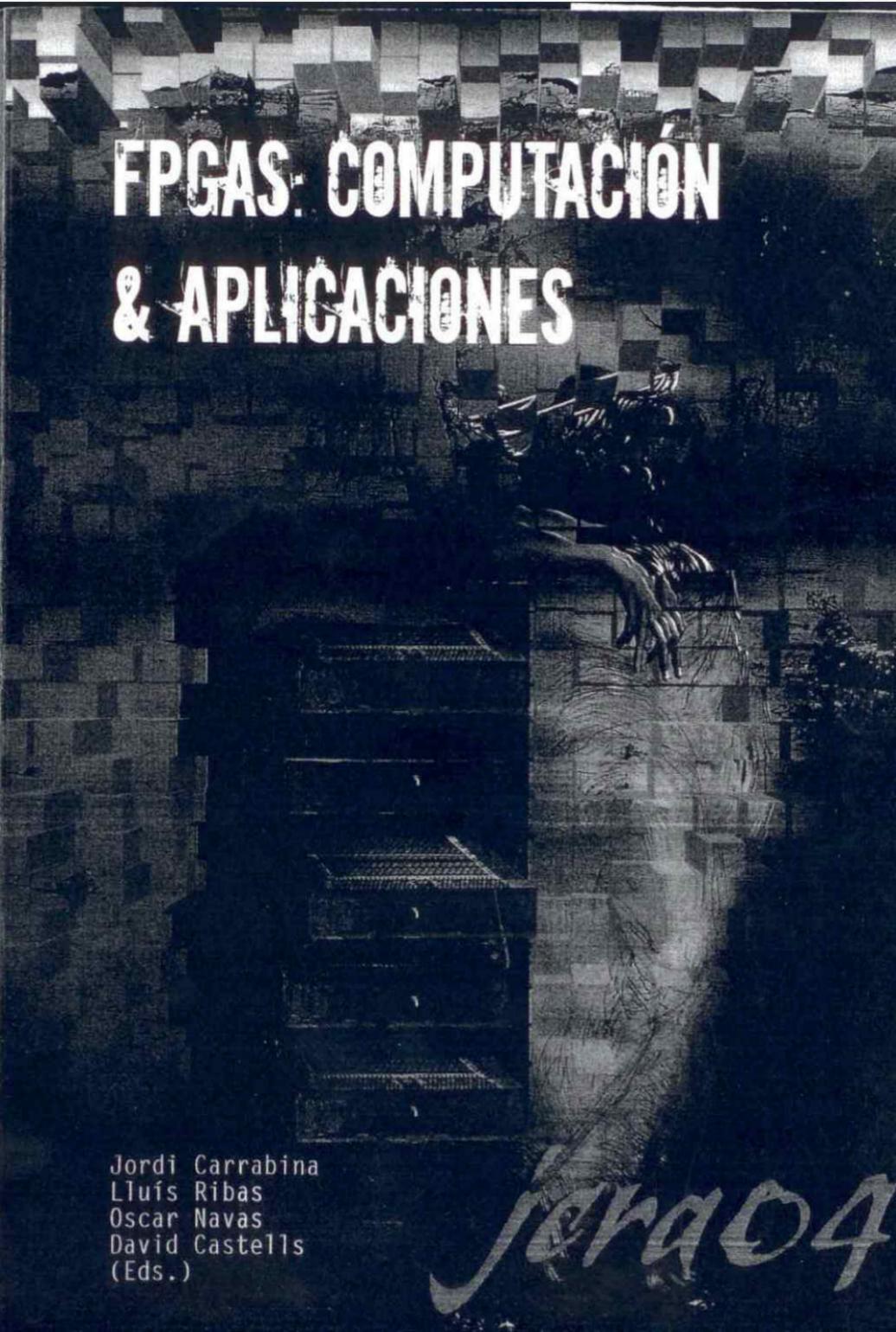


FPGAs: COMPUTACIÓN & APLICACIONES

The background of the cover is a dark, textured surface with a grid-like pattern, possibly representing a circuit board or a digital landscape. A hand is visible, holding a glowing, rectangular object that appears to be a component or a chip. The overall aesthetic is technical and futuristic.

Jordi Carrabina
Lluís Ribas
Oscar Navas
David Castells
(Eds.)

serva04

FPGAs: COMPUTACIÓN & APLICACIONES

*Artículos seleccionados de las IV Jornadas de
Computación Reconfigurable y Aplicaciones
Bellaterra, 13-15 de Septiembre de 2004*

*Jordi Carrabina
Lluís Ribas
Oscar Navas
David Castells*

Editores

**Centre de Prototips i Solucions HW/SW
Escola Tècnica Superior d'Enginyeries
Universitat Autònoma de Barcelona**

Título: FPGAs: COMPUTACIÓN & APLICACIONES

I.S.B.N.: 84-688-7667-4

Depósito Legal: 40.194-2004

Editores: J. Carrabina, Ll. Ribas, O. Navas, D. Castells
Centre de Prototips i Solucions HW/SW
Escola Tècnica Superior d'Enginyeries
Universitat Autònoma de Barcelona
08193 Bellaterra
cephis@cephis.uab.es
<http://cephis.uab.es>

Impresión: Format Digital S.L.
C/ Berlin 11-13, Baixos
08014 Barcelona
Tel: 934090005
formatdigital@telefonica.net

Reservados todos los derechos. Ni la totalidad ni parte de este libro puede reproducirse o transmitirse por ningún procedimiento electrónico o mecánico, incluyendo fotocopia, grabación magnética o cualquier almacenamiento de información o sistema de reproducción, sin permiso previo y por escrito de los titulares del Copyright.

9.1 Aplicaciones

- Módulo de Clasificación de Datos con Lógica de Control
Mínima.
Castells D., Monton, M., Ribas Ll, Carrabina, J......651
- A PSoC Component for Frequency Based Acquisition
Applications.
Aragonés R., Ferrer C., Oliver J. 661

9.2 Robots

- Sistema de Control Basado en Visión y Propiacepción de
Robots con FPGA.
*Agís R., Ros E., Díaz J., Mota S., Carrillo R.,
Ortigosa E., Pelayo F., Prieto A.*..... 677
- Locomoción de un Robot Ápodo Modular con el
Procesador MicroBlaze.
González J., Aguayo E. y Boemo E.675

Capítulo 10: Experiencias educativas683

- Laboratorio Remoto de Estructuras de Computadores
basado en Hardware Reconfigurable.
Grediaga A., Cuenca S., Colom J.F, Durá J.A......685
- Experiencias Docentes con FPGA's. Caso práctico: Diseño
de un Videojuego.
*Olivares J., Bravo E., Gómez L., Aguirre M.A., Tombs J.N., Muñoz F.,
Baena V., Torralba A.*.....695
- Experiencias Docentes con el Sistema NIOS.
Pastor E., López J......703
- Dispositivos y Plataformas para Docencia
de Informática y Electrónica.
Talavera G., Saiz J., Carrabina J.711

Sistema de control basado en visión y propiocepción de robots con FPGA.

Agís R, Ros E, Díaz J, Mota S, Carrillo R, Ortigosa E, Pelayo F, Prieto A.

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

Resumen. En este artículo presentamos la implementación completa de un sistema de control de posición para un brazo robótico de articulaciones con inercia (no rígidas) de dos grados de libertad. 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, basado en controladores PID (Proporcional Integrador Derivativo) es local y han sido implementados en plataforma de hardware reconfigurable que permite la evaluación y configuración del dispositivo de forma rápida. Además, el sistema dispone de un conjunto de librerías software que permiten la configuración y control del robot, mediante un PC. El objetivo principal, a medio plazo, es valorar la versatilidad y viabilidad del aprendizaje mediante redes neuronales de pulsos, de la propia dinámica interna del robot que supondría un importante avance al eliminar la necesidad de diseños estructurales bien definidos.

1. Introducción

Este trabajo ha sido desarrollado en el contexto de SpikeFORCE [1] y ECOVISION [2]. SpikeForce y ECOVISION son consorcios formados por investigadores con experiencia en la construcción de sistemas bioinspirados, tratamiento de imágenes por computador, visión y su aplicación en el campo de la robótica y automática. 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 más o menos sofisticados. Entre ellos, el más utilizado es el controlador PID. 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 [3,4], sobre todo cuando el elemento actuador es un elemento compuesto. Pensemos en el caso de un brazo robot con cinco o seis articulaciones controladas independientemente por un controlador local para cada una. Las acciones de control de una articulación influyen irremediabilmente sobre las otras produciendo sobre-oscilaciones e inestabilidad en el robot. El problema se agrava aún más cuando se modifica la dinámica del sistema. Por ejemplo; supongamos que el brazo de seis articulaciones coge un objeto de un determinado peso. En ese instante todos los ajustes de constantes que definían la dinámica del brazo y por consiguiente la acción

de control quedan desfasados. Para solucionar este problema proponemos una alternativa bioinspirada. En este caso se trata de aprender la propia dinámica del sistema. Imaginemos que el brazo comentado anteriormente fuese capaz de recordar y aprender los parámetros de control y aplicar unos u otros en función de la modificación de su dinámica (coger un peso mayor o menor). El resultado sería un sistema muy robusto, versátil y extrapolable a otros diseños estructurales muy complejos, como por ejemplo robots antropomórficos. Además, la explotación de la dinámica interna del robot conlleva un consumo energético óptimo. Siguiendo aún más allá, el sistema no sólo sería capaz de aprender su dinámica sino, con la inclusión de sistemas neuronales más completos, la predicción de trayectorias o secuencias de movimientos rutinarios (que se convierten en primitivas de movimiento complejas) como por ejemplo; la secuencia de movimientos necesaria para andar en el caso de robots bípedos. Indispensable para conseguir el objetivo comentado, es el estudio de redes neuronales de pulso [1, 5, 6] bioinspiradas y más concretamente el análisis de la estructura y funcionamiento del cerebelo biológico; puesto que este centro nervioso está altamente involucrado en las tareas de coordinación de movimientos [1]. El sistema que proponemos constituye la base de desarrollo para la investigación en este campo puesto que combina como métodos propioceptores; la visión y la percepción de las posiciones de las articulaciones, y como sistemas actuadores un conjunto de motores con una dinámica de funcionamiento con inercias (no rígido).

2. Descripción del sistema

El sistema desarrollado está compuesto por los siguientes elementos: Una librería de funciones de alto nivel software de control para PC. Una tarjeta de prototipado estándar de hardware reconfigurable. Una tarjeta interfaz de control de motores diseñada a medida, una mini cámara a color CMOS y el sistema electromecánico o brazo robot de dos grados de libertad (véase Fig. 1).

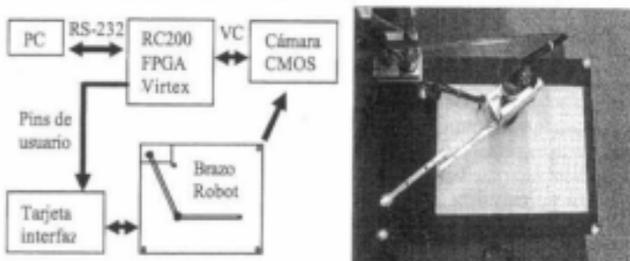


Fig. 1. Arquitectura general: Se observa los diferentes bloques físicos que componen el sistema (izquierda). Brazo robot (derecha).

La **tarjeta de interfaz** realiza la tarea de interconexión física entre el brazo robot (sensores y actuadores) con la plataforma reconfigurable. Esta tarjeta está realizada a

medida y consta de dos controladores de potencia modelo LMD18200 de *National Semiconductors* [7] para dirigir los dos motores del brazo (hombro y codo), además realiza la conversión de niveles lógicos TTL proporcionados por los sensores de posición ópticos (*encoders*) a niveles de 3.3 V compatibles con la FPGA. Estos circuitos disponen de tres entradas de control y han sido especialmente desarrollados por el fabricante para el control de motores mediante PWM (*Pulse Width Modulation*) en modo unipolar (signo-magnitud) o en modo antifase (*Locked Antiphase PWM*). La funcionalidad de las tres entradas son: Dirección, Freno y señal PWM. Estas tres entradas son controladas directamente por patillas auxiliares de la FPGA puesto que los niveles lógicos admisibles por este componente permiten trabajar a 3.3 V. La entrada de dirección indica en qué sentido deseamos que gire el motor (horario o antihorario). La entrada de frenado permite frenar el motor con la característica de permitir un frenado regenerativo si se combina con las otras dos señales [8]. Esto es, en el caso de hacer funcionar el sistema con baterías la energía producida por el motor durante su frenado pasaría a cargar las baterías permitiendo un pequeño ahorro en el consumo. Por último la entrada de control PWM permite la conmutación alternativa del motor para regular su velocidad y par de fuerza.

Los **sensores de posición** son de tipo incremental de 1024 ppr (pasos por revolución) con señal de ajuste de cero. Estos sensores disponen de cinco salidas de las cuales se utilizan solamente tres porque dos de ellas son señales complementadas. Estas señales son: Señal de paso por cero (CHN), señal cuadrada de paso (CHA), señal cuadrada de paso desfasada 90 grados con respecto de la anterior (CHB) (Véase Fig. 2). Todas estas señales son cuadradas y producen un cambio de nivel de tensión por cada paso que avanza el eje del codificador. Puesto que la señal (CHB) entregada está desfasada 90 grados con



Fig. 2 A la izquierda: Señales de posición del codificador. CHA (canal A), CHB (canal B) igual que la señal (A) pero desfasada 90 grados, CHN (señal de paso por cero). Derecha: Conjunto sensor de posición y motor.

respecto a (CHA) es posible multiplicar la resolución del dispositivo por cuatro si se cuentan los flancos de ambas señales. De este modo se puede alcanzar una resolución de 4096 ppr en vez de los 1024 ppr. Además es posible conocer el sentido de giro del eje, simplemente viendo qué secuencia de cambios de estado se producen en la señal.

El **sistema reconfigurable**, en el cual se han implementado todos los controladores, ha sido la RC200 de *Ceolosis* [9] que incorpora una FPGA *Virtex-II* [10]. Esta plataforma de desarrollo cuenta con diversos periféricos de entrada/salida. Entre ellos se distinguen E/S de vídeo compuesto, salida VGA, puerto serie, puerto

paralelo, indicadores *leds*, *display* de siete segmentos, etc. Además de un conector de expansión con pines conectados a patillas de la FPGA y que ha sido utilizados para la conexión con el robot. Para el sistema de visión se ha utilizado una **cámara CMOS**, con salida VC (vídeo compuesto) de calidad media en color, con una resolución máxima de 628 x 582 puntos a la cual se le ha añadido un filtro polarizador ajustable que permite disminuir la saturación en presencia de un grado alto de intensidad de luz ambiente.

El **software del PC** consta de una librería de funciones específica para la comunicación a través de puerto serie con la placa de control RC200. Esta librería permite activar y desactivar los controladores locales de la placa, mandar y recibir posiciones de las articulaciones del brazo, mover independientemente cada motor, ajustar las constantes de los controladores y recibir las posiciones ofrecidas por el sistema de visión. Haciendo uso de esta librería se ha desarrollado un sistema de aprendizaje de trayectorias. Básicamente el operador pone el sistema en modo aprendizaje y mueve el brazo siguiendo una trayectoria determinada. Al mismo tiempo el sistema muestrea las posiciones cada cierto periodo de tiempo. Finalmente, el operador activa el robot reproduciendo la trayectoria memorizada.

3. Implementación Hardware

El diseño presentado ha sido definido mediante un lenguaje de descripción hardware (HDL) de alto nivel (*Handel-C*) [11]. Esto ha hecho posible la extracción de resultados experimentales de forma fácil gracias a la versatilidad de modificación del dispositivo reconfigurable. La arquitectura interna del diseño consta de seis módulos

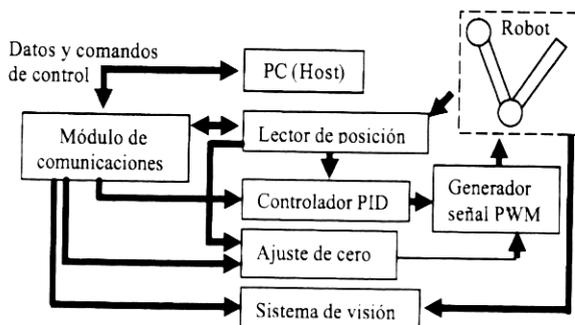


Fig. 3. Arquitectura interna del sistema. Se observan seis módulos funcionales trabajando en paralelo sincronizados mediante canales para procesamiento en tiempo real.

(Véase Fig. 3). Todos los módulos funcionan en paralelo y se sincronizan mediante canales *Handel-C* para obtener un rendimiento y sincronización óptimos. De este modo se consigue procesamiento y respuesta del sistema en tiempo real.

El **módulo de comunicaciones** se encarga de enviar y recibir datos mediante un conjunto de comandos predefinidos desde el PC. El conjunto de comandos sirve para indicar a la FPGA la tarea que se desea que realice, por ejemplo indicar una posición destino al controlador PID para una articulación, indicar que realice una calibración de sensores etc. Para la comunicación se utiliza el puerto serie RS-232 del PC a una velocidad 9600 bps (bits por segundo). Velocidad suficiente dado que la información que se transmite es mínima, pues todo el control está embebido en la FPGA. El envío y recepción de datos utiliza un protocolo de comunicaciones de diseño propio permitiendo recibir y enviar tramas de datos y tramas de control libres de errores. Para conseguir esto se numeran las tramas, se establecen comandos de confirmación ACK (*acknowledgment*) y control de paridad a nivel físico.

El **módulo lector de posición** calcula de forma continua las posiciones de las articulaciones mediante la comprobación del estado de los canales de los sensores de posición (Fig. 2). El proceso es el siguiente: Primeramente se detecta un cambio en cualquiera de los canales CHA o CHB obteniéndose un patrón (00, 10, 11 ó 01). Posteriormente el valor obtenido se compara con el patrón almacenado de una lectura anterior permitiendo conocer el sentido de giro de la articulación. Si el sentido es horario, el sistema incrementa una variable que almacena la cuenta de pasos del eje. Por el contrario, si el sentido de giro es anti-horario se decreta dicha variable. El resultado es un valor que codifica la posición de cada eje de giro. Cada sensor de posición tiene una resolución de 1024 ppr, como el mecanismo se basa en la detección de flancos (CHA, CHB), la resolución total del dispositivo es de 4096 ppr lo que supone aproximadamente 0.088 grados de precisión.

El **módulo controlador PID** se encarga de mantener continuamente la posición deseada de las articulaciones del brazo. Para realizar esta tarea se utiliza un controlador PID discreto y local para cada articulación. La ubicación del controlador se sitúa en la FPGA en vez de en el PC para evitar el flujo continuo de información de posiciones y acciones de control. De este modo el PC solamente realiza tareas de alto nivel (enviar posiciones deseadas, enviar parámetros de ajuste, etc.) mientras que la FPGA realiza todas las tareas que requieren procesamiento en tiempo real (generación de señal PWM, lectura continua de sensores, etc.). El controlador digital se implementa físicamente como un algoritmo programado en la FPGA, que obtiene la secuencia de valores $u(k)$ (secuencias de valores k -ésimos de acción para el motor) a partir de la secuencia de valores $e(k)$ (errores de posición k -ésimos de la articulación). Una posibilidad para obtener la representación discreta del controlador PID programable [12] consiste en partir de la ecuación diferencial continua (Eq. 1) y obtener una ecuación en diferencias discreta entre la señal $e(k)$ y la señal $u(k)$. La aproximación discreta consiste en aproximar la ecuación diferencial continua

$$u(t) = K_p \left(e(t) + T_d \frac{de}{dt} + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right) \quad (1)$$

calculando $u(kT)$ a partir de los valores $e(t)$ (error de posición: diferencia entre la posición deseada y la actual del eje en el instante t) en los instantes de muestreo [$e(T)$, $e(2T)$...]. Operando y sustituyendo los términos integral y derivativo en la (Eq. 1) por la aproximación más sencilla para la integral y la derivada (Eq. 2) da como resultado la expresión discreta del controlador (Eq. 3).

$$\int_0^{kT} e(t) dt \approx \sum_{j=0}^{k-1} e(jT) \cdot T \quad \frac{de(kT)}{dt} \approx \frac{e(kT) - e((k-1)T)}{T} \quad (2)$$

Observar que en (Eq.3) aparecen las constantes q_0 , q_1 y q_2 cuyo valor depende de los términos proporcional, integral y derivativo (Eq.4) que definen la dinámica de funcionamiento del controlador.

$$u_k = u_{k-1} + q_0 e_k + q_1 e_{k-1} + q_2 e_{k-2} \quad (3)$$

El ajuste de estos términos no es nada trivial porque dependen de las características mecánicas del dispositivo a controlar (peso, inercias, etc), además, un cambio en la

$$q_0 = k_p \left(1 + \frac{T_d}{T} \right) \quad q_1 = k_p \left(-1 + \frac{T}{T_i} - 2 \frac{T_d}{T} \right) \quad q_2 = k_p \frac{T_d}{T} \quad (4)$$

dinámica del dispositivo (por ejemplo sujetar un peso en el caso de brazos robots) puede necesitar un reajuste de estas constantes [3, 4]. La implementación del controlador digital en la FPGA, donde la disponibilidad de unidades de punto flotante es un recurso costoso, pasa por un escalado previo de valores que sitúe la respuesta del controlador en un rango de valores enteros suficientemente grande para disponer de precisión. El proceso de control para una articulación consiste en ejecutar periódicamente el código del controlador y realizar las siguientes operaciones: Leer el valor de la entrada (posición del brazo), calcular la acción de control según la ecuación del controlador, actualizar la salida (activar los motores del brazo). Generalmente este proceso se asocia a una interrupción de un temporizador que en el contexto de hardware reconfigurable será un registro contador incrementado cada ciclo de reloj. Un punto importante en la implementación es la comparación de la acción de control con los valores máximo y mínimo permitidos. Esta comparación es necesaria (en cada ciclo) para evitar el efecto de anti-reset (*wind-up*) del integrador. Si la acción de control se sale del rango admisible (el controlador se satura) y no se limita su valor en el programa, ésta puede crecer mucho sin verse reflejado en una acción sobre el proceso (por la saturación del actuador), lo que se traduce en una sobre-oscilación.

La detección de la posición del brazo robot, basada en el **sistema de visión**, usa como primitiva la información del color para segmentar los puntos de interés de la imagen, método comúnmente utilizado en aplicaciones de robótica [13]. La plataforma posee 4 leds rojos indicadores del campo de actuación del brazo, y un led verde que, colocado en el extremo del brazo, nos permite conocer su posición. El procesamiento realizado usa diferencias de componentes de color que permiten obtener un *mapa de saliencia* que presenta sólo los píxeles pertenecientes a los leds de posición (rojos y verdes). A partir de ellos calculamos las coordenadas del brazo con respecto a la imagen. La arquitectura hardware implementada en la FPGA se describe en la Fig. 4. Como entrada utiliza la imagen proporcionada por la cámara (VC). La salida del sistema consiste en la estimación de las posiciones de los leds con respecto al sistema de coordenadas de la cámara. La estimación numérica es enviada a la unidad de control del brazo y visualizada como un puntero superpuesto a la imagen original o procesada (dependiendo del modo de visualización) en un monitor VGA externo. El sistema consta de: (a) *Frame-grabber* para: adquisición, almacenamiento y envío de los píxeles a procesar. (b) Controlador del ratón; permite al usuario un calibrado inicial del sistema y la selección del modo de visualización (imagen original o imagen procesada). (c) Sistema de procesamiento y control de estados; los píxeles provenientes del *frame-grabber* son filtrados basándonos en diferencia de

componentes RGB. La posición estimada para cada *led* es el *centroide* de los pixeles filtrados de su color. Además, en el caso de los pixeles rojos utilizamos la información de su posición inicial introducida en la etapa de calibrado para restringir

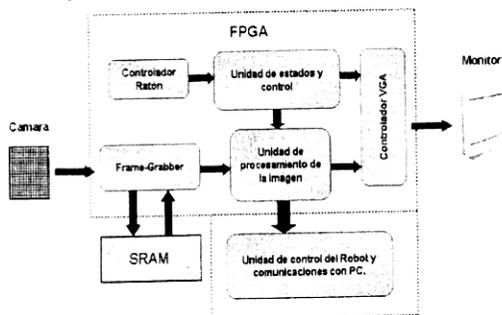


Fig. 4. Arquitectura hardware del sistema de detección de posición basado en visión.

el área de localización. Para cada imagen determinamos la posición del led verde colocado en el extremo del brazo (actualizándola cada 40 ms). Asimismo, dado que la posición de los *leds* rojos de la plataforma es estática, es suficiente con determinar por imagen la posición de sólo uno de los cuatro leds rojos de la plataforma (actualizando la información cada 160 ms). De esta manera podemos compartir el hardware de procesamiento para la estimación de la posición de los leds rojos mediante el uso de las señales de control apropiadas. (d) Controlador VGA; envía la información visual de la estimación al monitor.

El objetivo del **generador de señal PWM** consiste en mantener una señal cuadrada de frecuencia constante a 20Khz y tiempos en estado alto y bajo variables [8]. La arquitectura de este módulo es básicamente un conjunto de registros funcionando a modo de contadores. Cada contador definirá un periodo de tiempo en el que la señal de salida permanecerá en un estado (alto o bajo). Conociendo la frecuencia de reloj principal (50 Mhz en esta aplicación) se establecen los valores de dichos contadores para la generación de la señal PWM a la frecuencia deseada. La estrategia de generación de señal PWM elegida para el robot ha sido la de antifase (véase apartado 2). En este modo de funcionamiento la polaridad aplicada al motor cambia con el tiempo a una frecuencia de 20Khz (véase Fig. 5). Para producir este comportamiento se envía al controlador de potencia una señal cuadrada de frecuencia constante pero de tiempo en alta y baja variable. Si el tiempo que está la señal en estado alto es superior al tiempo en estado bajo el motor girará en un sentido (con una velocidad y par proporcional al tiempo en alta diferencial), mientras que si es al contrario girará en sentido opuesto. En el caso de ser iguales el motor permanecerá inmóvil.

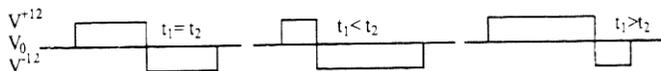


Fig. 5. Señal PWM en antifase. En cada ciclo se cambia la polaridad de alimentación del motor permitiendo parar, establecer el sentido de giro y la velocidad del mismo mediante la modificación del tiempo (t_1) de la señal.

4. Resultados experimentales y conclusiones

En esta experiencia mostramos un sistema robótico, basado en hardware reconfigurable (Véase Tabla 1), que sirve de base de estudio para sistemas de control adaptativo futuros. El objetivo a medio plazo pasa por dos líneas de investigación: (a) Ajuste dinámico del PID optimizando el consumo y error de posición utilizando algoritmos genéticos. (b) Sustitución del PID por una red neuronal de pulsos.

Módulos	% Consumo Slices	Frecuencia máx. Mhz
Comunicaciones	5.64	139.25
Lector de posición	0.97	127.68
Controlador PID	0.99	132.22
Ajuste de cero	0.97	127.68
Sistema de visión	34.3	25.62
Generador PWM	3.78	88.37
Lógica de apoyo (*)	1.85	121.62
TOTAL 1 (**)	37.9	33.82
TOTAL 2 (**)	64.98	22.54

Tabla 1. Porcentaje de ocupación y frecuencia máxima realizada con compilaciones parciales (aislamiento parcial de módulos). Software de desarrollo y síntesis DK-2 [9] e ISE-5.1i [10]. Nº slices de la FPGA : 5120. (*) Parpadeo de leds, displays etc. (**): Sin y con sistema de visión.

Referencias

1. SpikeFORCE, (EU Project: IST-2001-35271). <http://www.spikeforce.org>
2. EcoVision, (EU Project: IST-2001-32114). <http://www.pspc.dibe.unige.it/ecovision>
3. Ziegler-Nichols (1942): *Tuning rules and limitations*. <http://www.automationtechies.com/sitepages/pid1373.php>
4. Regulador PID: http://iaci.unq.edu.ar/materias/Identificacion_y_Control_Adaptativo/Apuntes/15-PID.pdf
5. Gerstner, W., Kistler, W.; *Spiking Neuron Models*. Cambridge University Press, (2002)
6. Jahnke, A.; Schoenauer, T.; Roth, U.; Mohraz, K.; Klar, H.: *Simulation of Spiking Neural Networks on Different Hardware Platforms*. ICANN97, LCNS, pp. 1187-1192 (1997)
7. National Semiconductors: <http://www.national.com>
8. Control de velocidad mediante PWM (Frenado regenerativo): <http://www.isa.cie.uva.es/proyectos/servos/info/PWM/PWM.htm>
9. Celoxica, <http://www.celoxica.com>
10. Xilinx, <http://www.xilinx.com>
11. Handel-C language referent manual. Celoxica 2003.
12. Roberto Sanchos: Implementación digital de controladores PID: http://www.tec.uji.es/asignatura/dir_asignaturas/3/67/Tema1.pdf
13. James Brusey and Lin Padgham. *Techniques for obtaining robust, real-time, colour-based vision for robotics*. 1999. In *IJCAI-99 Proceedings of the Third International Workshop on Robocup*. Stockholm, Sweden.