

SÍNTESIS DE SISTEMAS DE CONTROL DIFUSOS MEDIANTE HERRAMIENTAS DE DISEÑO DSP SOBRE FPGAS

S. Sánchez-Solano¹, M. Brox², A. Cabrera³

¹Instituto de Microelectrónica de Sevilla (CNM-CSIC). Sevilla, España.

²Dpto. Arquitectura de Computadores, Electrónica y Tecnología Electrónica. Univ. Córdoba, España.

³Dpto. Automática y Computación. CUJAE. Ciudad de La Habana, Cuba.

e-mail: santiago@imse.cnm.es

ABSTRACT

En esta comunicación se describe una nueva técnica de implementación de sistemas difusos, basada en el uso de una librería de módulos específicos, cuyo flujo de diseño combina las herramientas de modelado y simulación del entorno Matlab y las herramientas de síntesis e implementación de FPGAs de Xilinx. La estrategia propuesta aprovecha las ventajas de flexibilidad y facilidad de configuración que brindan las diferentes herramientas de Matlab y Xilinx, permitiendo acelerar considerablemente las etapas de descripción, síntesis y verificación funcional de los sistemas difusos bajo desarrollo.

1. INTRODUCCIÓN

A lo largo de los últimos años la lógica difusa ha demostrado ser una herramienta adecuada para emular los mecanismos de razonamiento utilizados por el cerebro humano [1]. La capacidad para describir el comportamiento de un sistema complejo a partir de la descripción lingüística del conocimiento de un operador experto, sin necesidad de emplear modelos matemáticos explícitos, ha motivado un incremento considerable del número de aplicaciones de control que emplean técnicas de inferencia basadas en lógica difusa [2]. Sin embargo, a pesar del grado de madurez de esta tecnología, existen actualmente muy pocas herramientas o entornos de diseño capaces de transformar la descripción de alto nivel de un sistema de inferencia difuso en una implementación hardware eficiente.

En la literatura especializada se recogen numerosas alternativas para la implementación hardware de controladores difusos mediante ASICs o dispositivos programables [3]. En particular, el constante desarrollo de las tecnologías de fabricación de circuitos integrados ha supuesto importantes avances en dispositivos lógicos programables como las FPGAs. Dicha transformación ha afectado, no solo al incremento del número de recursos de propósito general incluidos en los dispositivos, sino también a la introducción de elementos específicos, así

como a la existencia de numerosos bloques de sistema disponibles como módulos-IP o soft-cores que facilitan el desarrollo de sistemas de procesamiento adaptados a un determinado dominio de aplicación. Estos avances en las técnicas de implementación de sistemas microelectrónicos se complementan, asimismo, con la propuesta de nuevas metodologías de diseño y la disponibilidad de potentes herramientas de CAD que permiten acortar el ciclo de desarrollo de nuevos productos [4].

Esta comunicación presenta una novedosa estrategia de síntesis sobre FPGAs de sistemas de inferencia basados en lógica difusa. El flujo de desarrollo empleado hace uso de la herramienta *System Generator* (SysGen) de Xilinx dentro del entorno de modelado y simulación de Matlab. En la Sección 2 se describe brevemente el entorno de desarrollo. La librería de módulos específicos que facilita la síntesis de sistemas difusos mediante esta estrategia es detallada en la Sección 3. La Sección 4 ilustra la aplicación de la técnica propuesta al desarrollo de un controlador para el aparcamiento de un vehículo autónomo. Finalmente, las principales conclusiones de este trabajo se resumen en la Sección 5.

2. EL ENTORNO SYSGEN

El entorno SysGen facilita el desarrollo de algoritmos para sistemas de procesamiento digital de señal (DSP) sobre las FPGAs de Xilinx [5]. Está basado en Simulink, la herramienta interactiva para el modelado, la simulación y el análisis de sistemas dinámicos integrada en Matlab. SysGen incluye una librería de módulos Simulink (denominada "Xilinx Blockset") que proporciona los elementos básicos de un sistema digital, así como el software necesario para facilitar la síntesis e implementación sobre las FPGAs de Xilinx de los algoritmos descritos y simulados en el entorno Matlab.

El ciclo de diseño típico con esta herramienta comienza con la descripción esquemática del sistema mediante un modelo Simulink. La verificación se lleva a cabo mediante simulación y utilizando las distintas facilidades de generación y visualización de señales

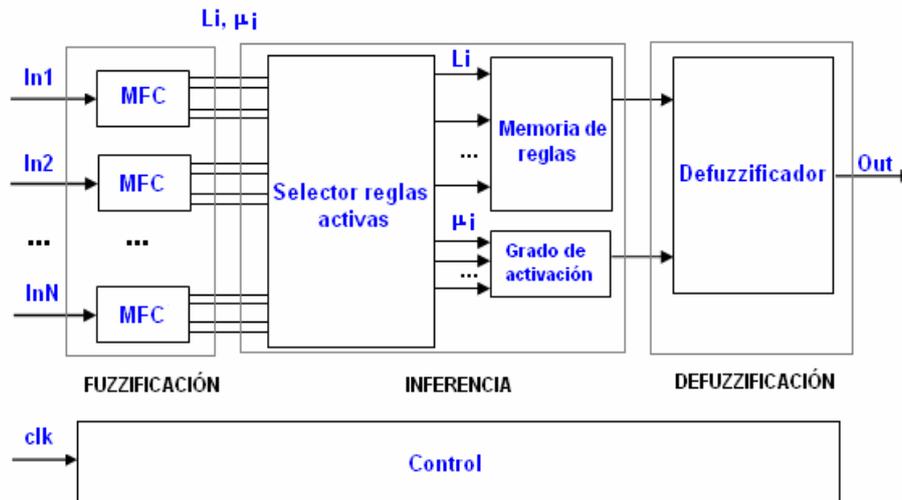


Fig. 1. Bloques constitutivos de una arquitectura de sistema difuso basada en reglas activas.

proporcionadas por Matlab. Una vez verificada la funcionalidad del diseño, SysGen lo traslada de forma automática en una implementación hardware optimizada en términos de área y velocidad.

3. LA LIBRERÍA XFUZZYLIB

La Figura 1 muestra los bloques que conforman la estructura general de una arquitectura eficiente para la implementación digital de sistemas difusos basada en el procesamiento de reglas activas [6]. Haciendo uso de la librería “Xilinx Blockset” y utilizando las facilidades proporcionadas por Simulink, se ha generado una librería de módulos especializados, a la que denominamos “Xfuzzy Blockset” o XfuzzyLib, que proporciona diferentes alternativas para implementar cada uno de los elementos básicos de esta arquitectura. En la Figura 2 aparece el navegador de librerías de Simulink con los diferentes módulos agrupados por bloques funcionales: generadores de funciones de pertenencia, selectores de reglas activas, conectivos de antecedentes, memorias de reglas, defuzzificadores y elementos de control.

A modo de ejemplo, la Figura 3 muestra el esquema de conexionado de un módulo generador de funciones de pertenencia mediante técnicas aritméticas. Los distintos módulos de la librería “Xilinx Blockset” admiten una serie de parámetros que definen detalles específicos de su funcionalidad, así como el tamaño o tipo de aritmética utilizada. Del mismo modo, una vez realizado el diagrama de bloques de un nuevo módulo de la librería XfuzzyLib, dicho módulo se encapsula como un “subsistema” y se añade una “máscara” para identificar los distintos parámetros que lo caracterizan. Cuando dicho subsistema es usado en un nivel jerárquico superior, estos parámetros pueden asignarse mediante valores numéricos o mediante variables a las que se les

asigna un valor numérico en la ventana de comandos de Matlab o a través de un fichero “.m” (Figura 4). La funcionalidad del diseño puede verificarse en todo momento con ayuda del simulador Simulink y sus diferentes facilidades para generar señales de excitación y para capturar y representar gráficamente los datos de salida. Utilizando los diferentes módulos de la librería, la construcción de un sistema difuso requiere únicamente la elección, interconexión y parametrización de los bloques necesarios. Una vez que se ha definido el sistema difuso de acuerdo a la arquitectura propuesta, puede seguirse el flujo de desarrollo descrito anteriormente para su implementación en una FPGA.

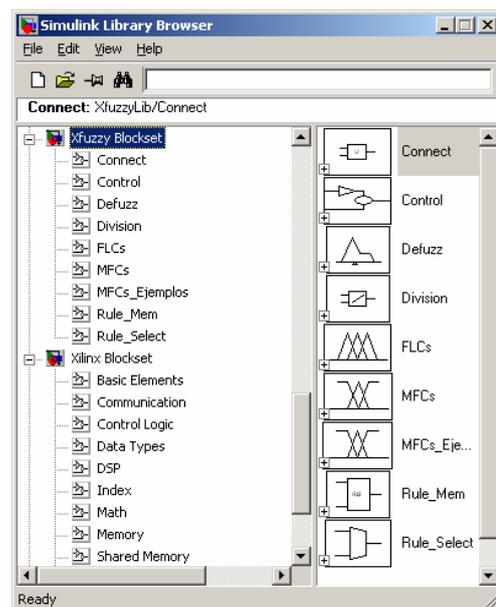


Fig. 2. Módulos de la librería XfuzzyLib integrados en el navegador de librerías de Simulink.

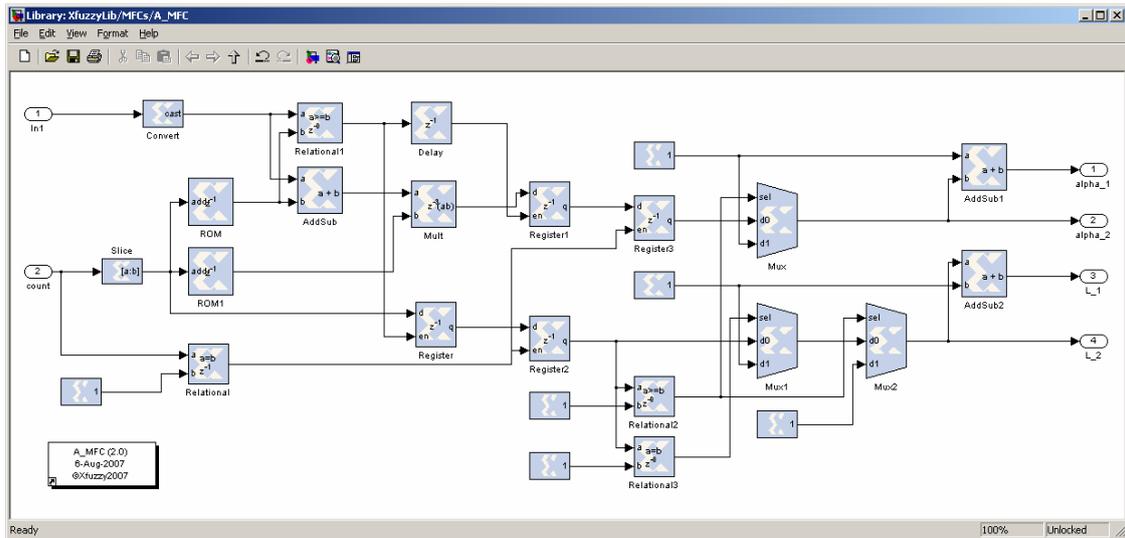


Fig. 3. Modelo Simulink del módulo de generación de funciones de pertenencia.

Además de los bloques que constituyen los elementos constructivos básicos de un sistema difuso, la librería XfuzzyLib incluye bloques con descripciones de arquitecturas tipo de módulos de inferencia difusos que se diferencian entre sí en el número de entradas, el conectivo que usan para calcular el grado de activación de las reglas y el método de defuzzificación empleado. Al desarrollar un sistema difuso para una aplicación determinada, el usuario puede utilizar estas arquitecturas tipo o bien definir nuevas arquitecturas mediante la interconexión de los bloques de elementos básicos que sean necesarios. Asimismo, es posible combinar de forma jerárquica estos módulos de inferencia para definir sistemas difusos de mayor complejidad.

La Figura 5 muestra el conjunto de controladores difusos tipo (FLCs) actualmente incluidos en la librería XfuzzyLib. En la Figura aparece el diagrama de bloques de un FLC de dos entradas que emplea el producto como conectivo y el método de defuzzificación FuzzyMean. Como ocurre con los bloques básicos de la librería, los bloques que describen arquitecturas de FLCs son totalmente parametrizables, por lo que el diseñador de sistemas difusos puede adaptar la funcionalidad del módulo a los requerimientos del problema planteado sin más que definir los parámetros correspondientes al dimensionado del sistema de inferencia y a la definición de la base de conocimiento del mismo. Análogamente a los bloques básicos de la librería, también aquí es

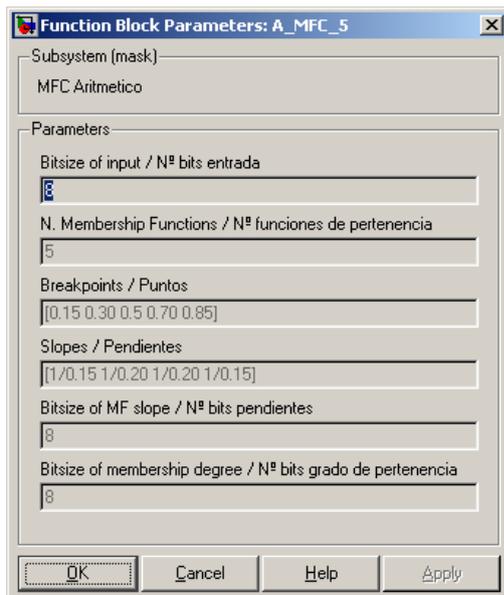


Fig. 4. Parámetros del módulo de generación de funciones de pertenencia.

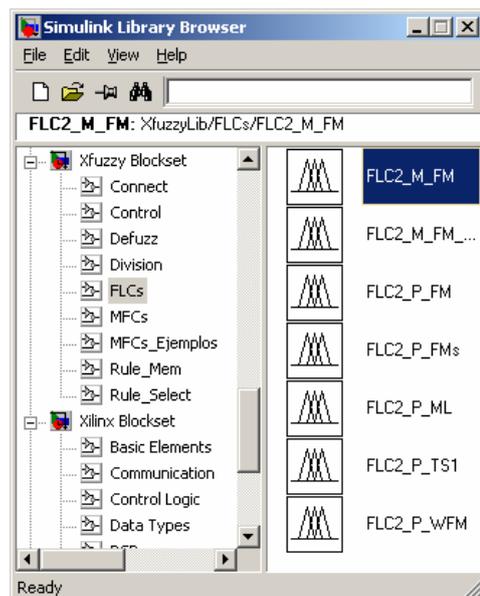


Fig. 5. FLCs tipo disponibles en la librería XfuzzyLib.

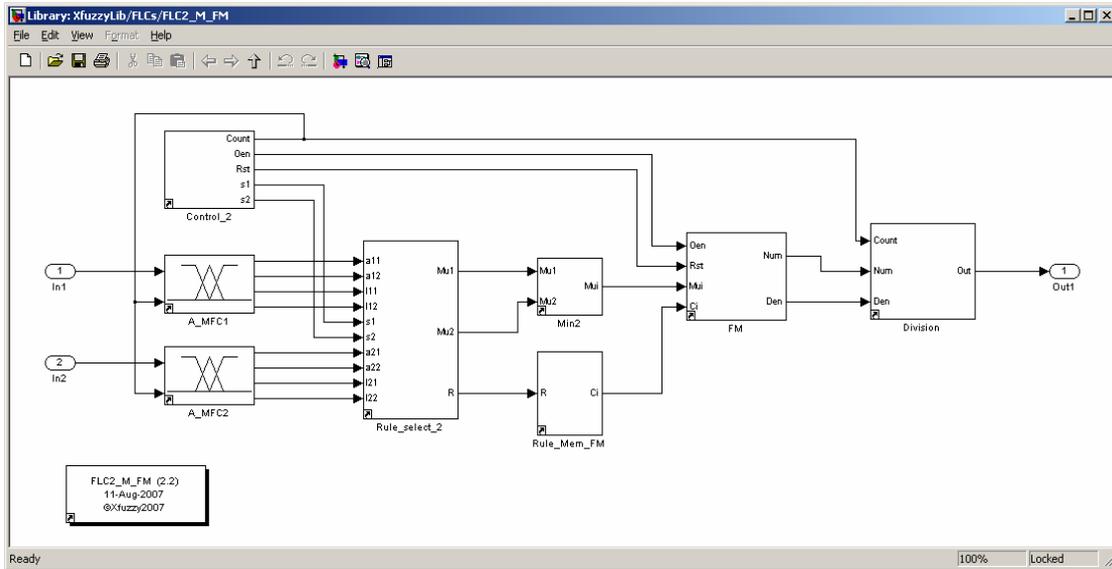


Fig. 6. Diagrama de bloques de un FLC con “mínimo” como conectivo y FuzzyMean.

posible aprovechar toda la funcionalidad del entorno Matlab para verificar la funcionalidad del sistema de inferencia.

La incorporación en el modelo del bloque “System Generator” facilita la síntesis hardware del diseño descrito en Simulink mediante la traslación del modelo del FLC a diferentes tipos de netlists, la generación del fichero de programación de la FPGA o, incluso, la verificación funcional del controlador en lazo cerrado a través de la cosimulación hardware/software de la implementación del sistema de control junto a un modelo Simulink de la planta.

4. APLICACIÓN A UN SISTEMA DE CONTROL PARA ROBÓTICA

La técnica de implementación descrita ha sido aplicada a la realización de un sistema de control difuso para el

aparcamiento de un vehículo autónomo. En este caso el vehículo puede estar ubicado en una posición cercana a la plaza de aparcamiento, por lo que la trayectoria a seguir debe combinar desplazamientos hacia delante y hacia atrás para alcanzar la posición objetivo [7]. El vehículo utilizado es un robot eléctrico dotado de distintos sensores que permiten calcular los valores de la posición, orientación y velocidad en cada momento. La salida del controlador actúa sobre los motores de tracción y dirección del vehículo para fijar los valores adecuados de velocidad y ángulo de giro de las ruedas.

El sistema de control propuesto utiliza una base de conocimiento jerárquica que incluye módulos de toma de decisiones (para discriminar si el vehículo debe circular en un sentido u otro) y módulos de control (para calcular la velocidad y el ángulo de giro de las ruedas). Para desarrollar esta estructura del controlador se ha hecho uso del entorno *Xfuzzy* [8], que permite describir y

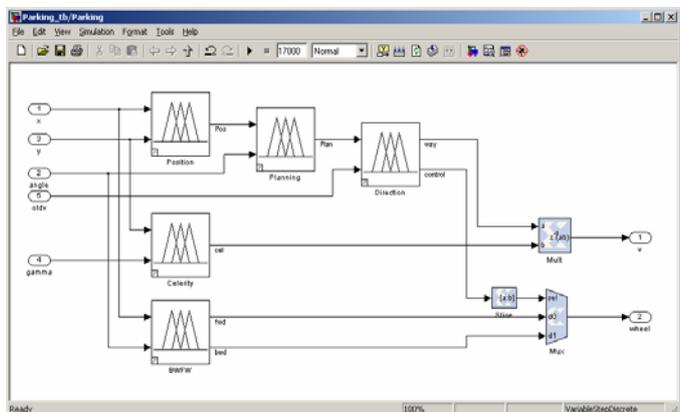
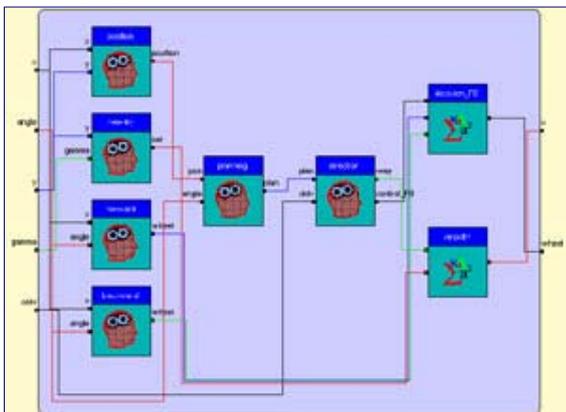


Fig. 7. Sistema de control difuso para el problema de aparcamiento de un vehículo autónomo: descripción del controlador en *Xfuzzy* y modelo Simulink con los módulos de la librería *XfuzzyLib*.

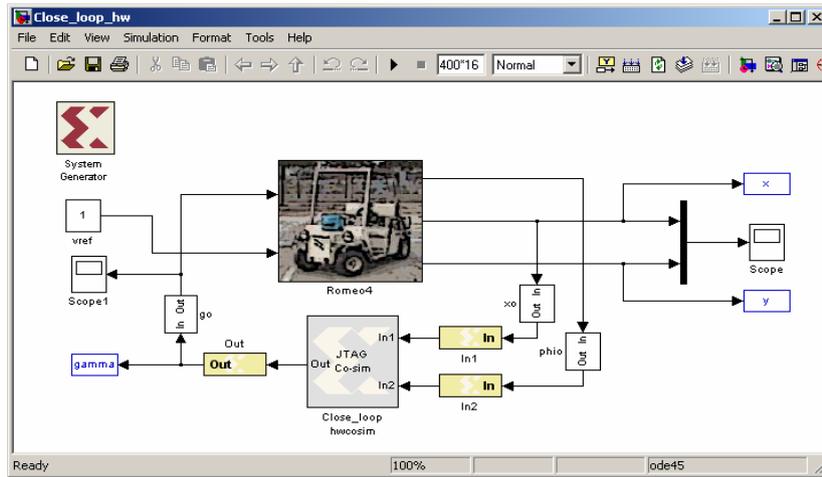


Fig. 8. Modelo Simulink para la verificación funcional del controlador mediante cosimulación hardware/software.

simular sistemas difusos con estructura jerárquica. La traslación de la especificación de un sistema difuso en el entorno *Xfuzzy* al flujo de diseño propuesto en este trabajo requiere la identificación y la definición de los parámetros de los bloques proporcionados por la librería *XfuzzyLib*. La Figura 7 compara la representación del controlador en ambos entornos de desarrollo.

Aprovechando la funcionalidad que proporciona SysGen, la verificación funcional del sistema de control puede llevarse a cabo mediante la integración en un lazo cerrado de cosimulación de un modelo software del robot descrito en Matlab y la implementación hardware del controlador sobre la FPGA. La Figura 8 muestra el modelo usado para verificar el sistema completo, mientras que la Figura 9 incluye alguno de los resultados experimentales obtenidos.

5. CONCLUSIONES

En esta comunicación se presenta una técnica de implementación de sistemas difusos sobre FPGAs

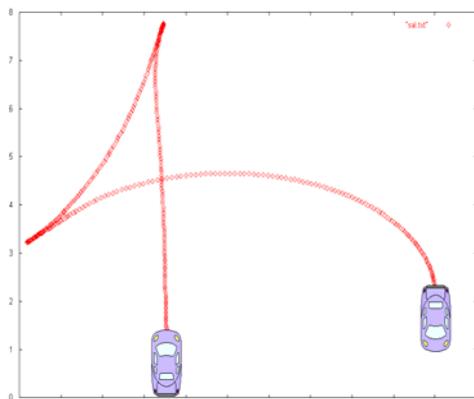


Fig. 9. Resultados experimentales.

basada en el uso de la herramienta de diseño DSP de Xilinx. Al estar integrado en el entorno Matlab, el flujo de diseño propuesto presenta ventajas significativas en cuanto a flexibilidad y configurabilidad, y permite el uso de las distintas facilidades de cálculo y representación gráfica que proporciona dicho entorno. La estrategia propuesta será la base de una nueva herramienta de síntesis para la próxima versión del entorno *Xfuzzy*, aportando como principales novedades la inclusión de nuevos operadores y métodos de defuzzificación, una funcionalidad mejorada de muchos de los elementos básicos de la arquitectura y la implementación directa de sistemas difusos con estructuras jerárquicas.

REFERENCIAS

- [1] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 3, No. 1, pp. 28-44. Jan. 1973.
- [2] T. J. Ross, "Fuzzy Logic with Engineering Applications", 2nd Edition, Wiley, 2004.
- [3] I. Baturone, A. Barriga, S. Sánchez-Solano, C.J. Jiménez, D. López, "Microelectronic Design of Fuzzy Logic-Based Systems", CRC Press, 2000.
- [4] W. Savage, J. Chilton, R. Camposano. "IP reuse in the System on a Chip Era". 13th Int. Symp. on System Synthesis, pp. 2-7. 2000.
- [5] Xilinx. "Xilinx System Generator v8.2 for Simulink". User Guide. Xilinx Blockset Reference Guide.
- [6] A. Cabrera, S. Sánchez-Solano, C. J. Jiménez, A. Barriga, I. Baturone, "Arquitectura eficiente para la implementación hardware de sistemas de inferencia difusos", *Ingeniería Electrónica, Automática y Comunicaciones*, Vol. XXIII, No. 1, pp. 59-66, 2003.
- [7] S. Sánchez-Solano, A. Cabrera, I. Baturone, F. J. Moreno-Velo, M. Brox, "FPGA Implementation of Embedded Fuzzy Controllers for Robotic Applications". *IEEE Trans. on Industrial Electronics*, Vol. 54, No. 4, pp. 1937-1945. Aug. 2007.
- [8] *Xfuzzy: Fuzzy Logic Design Tools*, (<http://www.imse.cnm.es/Xfuzzy>)