo acumulado para la trayectoria definida se ilustran en la Figura 5. En ella se observa la evolución del algoritmo genético a lo largo de 80 generaciones para las tres funciones de evaluación. El consumo mínimo acumulado en un ajuste manual de constantes [7] fue de $43.1 * 10^{-2}$ Amperios mientras que el consumo mínimo obtenido por el algoritmo fue aproximadamente de $50 * 10^{-2}$ Amperios. Esta diferencia fue compensada por el error de posición.

La Figura 6, muestra la evolución del error de posición acumulado obtenido con las tres funciones de evaluación comentadas. El error mínimo acumulado obtenido mediante ajuste manual de constantes fue de 3450 ppr siguiendo la técnica de ajuste de [7], mientras que el error

acumulado obtenido por el algoritmo fue de 200 ppr.

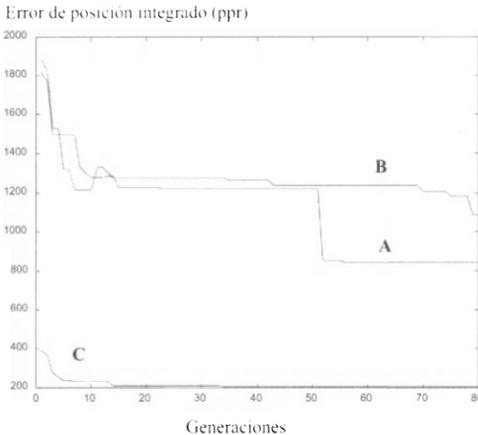


Figura 6. Curvas de error obtenidas con las funciones de evaluación A, B, C.

7. Conclusiones

El objetivo inicial de esta experiencia consistía en intentar que el propio robot aprenda su dinámica interna mediante un proceso evolutivo, que sirve de base de estudio, para sistemas de control adaptativo futuros.

Las conclusiones sobre este artículo se pueden resumir en los siguientes puntos:

1. El proceso evolutivo es lento y costoso para sistemas físicos porque los motores se calientan al tener que evaluar trayectorias de individuos malos. Quizás un sistema simulado pueda ser útil para la obtención de constantes que más tarde puedan ser afinadas en el modelo real. Además la velocidad de convergencia de la función de evaluación es importante.
2. La definición de la función de evaluación es un punto crítico en cuanto a los umbrales de consumo y elección de pesos para los distintos objetivos. Esto es debido a que el robot tiene multitud de parámetros electromecánicos difíciles de reflejar explícitamente en la función de evaluación.
3. Los experimentos realizados con el robot a priori muestran que solamente con una

magnitud, como por ejemplo el error de posición, no se obtienen soluciones buenas para las constantes del controlador. Por ejemplo la constante integral está íntimamente relacionada con el consumo y es preciso disponer de algún parámetro en la función de evaluación que implique esta magnitud.

4. Variación en las mediciones de los parámetros de error y consumo con la temperatura. Un mismo individuo puede ser evaluado de forma diferente como consecuencia de modificaciones de rendimiento ocasionadas por variaciones térmicas en los motores. Este tipo de factores son difíciles de simular y justifican experimentalmente el trabajo con un robot real.
5. Se ha observado que entre 20 y 80 generaciones (dependiendo de la función de evaluación) para una población inicial de 8 individuos se obtiene un ajuste más fino que el realizado manualmente.

Agradecimientos

Este trabajo ha sido financiado por el proyecto europeo SpikeFORCE (IST-2001-35271) y el proyecto nacional DEPROVI (DPI 2004-07032).

Referencias

- [1] Robótica manipuladores y robots móviles. Anibal Ollero. Capítulos 5 y 8. ISBN: 84-267-131-30
- [2] Celoxica, <http://www.celoxica.com>
- [3] Xilinx, <http://www.xilinx.com>
- [4] Agis R., Ros E., Díaz J., Mota S., Carrillo R., Ortigosa E., Pelayo F., Prieto A., Sistema de control basado en visión y propiocepción de robots con FPGA. JCRA, pp. 667-674. 13-15 de Septiembre, 2004.
- [5] Informática evolutiva: Algoritmos Genéticos. J. Julián Merelo. <http://geneura.ugr.es/~jmerelo/ie/ags.html>
- [6] Hoare, C.A.R. "Quicksort", *Computer Journal*, pp. 10-15, April 1962.
- [7] Ziegler, J.G. and Nichols, N.B. "Optimum settings for automatic controllers", *Trans. ASME*, 1942, 65, pp. 433-444.