

Controlador óptimo de tiempo mínimo para un motor DC controlado en posición

Miguel A. Melgarejo R.

Alexis Javier Pirajan Aranguren

Cesar Augusto Valdés

RESUMEN

Este artículo presenta el proceso de modelado y realización de un controlador óptimo de tiempo mínimo para un motor DC controlado en posición. La arquitectura del controlador ha sido desarrollada usando un enfoque de codiseño, donde la adecuación de señal es ejecutada por hardware dedicado basado en un CPLD (Complex Programmable Logic Device) y las estructuras de control son módulos virtuales especificados en alto nivel. La validación funcional del controlador se ejecuta comparando algunas respuestas transitorias del sistema de control tanto en simulación como en operación real.

Palabras claves: Control óptimo, Controladores bang-bang, lógica programable, sistemas de control discreto.

Minimun time optimal controller for a positon controlled DC motor

ABSTRACT

This article presents the modeling and realization of a minimum time optimal controller for a position controlled DC motor. Controller architecture is implemented using a codesign approach where a CPLD (Complex Programmable Logic Device) is used as a signal adjustment system, and the control structures are high level specified virtual models. Controller functionality is verified comparing some transient responses of the control system under simulation and real work operation.

Keywords: Optimal Control, Ban-Bang Controllers, Programmable Logic, Discrete time control systems.

I. INTRODUCCIÓN

Son abundantes los trabajos enfocados al estudio y diseño de controladores óptimos pero son muy escasos aquellos que tratan su realización practica. Por lo tanto, esta clase de controladores queda excluida de una serie de aplicaciones industriales donde ofrecerían una solución eficiente a los problemas tratados con controladores convencionales como PIDs.

Teniendo en cuenta el panorama anterior, este trabajo pretende llevar a cabo el modelado para implementación práctica de un controlador óptimo de tiempo mínimo con arquitectura BANG-BANG para un motor DC controlado en posición.

Se ha adoptado un enfoque de codiseño para la solución del problema, debido a la clara metodología que esta concepción aporta al momento decidir que estructuras han de ejecutarse como software y cuales otras como módulos hardware. En este caso, todo el conjunto de sensores y actuadores han sido especificados como entidades funcionales empleando VHDL (Very High Speed hardware Description Language) para su realización en un CPLD (Complex Programmable Logic Device). El modelo de control es una representación de alto nivel en MATLABÓ simulinkÓ, la cual es la misma tanto en los procesos de simulación, como de operación real. Para efectos de esta última, se ha usado el estándar LPT (Local Paralel Port) en la interconexión del modelo virtual con el modulo hardware.

La funcionalidad del controlador es validada comparando los resultados de simulación y los observados en operación real. Con el fin de establecer un dominio de aplicación y perspectivas en el desarrollo de controladores óptimos.

II. CONTROL OPTIMO BANG-BANG

El modelado del controlador bang-bang inicia por la identificación de las variables de estado, de acuerdo a (1).

$$\begin{aligned} \bullet & \quad \dot{x}_1(t) = x_2(t) \\ \bullet & \quad \dot{x}_2(t) = -ax_2(t) + u(t) \end{aligned} \quad (1)$$

siendo la posición, la velocidad y a es la constante temporal del motor. La entrada $u(t)$ hace referencia a la función de conmutación cuyos extremos están en $+1$ y -1 .

Lo siguiente es encontrar el grupo de funciones denotadas como $O+$ y $O-$, al tomar $u = +1$ o $u = -1$. Las soluciones para las ecuaciones diferenciales (1) se presentan en (2):

$$\begin{aligned} x_2(t) &= c_1 e^{-at} \pm \frac{1}{a} [1 - e^{-at}] \\ x_1(t) &= -\frac{c_1}{a} e^{-at} \pm \frac{1}{a} t \pm \frac{1}{a^2} e^{-at} + c_2 \end{aligned} \quad (2)$$

Estas ecuaciones definen dos familias de curvas las cuales pasan por el origen, asumiendo $(t) = (t) = 0$ y $t = 0$ (puesto que el sistema es invariante en el

La funcionalidad del controlador es validada comparando los resultados de simulación y los observados en operación real. Con el fin de establecer un dominio de aplicación y perspectivas en el desarrollo de controladores óptimos.

La modificación de la arquitectura consiste en incluir una etapa de cálculo para error y su tendencia con el fin de generar un sistema de regulación capaz de llevar el estado de control hacia cualquier punto particular dentro de un rango definido.

La modificación de la arquitectura consiste en incluir una etapa de cálculo para error y su tendencia con el fin de generar un sistema de regulación capaz de llevar el estado de control hacia cualquier punto particular dentro de un rango definido. La figura 2a presenta la arquitectura modificada de control donde se observa una variable de proceso de naturaleza discreta tanto en amplitud como en tiempo (modelado del ADC basado en CPLD).

La función de conmutación es implementada como un esquema LUT (Look Up Table), el cual es una recopilación de valores a manera de tabla que describen adecuadamente dicha función en un rango particular. La salida de la función de conmutación es llevada hacia un verificador de signo que actúa como un conmutador perfecto.

El modelo del motor DC controlado en posición es descrito empleando la representación por función transferencia en velocidad más integración [4]. Se acopla igualmente una zona muerta para modelar la región de no activación del motor y así obtener una descripción del sistema más cercana a la realidad.

B. Interfaz de comunicación basada en LPT.

Se ha desarrollado una subrutina encargada de la interconexión de la estructura de control virtual con el sistema de muestreo basado en CPLD.

Las subrutina se invoca desde el modelo virtual como una función MATLAB[5], dentro de esta función se envían una serie de códigos de operación los cuales se encargan de sincronizar el proceso de conformación de palabra (de posición del motor) a partir de bloques de 4 bits, igualmente fijan las señales de conmutación a la salida del controlador bang-bang. Cada código de operación es un conjunto de bits que se establece en el puerto de control de LPT (Local Parallel Port)[6], se ha escogido este estándar debido a su funcionalidad y fácil operación.

La subrutina de comunicación reemplaza al modelo de motor controlado en posición, por tal razón es necesario complementarla con un retardo a su salida para garantizar condición de arranque en $t=0$. A partir de ese instante, la Subrutina es ejecuta en cada ciclo de simulación. La figura 2b presenta el modelo de control para operación real, se observa una ganancia previa a la entrada del controlador con el fin de escalar en posición la salida de la subrutina.

Es claro que el salto ejecutado entre el modelo de control para simulación y el modelo de operación real es mínimo, tan solo ha requerido de la modificación de un bloque y no ha sido necesario ejecutar una descripción de código que represente toda la estructura de control, por lo tanto el tiempo de diseño empleando este enfoque tiende a ser menor que una solución tradicional que requiera de la elaboración completa de un programa de control.

C. Plataforma hardware

La primera de las dos partes del controlador, se basa en un sistema digital de conversión, el cual toma la información proveniente del sensor y la transforma en datos digitales, además de esto, se debe realizar un ajuste en la longitud del dato final, que se pueda ingresar por el LPT hacia el PC usando modo normal, siendo esta longitud de tan solo 4 bits, ya que se hace necesaria la utilización del resto del puerto para control y escritura.

Todo este trabajo de bajo nivel, es ejecutado por un dispositivo lógico programable (CPLD), el cual proporciona la flexibilidad necesaria para realizar cualquier tipo de cambio de manera rápida, y ajustar el sistema a las condiciones que se impongan. Dentro de este se incluye tanto el conversor ADC, como la multiplexación y adecuación de la información que ingresa al PC, así como también, las señales de control.

Se hace necesaria la implementación de un sensor que permita la correcta operación del sistema, este sensor es un par infrarrojo emisor-receptor figura 3b, el cual proporciona información de la posición angular del sistema a partir de un disco perforado en su periferia, este disco posee 20 agujeros, los cuales permiten que el controlador inicie en cualquier punto, pero se detenga en el ángulo que se requiera con cierta exactitud (figura 3a), a continuación, la señal obtenida del par infrarrojo, es procesada por el sistema HARDWARE diseñado.

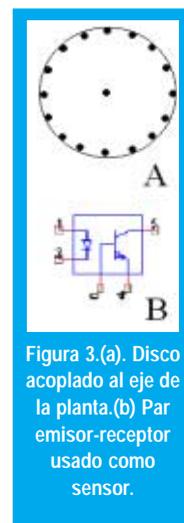


Figura 3.(a) Disco acoplado al eje de la planta.(b) Par emisor-receptor usado como sensor.

Este sistema HARDWARE fue implementado dentro de un dispositivo lógico programable (XC95108PC84) [2], el cual está conformado por unidades que cumplen funciones específicas al interior del chip, estas unidades son: Bloques de entrada salida (I/O blocks), bloques de función (function blocks), y por último una matriz de interconexiones (figura 4a).

Los bloques de entrada salida, son la interfaz entre el mundo externo, y la lógica interna del dispositivo, estos bloques disponen de buffers de entrada, drivers de salida, multiplexores de selección de salida, cada uno de estos bloques permite la configuración que se desee del pin, (estados de alta impedancia, entradas o salidas, bidireccionalidad). Además de proporcionar el tipo de adecuación del pin externo a cualquier necesidad, estos bloques proporcionan la interfaz, para cualquier tipo de tecnología a la cual se desee acoplar, desde TTL (5V), hasta tecnologías CMOS(2,5V).

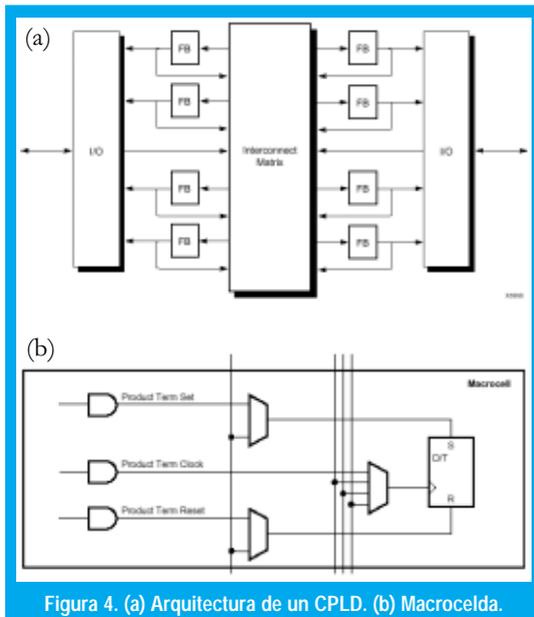


Figura 4. (a) Arquitectura de un CPLD. (b) Macrocelda.

Una vez las señales que se desean procesar pasan por los bloques I/O, son puestas a disposición de los bloques funcionales, los cuales están conformados por tres estructuras, un arreglo programable de compuertas and, un expansor de términos, y un grupo de 18 macroceldas, en el primero mencionado, se ejecutan todas las posibles operaciones de 36 entradas que posee cada bloque, seguido de este, el expansor de términos direcciona cada uno de esos resultados hacia las respectivas macroceldas.

En las macroceldas (figura 4b) se centra la operación del CPLD, cada una de estas posee un flipflop, el cual puede ser configurado como D o T, dependiendo de la función, además cada macrocelda, posee señales de control, tales como clock, set, reset, haciendo que se virtualmente se pueda configurar, cualquier tipo de operación sobre este dispositivo.

Por último, la matriz de interconexión, es la encargada de hacer posible la comunicación interna de las señales, por medio de esta, todos los bloques de función, están conectados entre si, solo es necesario habilitar algunas conexiones, para dirigir los productos útiles hacia donde se requieren.

D. Especificación del sistema.

La descripción hardware cuenta con un convertor ADC, un multiplexor, y las respectivas señales de control.(figura 5).

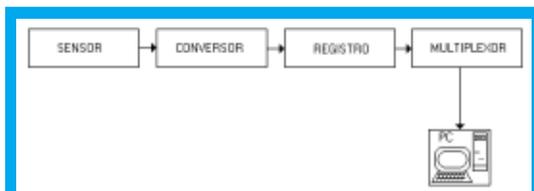


Figura 5. Diagrama de bloques del sistema desarrollado en hardware

La conversión de los datos analógicos a digital, se hace por medio de un contador descrito en VHDL (figura 6), este contador tiene por función, determinar el numero de agujeros que pasan por él, este contador es binario de 16 bits, lo cual permite tener una precisión alta en cuanto a posición angular se refiere.

La longitud del contador, es demasiado grande para hacer que ingrese a través del LPT hacia el PC, por lo tanto, se hace necesaria la intervención de un registro, y un multiplexor. En el registro se almacenan temporalmente los datos del contador, mientras que el PC captura la palabra de 16 bits, segmentada en palabras de 4 bits, esta segmentación es la que se logra por medio del multiplexor, de tal manera que el PC, lee cuatro palabras de 4 bits, y una vez que termina, refresca el dato que esta en el registro iniciando una nueva lectura del contador.

Dentro de la descripción VHDL, esta incluida una interfaz para la salida de las señales de control, las cuales inciden sobre los actuadores.

El actuador es un sistema de conmutación MOSFET [7] en configuración H, el cual permite fácilmente a partir de señales digitales, cambiar el sentido de giro al motor, y su velocidad.

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity control is
port ( clk,set: in bit;
      dac: out std_logic_vector (3 downto 0);
      acto:out integer range 0 to 3;
      sel,acti: in integer range 0 to 3
);
end control;

architecture gilp of control is
signal cc,co: std_logic_vector (15 downto 0);
begin

process(clk)
variable ct: std_logic_vector(15 downto 0);
begin
if (clk'event and clk='1') then
ct:=ct+1;
cc<=ct;
end if;
end process;

process (set,co,cc)
begin
if (set'event and set='1') then
co<=cc;
end if;
end process;

process(sel,co)
begin
case sel is
when 0 => dac <= co(15 downto 12);
when 1 => dac <= co(11 downto 8);
when 2 => dac <= co(7 downto 4);
when 3 => dac <= co(3 downto 0);
end case;
end process;

process(acto,acti)
begin
acto<=acti;
end process;
end gilp;

```

Figura 6. Descripción VHDL del sistema sensor-actuador

III. VALIDACION DE FUNCIONALIDAD.

La figura 7 presenta la comparación de las respuestas transitorias para un punto de establecimiento de 500 revoluciones (giros efectivos), se observa que la única diferencia notable entre las dinámicas aparece

La conversión de los datos analógicos a digital, se hace por medio de un contador descrito en VHDL, este contador tiene por función, determinar el numero de agujeros que pasan por él.

La fuerte conmutación que se presenta al final de la respuesta transitoria es necesaria para frenar el motor en la posición de establecimiento, esta apreciación lingüística del fenómeno, implica que el controlador BANG-BANG ejecuta acciones de aceleración y de frenado.

en la región inicial donde se presenta un crecimiento más lento para el modelo de simulación, esto indica que la zona muerta estimada para el motor es muy grande, de hecho medirla experimentalmente puede ser engorroso, pero los resultados demuestran que no es un parámetro crítico en este caso.

Las señales de conmutación del controlador BANG-BANG teórico y real se presentan en la figura 8, es necesario destacar que la magnitud ± 30 corresponde a los valores límites disponible en la fuente de alimentación, por lo tanto no es necesario incluirlos dentro del modelo de operación debido a que este las envía al modulo actuador como señales binarias con representación complemento dos (+1 "01" y -1 "11").

La fuerte conmutación que se presenta al final de la respuesta transitoria es necesaria para frenar el motor en la posición de establecimiento, esta apreciación lingüística del fenómeno, implica que el controlador BANG-BANG ejecuta acciones de aceleración (zona inicial) y de frenado (zona final). Es evidente que aunque se este llevando a cabo un establecimiento en tiempo mínimo, el consumo de energía tiende a ser elevado debido a estas acciones de frenado.

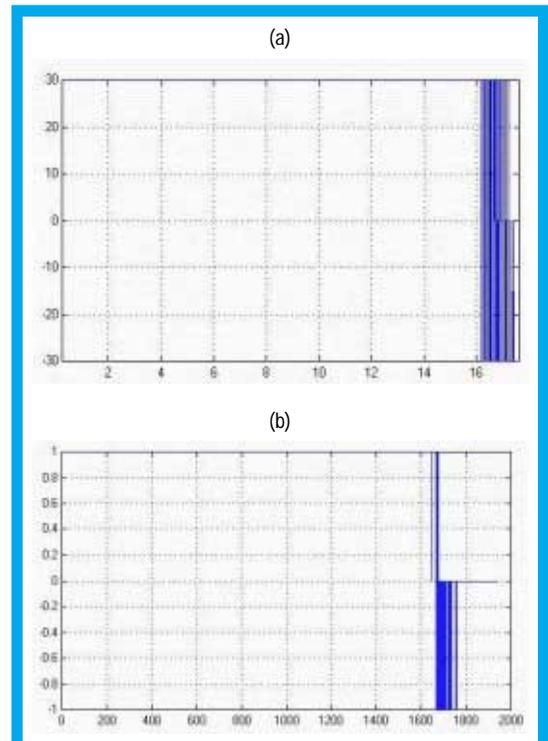


Figura 8. Señales de conmutación del controlador BANG-BANG. (a) Modelo de simulación. (b) modelo de campo.

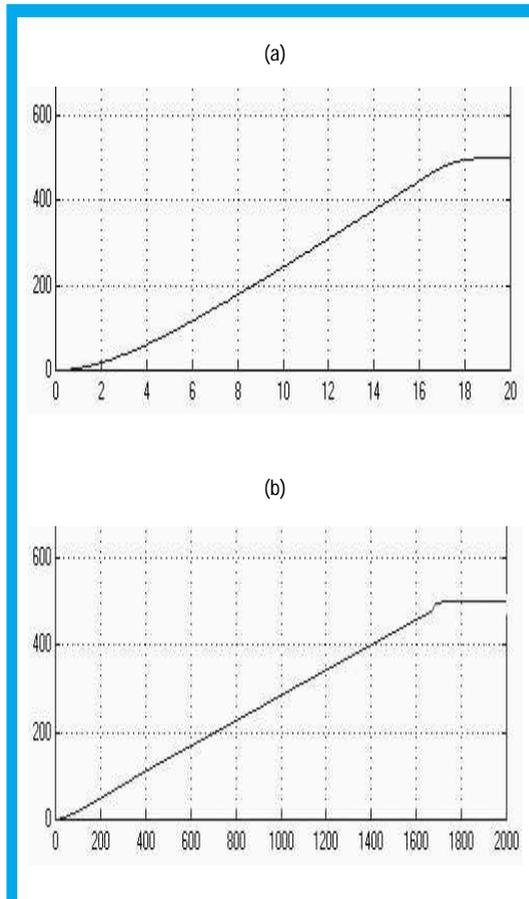


Figura 7. Respuesta transitoria del sistema de control. (a) Observada en simulación. (b) Observada en campo., Frecuencia de muestreo 100Hz, punto de establecimiento 500 revoluciones.

La figura 9 presenta la dinámica del error obtenida en simulación y operación para un punto de establecimiento de 300 revoluciones, para este caso la consideración de una zona muerta elevada hace que la predicción tenga una diferencia de tiempo con la dinámica real cercana a dos segundos.

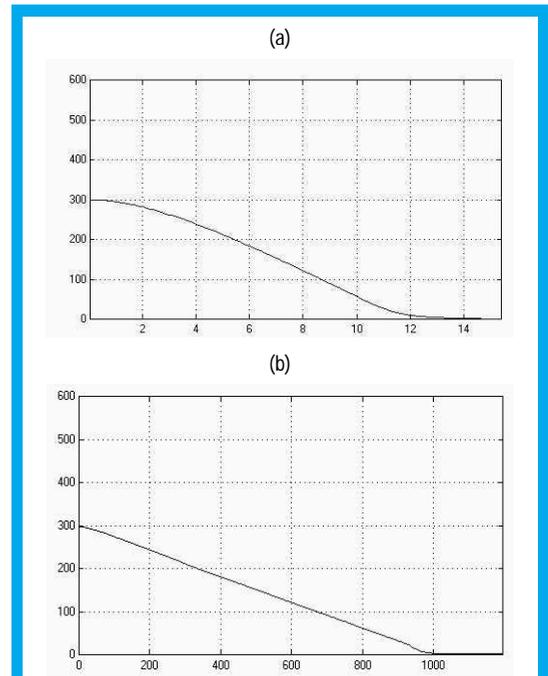


Figura 9. Dinámica del error para un punto de establecimiento de 300 revoluciones. (a) Obtenida en simulación. (b) Observada en Operación (frecuencia de muestreo 100 Hz).

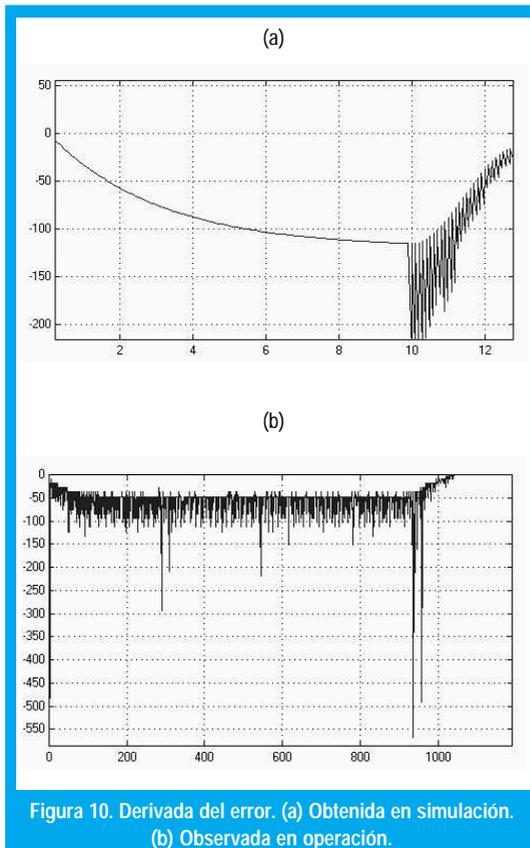


Figura 10. Derivada del error. (a) Obtenida en simulación. (b) Observada en operación.

Por último se presenta la derivada de la señal de error en la figura 10. Se observa que en el caso de operación en campo la derivada esta contaminada de pulsos de alta frecuencia esto, debido a que se esta ejecutando el calculo de una derivada continua sobre una señal de naturaleza discreta en tiempo y amplitud. Luego la transición de la señal discreta de error (que se origina en un tiempo muy pequeño) genera un valor elevado en función del proceso a limite cero que se ejecuta en el calculo de la derivada.

Esto indica que es necesario realizar el modelado del controlador bang-bang sobre una derivada discreta con el fin de observar las implicaciones de su uso, de hecho es de vital importancia al considerar una realización completa en hardware de la arquitectura BANG-BANG.

La frecuencia de muestreo indicada en cada una de las pruebas depende exclusivamente de la velocidad de procesamiento (captura de puerto, calculo o y visualización). Quizás esta sea la principal restricción en el uso de la herramienta simulink + LPT para el tratamiento en línea de procesos reales. Aunque, el periodo de muestreo resultante es valido para una gran cantidad de procesos de control.

Esto demuestra que el controlador puede ser una solución apropiada en brazos móviles o bandas transportadoras donde se requiere rápida respuesta y fina precisión.

V. CONCLUSIONES

Se ha presentado el desarrollo practico de un controlador óptimo de tiempo mínimo con arquitectura bang-bang para un motor DC controlado en posición.

Una gran parte de los componentes que conforman el sensor dedicado y el actuador han sido embebidos en un solo dispositivo con un costo razonable de recursos físicos, dándose el primer paso en la implementación hardware completa del controlador.

La validación funcional ha permitido observar que es necesario estimar apropiadamente la zona muerta en motores DC con el fin establecer un modelo de simulación mas conveniente para puntos de establecimiento de pequeña magnitud.

Se ha observado el comportamiento de tiempo mínimo y cero sobre impulso en las diferentes pruebas presentadas, esto demuestra que el controlador puede ser una solución apropiada en brazos móviles o bandas transportadoras donde se requiere rápida respuesta y fina precisión.

REFERENCIAS

- [1] Kirk D., " Optimal Control ", Prentice Hall, 1995.
- [2] Xilinx Co., " The programmable Logic Databook ", Xilinx, 1999.
- [3] The Mathworks Co., " MATLAB simulink user guide ", Mathworks, 1999.
- [4] Ogata K., " Ingeniería de control moderna", Prentice Hall, 1995.
- [5] The Mathworks Co., " MATLAB help", Mathworks, 1999.
- [6] Craig P., "Interfacing the Standard Parallel Port", www.senet.com.au/~cpeacock,1998.
- [7] National Semiconductor, " Analog and Interface products databook ", National, 1999 edition.

Miguel A. Melgarejo R.

Ingeniero Electrónico Universidad Distrital. Desarrollo su trabajo de grado en el diseño e implementación de un controlador difuso basado en FPGA, presentado en el VII Workshop IBERCHIP IWS'2001, Uruguay. Uno de los fundadores del Grupo de Investigación en Lógica Programable y técnicas digitales UD (GILP). mikael_xie@hotmail.com

Alexis Javier Pirajan Aranguren

Ingeniero Electrónico Universidad Distrital, Desarrolló su trabajo de tesis sobre síntesis digital de voz basada en transformada Wavelet con uso especial de MATLAB. Uno de los fundadores del Grupo de Investigación en Lógica Programable y técnicas digitales UD (GILP). alexisjavier79@hotmail.com

Cesar Augusto Valdés

Ingeniero Electrónico Universidad Distrital. Desarrolló su trabajo de tesis sobre síntesis digital de voz, específicamente en el modelamiento frecuencial y temporal de los fonemas. elecesar@hotmail.com