

Implementación de un controlador borroso usando técnicas de PWM analógico

J.Tombs, A.Torralba y L.G.Franquelo

Dpto. de Ingeniería Electrónica, de Sistemas y Automática

Escuela Superior de Ingenieros

Avda. Reina Mercedes, s/n, SEVILLA-41012

Tlf: 34-5-4556849,

Fax: 34-5-4556849

e-mail: jon@gtex.esi.us.es

Resumen

Hoy día la mayoría de los sistemas de control borroso se implementan en software usando procesadores y micro controladores digitales. Los diseños digitales programables son capaces de resolver muchos problemas de control, pero en sistemas de tiempo real, donde los señales de control tienen que cambiar en solo unos ciclos del reloj, la implementación hardware es la única solución posible.

En este artículo se presenta un diseño de VLSI hardware, basado en las técnicas de diseño analógicas usando trenes de pulsos. El diseño implementa todas las partes de un controlador borroso de alta velocidad. Se ha construido un chip de prueba en 1.5µm CMOS.

1 Introducción

El uso de lógica borrosa tiene una larga historia, empezando en las primeras publicaciones de L.A.Zadeh [1] en 1986. No obstante, el uso de controladores borrosos ha aparecido solamente en los últimos años. Hoy día pueden encontrarse controladores borrosos en muchos campos, desde la robótica hasta la electrónica de consumo [24], presentando en todos éstos campos la lógica borrosa presenta una solución de alta eficiencia.

El primer chip lógico-borroso fue diseñado por Togai y Watanabe [2]-[3], usando innovadores diseños digitales. Después Yamakawa [4]-[5] desarrolló un controlador borroso usando circuitos analógicos. En los últimos años se han sido propuestos [7]-[12] diseños utilizando ambas técnicas, digital y analógica. En esta trabajo se presenta un diseño que utiliza un híbrido de las técnicas digital y analógica.

El uso de circuitos analógicos es muy atractivo para VLSI borroso, ya que sencillas cel-

das analógicas permiten la implementación de las funciones necesarias para un controlador borroso. Los circuitos analógicos son eficientes, usan poco silicio y permiten, con sencillez, el procesamiento en paralelo. Los circuitos digitales son robustos ya que no sufren los efectos del ruido ni la dispersión de propiedades a consecuencia del proceso de fabricación como en el caso de circuitos analógicos. Sin embargo, en circuitos digitales las operaciones del controlador borroso usan mucho silicio, y el procesamiento en paralelo es muy complejo.

Se proponen técnicas de trenes de pulsos [13]-[15] con el objetivo de superar las desventajas de circuitos analógico, las. En las técnicas de trenes de pulsos, los pulsos digitales representan valores analógicos en su frecuencia, ancho o fase. En este trabajo se usa la técnica de "pulse width modulation" (PWM) usando el ancho de cada pulso para representar valores. El uso de PWM ha sido elegido por su velocidad, y su fácil aplicación a las funciones de un controlador borroso.

2 Estructura del controlador

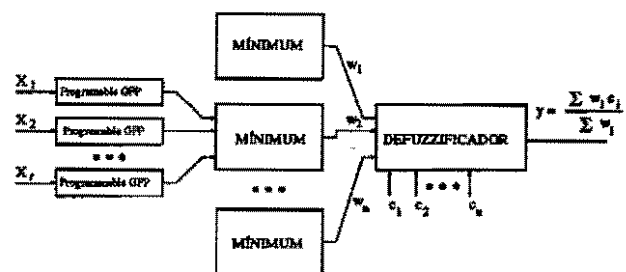


Figura 1: Diagrama del controlador

En figura 1 se muestran los bloques del controlador borroso. El controlador usa el método de

correlación-producto[16] para el proceso de inferencia. La operación de y borrosa se implementa usando la operación MÍNIMO. Entonces el proceso de implementación de la regla i es:

Regla i : Si x_1 es A_{1i} ; y ... y x_r es A_{ri}
 ENTONCES y es C_i

Donde x_i es la entrada i , r es el número de entradas, y A_{ri} representa el grado de pertenencia a un conjunto borroso.

Esto se implementa usando la expresión:

$$\begin{aligned} \mu_{C_i} &= \text{MIN}(\mu_{A_{1i}}(x_{1k}), \dots, \mu_{A_{ri}}(x_{rk})) \times \mu_{C_i} \\ &= w_i \times c_i \end{aligned} \quad (1)$$

Las funciones de pertenencia A_{ri} se asumen triangulares, y para la salida se consideran pulsos unitarios (*singletons*) con valor c_i centrados en μ_{C_i} . La salida de defuzzificador y se calcula usando el método de centroide, entonces:

$$y = \frac{\sum_1^n w_i \times c_i}{\sum_1^n w_i} \quad (2)$$

Donde n es el número de reglas en el controlador. Los valores de c_i y el tamaño y posición de las funciones de pertenencia son programables.

En éste diseño, hay tres fases de procesamiento, cada una de 500ns; la primera es de reset, donde los condensadores se descargan y se evaluó el valor de pertenencia de cada entrada. En la segunda fase se ejecuta la multiplicación y adición de las reglas, y durante la última fase, se calcula y almacena el resultado.

2.1 Generación de las funciones de pertenencia

En la figura 2 se muestra la estructura interna del circuito para generar las funciones de pertenencia, en la que se usa el "current mode approach" [17]. En la figura 2 se puede ver como la corriente I_{in} se resta de la corriente I_{mid} usando espejo de corriente. La corriente I_{mid} especifica el centro de la función de pertenencia. La diferencia se rectifica usando el circuito propuesto en [18] y se amplifica. Al resultado se le resta la corriente de referencia I_{max} . El valor de la corriente final muestra el valor de pertenencia. La corriente I_{mid} y la amplificación del espejo de corriente se programa digitalmente usando interruptores analógicos. El circuito puede generar una función de pertenencia triangular diferente para cada valor de I_{mid} y con la ganancia del espejo se especifica el ancho. Un circuito similar ya ha sido usado antes en [19].

A partir de la salida de corriente analógica, es necesario convertir a la forma PWM. La corriente

I_{out} se usa para cargar un condensador por un tiempo fijo. Así se produce una tensión proporcional a la intensidad. En el diseño se ha aprovechado el tiempo de carga para reiniciar el multiplicador quedando así preparado para la próxima combinación de las reglas. Durante la fase de multiplicación, se compara la tensión que representa la pertenencia con una señal de dientes de sierra V_{ref} , y se genera un pulso se genera durante el tiempo que V_{ref} está por debajo de la tensión de pertenencia. Este pulso V_{wi} de la regla i es el valor en forma de PWM de I_{out} . El espejo de corriente se usa para generar la señal de sierra, y calibrar así automáticamente la señal para que un valor de máxima pertenencia provoque un pulso de máximo ancho. El uso de un espejo produce una alta robustez contra los problemas de variación en temperatura o fabricación sin necesidad de controles externos.

Como las entradas al bloque de MÍNIMO son todas en forma PWM, sincronizadas en su flanco de subida, una sencilla puerta *and* basta para ejecutar la operación de MÍNIMO.

Para calcular la salida del controlador en formato 'escalar', hay que evaluar la expresión (2). Para ello hace falta un circuito que pueda implementar multiplicación, adición y también división. Este proceso se ejecuta en dos pasos.

2.2 Multiplicación y adición

En la figura 3, se muestra el circuito de multiplicación. El circuito usa un amplificador de transconductancia adaptado para PWM. El tamaño de la corriente I_1 en la figura 3 es proporcional a la tensión V_{gi} , y dura el ancho de la entrada en PWM V_{wi} . La multiplicación esta representada por la carga transferida. Las tensiones V_{gi} son programadas a los valores de los singletons V_{ci} . Se ha incluido el transistor $M4$ de la figura 3 para reducir el ruido en las líneas de alimentación.

También puede verse el circuito de adición, en figura 3. Aquí se usa un amplificador de bias dinámico (un inversor) y un condensador de integración. Usando un amplificador tan simple se da estabilidad al circuito, sin los problemas de compensación de un amplificador de más etapas. Al final de la fase de adición se muestra la salida del amplificador. Esta tensión es proporcional al valor $\sum g_i c_i$ deseado. En orden a calcular la adición, la capacidad de C tiene que ser constante, por tanto no se puede usar la puerta de un transistor como condensador.

Este diseño propuesto en [14] ha sido usado antes en redes neuronales. Un circuito de prueba, hecho en 1.5 μm CMOS ha mostrado que se puede obtener una alta precisión usando poco área

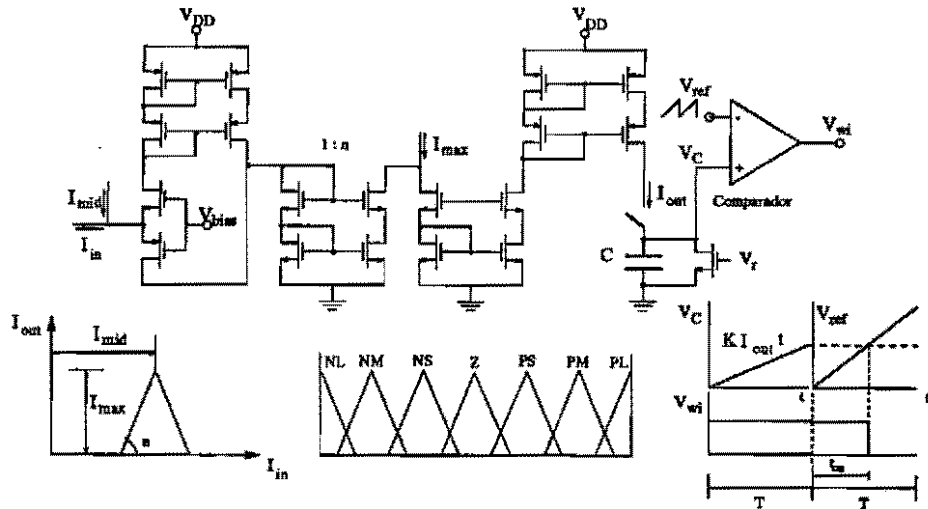


Figura 2: Generación de funciones de pertenencia

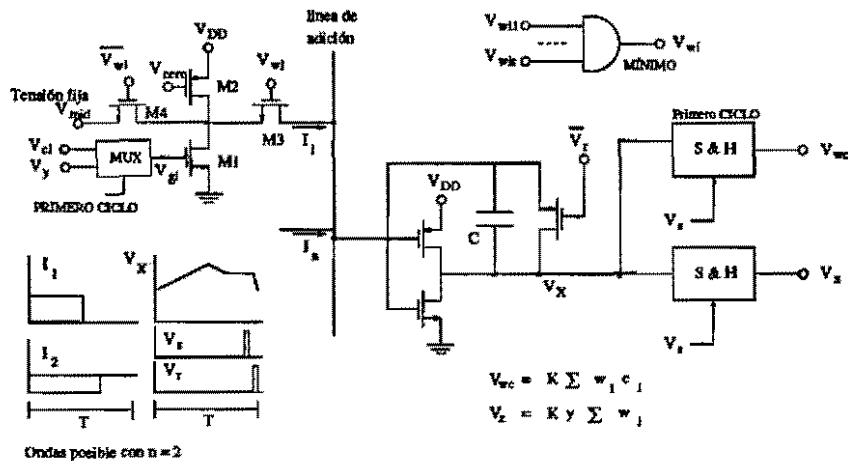


Figura 3: Circuito de multiplicación y adición.

de silicio [20]-[21].

2.3 División

La multiplicación y adición se puede ejecutar de la forma descrita anteriormente, pero en la defuzzificación, es la división la que resulta más difícil. Es posible implementar la división analógica de diversas formas: completamente analógica, aproximaciones sucesivas o con trenes de pulsos. En la primera de ellas, dos amplificadores operacionales están configurados para ejecutar la división [25]-[27], mientras que con aproximaciones sucesivas, se repite la multiplicación para acercarse a la solución de manera progresiva. Por último, con la utilización de trenes de pulsos

se aprovechan las propiedades de la señal para ejecutar la división. En el chip de prueba diseñado, están implementadas tanto las aproximaciones sucesivas como la división por trenes de pulsos. Esto permite la comparación entre los dos métodos. Aunque las aproximaciones sucesivas tienen como ventaja el producir salidas digitales y analógica a la vez, resulta ser un método lento. La utilización de trenes de pulsos resulta ser más rápida, pero debido a que el tiempo de división es corto, los errores motivados por el ruido y el retraso de las señales pueden derivar en errores en el resultado.

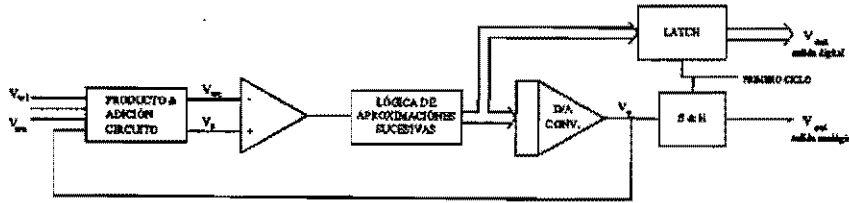


Figura 4: Circuito de división usando aproximación sucesiva

2.4 División por aproximación sucesivo

La figura 4 muestra el circuito de aproximaciones sucesivas usando el circuito de multiplicación/adición de la figura 3. En el primer ciclo de inferencia, los multiplicadores reciben como entradas el PWM de pertenencia y las tensiones V_{gi} representativas de los singletons V_{ci} , almacenando la salida $K \sum w_i c_i$. En los próximos ciclos, se aplica la entrada V_y a las puertas de todos los transistores M1, i.e. $V_{gi} = V_y \quad i = 1 \dots n$, siendo la salida:

$$V_x = Ky \sum w_i$$

En la lógica de aproximaciones sucesivas, se genera la entrada V_y que en $m+1$ ciclos converge a la solución de 2:

$$\begin{aligned} Ky \sum w_i &= K \sum w_i c_i \\ \Rightarrow y &= \frac{\sum w_i c_i}{\sum w_i} \end{aligned} \quad (3)$$

Hay que notar que el resultado no depende de la constante de integración K (que depende de valor de condensador C). Al final de $m+1$ ciclos, el controlador produce una señal de validación y el valor del resultado se puede leer tanto en forma digital como analógica.

2.5 División usando trenes de pulsos

Para ejecutar la división usando trenes de pulsos se usa el mismo circuito de la figura 3 es usado. Como antes, en el primer ciclo de una inferencia se almacena la salida en un condensador V_{wc} . En el siguiente ciclo todos los singletons están puestos al mismo valor, produciéndose a un tiempo t una tensión $V_{wc'}$ en el circuito de adición, donde $V_{wc'}$ es:

$$V_{wc'} = \alpha t \sum W_i \quad (4)$$

Donde α es una constante de proporcionalidad. Después de un tiempo T se llega a la tensión V_{wc} :

$$\begin{aligned} \alpha T \sum W_i &= K \sum W_i C_i \\ \Rightarrow T &= K' \frac{\sum W_i C_i}{\sum W_i} \end{aligned} \quad (5)$$

Donde K' es una nueva constante de proporcionalidad, elegida para controlar el valor máximo de T . El circuito genera un pulso que se mantiene en alto mientras $V_{wc'}$ es menor que V_{wc} . Así se produce un resultado por cada dos ciclos de reloj, resultando ser cinco veces más rápido que el método de aproximaciones sucesivas. Internamente, un condensador esta cargado durante el tiempo T , de esta forma al final de la inferencia, el valor de salida esta disponible tanto en la forma de tensión como de trenes de pulsos.

3 Resultados simulado

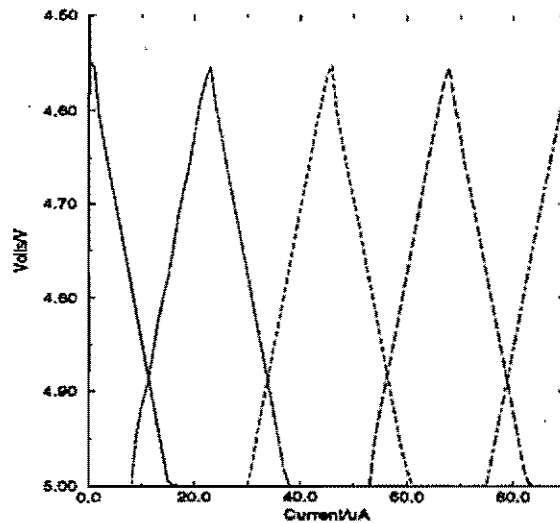


Figura 5: Funciones de pertenencia producidas

En la figura 5 se muestra la respuesta transitoria de las funciones de pertenencia usando cinco

de las posiciones programables. Para la simulación, se extrae el circuito directamente del diseño físico. Las curvas se reducen cambiando la entrada rápidamente entre 0 y 90uA en 90us (1A/s). Las curvas representan el rápido cambio de la salida en función de las entradas y la programabilidad de la posición de las funciones de pertenencia.

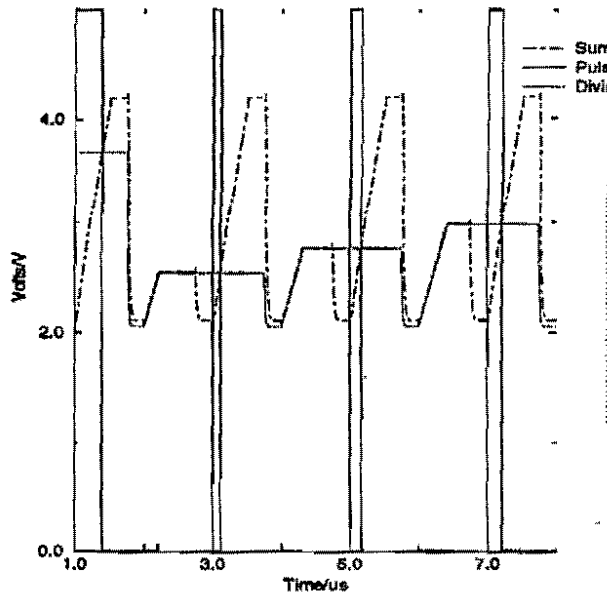


Figura 6: Ondas al salida del defuzzificador

La figura 6 muestra una simulación del circuito defuzzificador. En esta simulación se pueden ver las tres fases del circuito: carga de la capacidad, comparación y reset. La defuzzificación se ejecuta cada dos ciclos, y en la figura 6 la salida toma los valores 0.2, 0.3 y 0.4.

4 Conclusiones y futuras líneas de investigación

En la figura 7 se muestra el chip de prueba, actualmente en fabricación. El chip tiene dos entradas, dieciocho funciones de pertenencia programables, nueve reglas y una salida.

Se puede ver en la parte superior derecha la estructura regular que contiene la parte analógica del controlador en su totalidad. El resto del chip contiene la lógica de aproximaciones sucesivas, los registros digitales programable para la configuración de las reglas y singletons y un interfaz al bus de un PC/XT. El tamaño y número de reglas de este chip junto con un reloj master a 660KHz conseguirá aproximadamente 100 KFLIPS en el modo de aproximaciones sucesivas, y 500 KFLIPS en el modo PWM. Se aplicará el

chip al problema del control de una fuente resonante [23, 22].

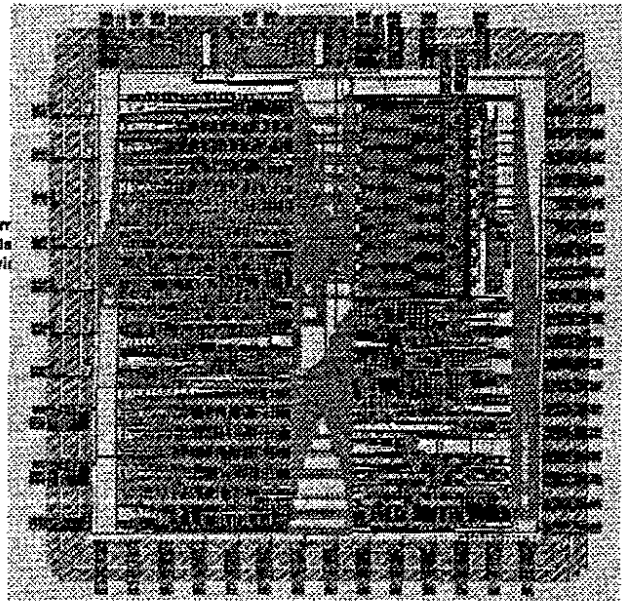


Figura 7: Chip de prueba

Como el diseño usa los mismos circuitos para la división y la multiplicación, las principales fuentes de errores son del tipo "clock feed-through" y de los retrasos y offsets de los comparadores. Estos errores son normalmente sistemáticos, pudiendo ser eliminados por medio de compensación externa, o en un proceso de aprendizaje *insitu*. Próximos diseños incorporarán la capacidad de aprender.

También se están estudiando métodos de implementar todo el controlador usando técnicas exclusivamente analógicas.

Referencias

- [1] L.A.Zadeh. "Fuzzy Sets". *Inform.Contr.*, vol. 8, pp. 338-358, 1965.
- [2] M.Togai and H.Watanabe. "Expert system on a chip: an engine for real-time approximate reasoning". *IEEE Expert Syst. Mag.*, vol. 1, pp. 55-62, 1986.
- [3] H.Watanabe, W.Dettloff and E.Yount. "A VLSI Fuzzy Logic Inference Engine for Real-Time Process Control". *IEEE Journal of Solid-State Circuits*, vol. 25, no. 2, pp. 376-382, April 1990.
- [4] T.Yamakawa and T.Miki. "The current mode fuzzy logic integrated circuits fabricated by the standard CMOS process". *IEEE Trans. Computers*, vol. C-35, no. 2, pp. 161-167, 1986.
- [5] T.Yamakawa. "Fuzzy microprocessor rule chip and defuzzifier chip". *Proc. of Int. Workshop on Fuzzy Syst. Appl.*, Aug. 1988, pp. 51-52.
- [6] M.Sasaki, T.Inone, Y.Shirai and F.Ueno. "Fuzzy multiple-input maximum and minimum circuits in current mode and their analysis using bounded-difference equations". *IEEE Trans. on Comput.*, vol. C-39, no. 6, June 1990, pp. 768-774.

- [7] H.Watanabe, J.R.Symon, W.Dettloff, and E.Yount. "VLSI Fuzzy Chip and Inference Accelerator Board Systems". *Proc. of IEEE Int. Symp. on Multi-Valued Logic*, pp. 120-127, May 1991.
- [8] T.Taukano, T.Inone, F.Ueno. "A Design of Current-Mode Analog Circuits for Fuzzy Inference Hardware Systems", *Proc. Int. Conf. Fuzzy Systems*, San Diego, March 1992, pp. 1385-1388.
- [9] T.Miki, H.Matsumoto, K.Ohto, T.Yamakawa. "Silicon Implementation for a Novel High-Speed Fuzzy Inference Engine: Mega-FLIPS Analog Fuzzy Processor". *Journal of Intelligent & Fuzzy Systems*, vol.1, no. 1, 1993, pp. 27-42.
- [10] F.Colodro, A.Torralba and L.G.Franquelo. "A Fuzzy-logic chip using stochastic-logic: the defuzzifier". *Proc. of the Int. Conf. on Fuzzy Control Applications, IFCA'93*, Tarrasa, February 1993.
- [11] J.L.Huertas, S.Sánchez-Solano, A.Barriga, I.Baturone. "A Fuzzy Controller using Switched-Capacitor techniques". *Proc. Int. Conf. Fuzzy Systems*, San Francisco, March 1993, pp. 516-520.
- [12] F.Colodro, A.Torralba and L.G.Franquelo. "A Digital Fuzzy-logic controller with a simple architecture". Submitted for presentation at the *Int. Symp. Circ. and Syst., ISCAS'94*.
- [13] A.F.Murray. "Pulse Arithmetic in VLSI Neural Networks". *IEEE Micro Mag.*, Dec. 1989, pp. 64-74.
- [14] A.F.Murray, D.del Corso and L.Tarassenko, "Pulse-Stream VLSI Neural Networks mixing Analog and Digital techniques". *IEEE Trans. on Neural Networks*, vol. 2, no. 2, March 1991.
- [15] A.Hamilton, A.F.Murray, D.J.Baxter, H.Martin Reeki and L.Tarassenko. "Integrated pulse stream Neural Networks: Results, Issues and Pointers". *IEEE Trans. on Neural Networks*, vol. 3, no. 3, May 1992, pp. 385-393.
- [16] C.C.Lee. "Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Parts I and II". *IEEE Trans. Syst. Man Cybern.*, vol. 20, no.2, pp. 404-435, Mar./Apr. 1990.
- [17] C.Tomazou, F.J.Lidgely and R.Haigh. *Analog IC Design: the Current-Mode Approach*. Peter Peregrinus Ltd., 1990.
- [18] Z.Wang. *Current-Mode Analog Integrated Circuits and Linearization Techniques in CMOS Technology*. Hartung-Gorre Verlag, Konstanz. Series in microelectronics; vol. 7, 1990.
- [19] I.Baturone, A.Barriga, S.Sánchez-Solano, J.L.Huertas. "Una implementación Hardware de Circuitos Básicos de Lógica Difusa". *Actas del VII Congreso de Microelectrónica*, pp. 407-412, Toledo, 1992 (in spanish).
- [20] J.Tombs, *Multilayer Neural Networks and their Implementation in Analogue VLSI*, D.Phil. Thesis Trinity Term 1992, Oxford University Engineering Department.
- [21] J.Tombs, L.Tarassenko and A.Murray. "A Novel Analogue VLSI Design for Multi-Layer Networks". *IEE Special Issue on Artificial Neural Networks*, 1992.
- [22] J.M. Quero, J.M. Carrasco and L.G. Franquelo. "Adaptive Energy Feed-Back Control for Resonant Converter Using Neural Networks." *IEEE Power Electronics Specialist Conference*. vol. 2, pp. 800-806, Toledo Spain, 1992.
- [23] J.M. Carrasco. "Nuevas Implementaciones Electrónicas para el Control de Convertidores Resonantes" Ph. D. Dissertation, Escuela Superior de Ingenieros Industriales, Universidad de Sevilla, Diciembre 1992.
- [24] M.Sugeno Ed. *Industrial Applications of Fuzzy Control*. North-Holland, Amsterdam, 1985.
- [25] M.Ismail. "Four transistor continuous-time MOS transistor". *Electronic Letters*, vol. 23, no. 22, pp.1099-1100, Sept. 1997.
- [26] N.Khashab and M.Ismail. "Novel continuous-time all-MOS four quadrant multipliers". *Proc. IEEE Int. Symp. on CAS*, pp. 762-766, May 1987.
- [27] M.Ismail, R.Brannen, S.Tagaki, O.Aaserud, N.Fuji and N.Khashab. "A configurable CMOS multiplier/divider for analog VLSI". *Proc. IEEE Int. Symp. on CAS*, pp. 1085-1088, May 1993.