

CLASSIFICATION OF CHEMICAL COMPOUNDS USING A MULTISENSORIAL SYSTEM (ELECTRONIC NOSE), DEVELOPED ON A HARDWARE DEVICE (FPGA)

Article Information:

Received: October 21, 2013

Accepted: December 23, 2013

Keywords: Multisensory systems, Neural networks, Gas sensors, VHDL, FPGA

Abstract: In multisensory systems specifically for the operation of the current electronic noses, the computer is the most widely used for the processing and classification of data from an array of chemical gas sensors. The main objective of this study is to replace the usual computer by using a hardware device, in order to obtain a system that is more portable and provides results in real time through the acquisition, processing and samples classification. In developing the system, the design of an algorithm in VHDL (Very High Speed Integrated Circuit - Hardware Description Language) to be implemented in a FPGA (Field Programmable Gate Array) Spartan 3E was performed. The results were as expected, since the module developed different experimental tests on both a computer and the FPGA is performed, obtaining a classification rate of 100% success.

CLASIFICACIÓN DE COMPUESTOS QUÍMICOS USANDO UN SISTEMA MULTISENSORIAL (NARIZ ELECTRÓNICA) DESARROLLADA SOBRE UN DISPOSITIVO HARDWARE (FPGA)

Oscar Eduardo Gualdrón Guerrero¹, Ph.D., Cristhian Manuel Durán Acevedo², Ph.D., José Antonio Araque Gallardo, M.Sc., Jesús Eduardo Ortiz Sandoval³, M.Sc. (c)

Grupo de Investigación en sistemas Multisensoriales y Reconocimiento de Patrones, Universidad de Pamplona, Pamplona, Colombia.

E-mail: {oscar.gualdrón¹, cmduran², jesus.ortiz³}@unipamplona.edu.co

Información del artículo:

Recibido: 21 de octubre de 2013

Aceptado: 23 de diciembre de 2013

Palabras Claves: Sistemas multisensoriales, Redes neuronales, Sensores de gases, VHDL, FPGA.

Resumen: En los sistemas multisensoriales, específicamente para el funcionamiento de las narices electrónicas actuales, el computador es el más utilizado para realizar el procesamiento y la clasificación de los datos provenientes de una matriz de sensores de gases químicos. El objetivo principal del presente estudio es sustituir la computadora habitual mediante el uso de un dispositivo hardware, con el fin de obtener un sistema que sea más portable y que proporcione resultados en tiempo real a través de la adquisición, procesamiento y clasificación de las muestras. En el desarrollo del sistema, se realizó el diseño de un algoritmo en lenguaje VHDL (Very High Speed Integrated Circuit - Hardware Description Language) para ser implementado en una FPGA (Field Programmable Gate Array) Spartan 3E de Xilinx. Los resultados fueron los esperados, ya que con el módulo desarrollado se realizaron diferentes pruebas experimentales tanto en una computadora como en el FPGA, obteniendo un porcentaje de clasificación del 100 % de acierto en la clasificación de algunos compuestos químicos.

1. INTRODUCCIÓN

La nariz electrónica es una investigación que se ha venido consolidando en las últimas décadas, con el fin de ser aplicado en casi todos los sectores (es decir, salud, procesamiento de productos, control de calidad, industria química, etc.) Es importante resaltar que aún se siguen desarrollando estudios donde se han encontrado algunas limitantes que afectan a estos sistemas de medida y a su vez se han buscado alternativas importantes para la solución de algunos problemas frecuentes[1].

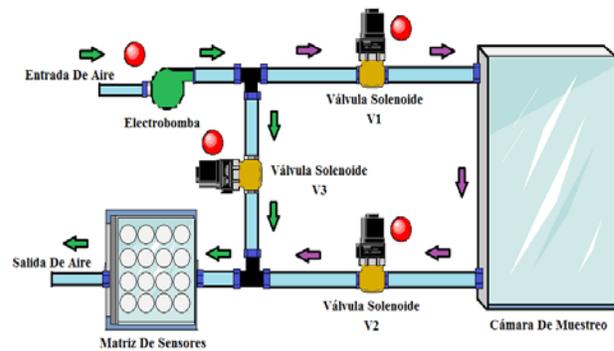
En términos generales, la nariz electrónica es un dispositivo compuesto por cuatro etapas funcionales [2]:

- La primera realiza la adecuación de la mezcla gaseosa y el muestreo.
- El conjunto de sensores de gases hace la detección de los compuestos volátiles.
- La electrónica de control se dedica a la gestión del conjunto de sensores y adecuación de la señal.
- Finalmente, se hace uso de una computadora donde se extraen los rasgos característicos o “huellas” de cada aroma y se aplican algoritmos de clasificación de patrones (Ej.: Redes Neuronales Artificiales) y (Análisis de Componentes Principales (PCA, por sus siglas en inglés)).

Esta configuración se sigue aún implementando, ya que con este diseño basado en los sistemas multisensoriales, se tiene como objetivo principal la clasificación de diversos compuestos. Cabe resaltar que el presente artículo va enfocado a utilizar dispositivos avanzados (es decir, Field Programmable Gate Array o FPGA), que puedan ser implementados en sistemas de olfato electrónico. En la Figura 1 se observa un esquema típico de una nariz electrónica que es acondicionada a través de un sistema de muestreo con flujo continuo, el cual fue una referencia importante para el desarrollo del presente estudio [3].

El rendimiento de los sistemas de olfato electrónico puede ser optimizado introduciendo el uso de nuevas tecnologías. Un limitante importante de estos sistemas, es que se debe realizar el procesamiento de la información en el computador, lo que eleva el nivel de dificultad en el momento en que se necesiten realizar pruebas de campo. Basados en esta premisa se propone diseñar un módulo compuesto de una tarjeta FPGA, que pueda realizar en forma óptima el proceso de clasificación y mando del sistema multisensorial.

Figura 1. E-Nose Universidad de Pamplona.



Fuente: los autores, 2013

2. METODOLOGÍA

En el siguiente apartado se describen cada uno de los métodos utilizados para el desarrollo del sistema embebido.

2.1 Redes Neuronales

Las Redes Neuronales Artificiales, ANN (*Artificial Neural Networks*) están inspiradas en las redes neuronales biológicas del cerebro humano. Están constituidas por elementos que se comportan de forma similar a la neurona biológica en sus funciones más comunes. Estos elementos están organizados de una forma parecida a la que presenta el cerebro humano [4].

Las redes neuronales al margen de parecerse al cerebro, presentan una serie de características propias. Por ejemplo, las ANN aprenden de la experiencia, generalizan de ejemplos previos a otros nuevos y abstraen las características principales de una serie de datos.

Para realizar este proceso se deben cumplir las siguientes etapas: *Aprender*: adquirir el conocimiento de una cosa por medio del estudio, ejercicio o experiencia. Las redes neuronales pueden cambiar su rendimiento en función del entorno de aprendizaje. *Generalizar*: las redes neuronales generalizan automáticamente debido a su propia estructura y naturaleza. Estas redes pueden ofrecer, dentro de un margen determinado, respuestas correctas a entradas que presentan pequeñas variaciones debido a los efectos de ruido. *Abstraer*: las redes neuronales por su esencia capturan la información importante de las entradas.

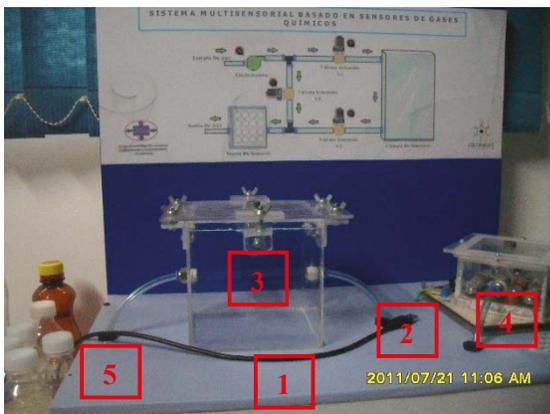
2.2 Nariz Electrónica

Un concepto muy general de la nariz electrónica es la de un instrumento que consiste en un conjunto de sensores electroquímicos con una especificidad parcial y un sistema adecuado de reconocimiento de patrones, capaz de reconocer olores simples o complejos [5].

En la Figura 2 se aprecia el sistema de nariz electrónica disponible para el desarrollo de este proyecto, la cual consta de las siguientes partes:

1. Bomba de aire y válvulas de paso eléctricas.
2. Circuito de control de potencia.
3. Cámara de concentración de volátiles.
4. Cámara con sensores de gas.
5. DAQ de 16 Canales

Figura 2. Nariz Electrónica (Universidad de Pamplona)



Fuente: los autores, 2013

2.3 Field Programmable Gate Arrays (FPGA)

La FPGA hizo su primera aparición en la mitad de la década de los 80. La diferencia de estos dispositivos con respecto a la CPLD se encuentra en su arquitectura, tecnología, construcción y costos, la FPGA soporta la implementación de circuitos de gran tamaño y muy buena eficiencia. La tarjeta de desarrollo Spartan 3E, Xilinx es fabricada por Digilent y aparte de la FPGA incluye muchas herramientas que hacen al kit de desarrollo muy útil y completo para diversas aplicaciones.

2.4 Preparación de las muestras

Para realizar la adquisición del conjunto de medidas de entrenamiento fueron empleados los compuestos volátiles: Benceno, Tolueno, Xileno, Etanol y Metanol.

2.5 Captura de datos

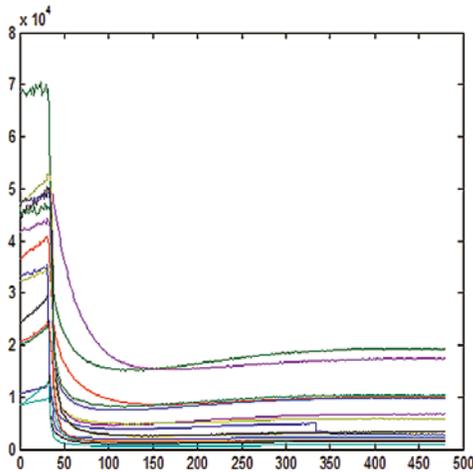
La tarjeta DAQ captura los valores de voltaje proveniente de cada sensor, un algoritmo acondiciona la señal y a su vez calcula la resistencia correspondiente mediante la ecuación (1). Donde R_{Ω} es el valor de resistencia muestreado por la tarjeta de adquisición de datos, V_c es el voltaje de alimentación del sensor que equivale a 10V y V_{in} es el voltaje presente en la salida del sensor al momento del muestreo de información.

$$R_{\Omega} = \left(\frac{V_c}{V_{in}} - 1 \right) * 4700 \quad (1)$$

El algoritmo almacena los datos en un archivo plano compatible con Matlab para continuar el análisis de la muestra. La tarjeta NI 6014 está acondicionada con 16 canales analógicos y hace el muestreo cada segundo, utilizando un multiplexor de salidas para evitar que haya colisiones y se pierda información. Tal y como se observa en la Figura 3, al inicio de cada una de las señales de los sensores ocurre una leve fluctuación del estado del sensor y a su vez se presenta el cambio que se produce al detectar un volátil. El total de tiem-

po adquirido fue de 480 segundos que equivale a los ocho minutos configurados en el software.

Figura 3. Data de Benceno.



Fuente: los autores, 2013

Los hidrocarburos al estar conformados en su mayoría por átomos de carbono tienen características muy parecidas que al momento de hacer clasificación se puede presentar un solapamiento, la clave para evitar este suceso es tomar adecuadamente las muestras.

Una señal se puede manipular de diferentes maneras, algoritmos complejos, extracción de parámetros, potencia, energía, mediana, promedio y muchas más herramientas que calculan la información vital de la señal. En el software de Matlab se necesita de un patrón estático o de sensibilidad para poder crear la red neuronal. La base de datos que se toma y se modifica posee los valores de resistencia de cada uno de los sensores, pero el patrón que se estima puede ser mejor para la red neuronal es el de Conductancia Máxima dividida por la Conductancia Mínima, un par de algoritmos se encargan de crear el vector con este parámetro estático.

2.6 Red MLP en Matlab

Matlab® cuenta con la *toolbox ANN* que permite trabajar con redes neuronales artificiales multicapa. En un algoritmo sencillamente se asigna

con qué dígitos, vectores o matrices la red va a entrenar y con cuáles va a validar. Queda a disposición del usuario configurar cuantas neuronas va a utilizar, función de activación de cada una de las capas, iteraciones entre neuronas, etc. Una vez la *toolbox* ha finalizado el proceso en una estructura llamada *net* entrega toda la información de la red, y ahí es donde precisamente se obtienen los pesos y los umbrales para llevarlos al código fuente.

La metodología desarrollada para el entrenamiento de la ANN fue la siguiente:

1. Adquisición de los datos de cada compuesto químico, cada uno con la misma cantidad de tiempo de concentración y tiempo de captura.
2. Pre-procesamiento de la información con algoritmos para calcular la conductancia de cada sensor y parametrizar por cada entrada la conductancia máxima y mínima.
3. Con el parámetro establecido para cada muestra se construye la matriz con todas las muestras y parámetros para realizar el entrenamiento de la ANN.
4. Una vez finalizado el entrenamiento se validan los resultados con nuevos datos adquiridos y se verifica que el porcentaje de error sea 0.
5. Con la ANN definida se obtienen los pesos de las neuronas para implementar en el dispositivo lógico programable.

2.7 Estructura de la programación VHDL

Para realizar un diseño eficiente en VHDL [6] es necesario jerarquizar el diseño pues contiene muchas etapas importantes, para lograr este objetivo se diseñan unos componentes que permiten fraccionar el código.

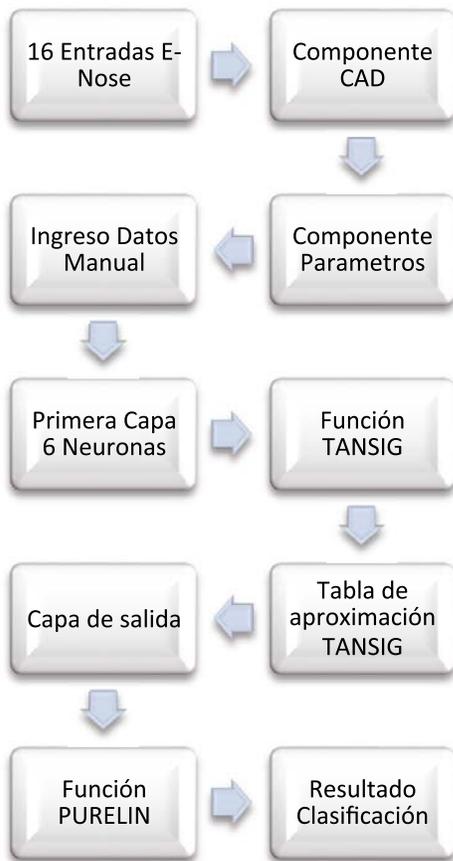
Para utilizar un componente primero éste debe ser declarado, ya que por la facilidad y el orden en que quede estructurado el archivo, la programación del proyecto se divide en tres componentes, CAD o conversor análogo digital, Extracción de parámetros y la red neuronal. Los tres componen-

tes se conectan mediante la función MAIN sistema propuesto.

2.8 Modelo neuronal diseñado

En la red neuronal aparecen las seis neuronas de la capa oculta y la neurona de la capa de salida, la función de activación para las primeras es Tansig y para la de salida es Purelin.

Figura 4. Proceso para implementar la ANN en el dispositivo lógico programable.



Una vez que el vector con los patrones está listo, pasa por las seis primeras neuronas donde se multiplican por sus pesos, se le suma el umbral y se calcula la función Tansig, la salida de cada neurona de la capa oculta es una entrada de la neurona de salida. En la neurona de salida las señales provenientes de la capa oculta se multiplican por los pesos se suma el umbral y la salida al ser Purelin, es el mismo resultado. Para poder graficarlo exactamente se redondea con condicionales y se envía por un puerto al Microcontrolador.

La función de activación Tansig es comúnmente usada en redes neuronales multicapa que son entrenadas por el algoritmo backpropagation. La función es compleja de implementar en el hardware digital debido a que usa serie infinita de exponenciales [7]. Una de las maneras más sencillas para poder implementar la función Tansig en VHDL es usando una tabla de aproximación de valores, pero puede consumir gran parte de recursos computacionales. Esta tabla de aproximación es una forma de implementar dicha función “on-chip”, pero no es la más recomendada, por ende, para asegurar la correcta clasificación del conjunto de datos se construyó una tabla de aproximación para emular la función de activación.

En esta parte del algoritmo, el cual fue diseñado, se da una muestra de cómo funciona una neurona en VHDL no se escribe todo el código, debido a que es muy largo, pero acá se describe en general, la multiplicación de las entradas, la suma con el umbral y la función de activación “Tansig”, además, se diseñan otros componentes que son de vital importancia en el módulo de clasificación.

3. RESULTADOS

3.1 Prueba con datos reales

La prueba para poder verificar que el algoritmo está funcionando correctamente es relativamente sencilla. A la matriz de parámetros se le asignan valores ya conocidos de las datas con las que se hizo el proceso de entrenamiento, como la FPGA no maneja el punto flotante y se escalan los números para eliminar el punto, este aspecto también se maneja y se ingresan los parámetros como números enteros.

Una vez que se valida el funcionamiento del módulo con los datos conocidos, se realizaron pruebas funcionales con el módulo y el sistema multisensorial que se observa en la Figura 5, efectuando mando y captura de información.

Figura 5. Módulo ensamblado con E-Nose.



Fuente: los autores, 2013

Cuando se finaliza la adquisición, el módulo realizó el procesamiento de la información tomando el parámetro de conductancia máxima de cada sensor. El algoritmo alcanza las 850 líneas de programación, pero con un consumo de recursos bajo ya que en esta instancia no se están manejando ni matrices de gran tamaño, ni ciclos que representaran una carga alta de memoria y recursos lógicos.

Como se describe en la segunda sección, la FPGA no soporta en sus estándares oficiales la utilización del punto fijo o punto flotante, por tal motivo al tener en el proyecto todos los valores con punto fijo, se hizo necesario escalar todos los números, y en el momento que se hacen algunas operación se busca mantener al dato alejado de los límites definidos de 31 bits para los tipos integer, si ocurriese un desbordamiento todo el proceso de clasificación se dañaría y daría resultados inesperados. Con base en este inconveniente, y como el código se encuentra modulado, se realizó el planteamiento de fraccionar el programa. En un primer instante el módulo se encarga de realizar el mando y adquisición de las muestras, cuando ha finalizado este proceso la FPGA realiza el procesamiento de la información y a través de la pantalla se entregan los parámetros encontrados de cada sensor.

En la figura 6 se aprecia el resultado de esta operación, este programa se aloja en la memoria vo-

látil de la tarjeta. Con los parámetros obtenidos, se realiza la clasificación de las muestras, el algoritmo correspondiente se aloja en la EEPROM de la tarjeta, por lo que una vez finalizó la visualización de los parámetros, se borra la memoria volátil y se ejecutó el algoritmo almacenado en la EEPROM con la red neuronal entrenada. Se ingresaron los parámetros encontrados por la tarjeta en su etapa de procesamiento y el dispositivo ejecuta la red neuronal y el resultado se visualiza en la pantalla que se puede comprobar en la figura 7.

Figura 6. Resultado de la adquisición de datos



Fuente: los autores, 2013

Figura 7. Resultado de la red neuronal



Fuente: los autores, 2013

3.2 Resultados y porcentaje de error

El resultado de la clasificación una vez finalizó el proceso se visualiza en la pantalla y también en los diodos LED de la tarjeta FPGA en código binario, que es el formato en el cual se transmite la información al firmware del módulo ensamblado.

Se realizaron 11 pruebas con las diversas sustancias químicas objeto de estudio y para validar los resultados se analizaron los datos tanto en el módulo directamente, consultando los indicadores

LED de la FPGA como en el PC con el software MATLAB. Al momento de la programación, se definen las 5 clases de sustancias químicas, y cada una toma un valor binario entre 001(Benceno), 010 (Tolueno), 011(Xileno), 100 (Metanol) y 101(Etanol). En la tabla 1 se observan los resultados de cada prueba, que al analizar el resultado en la FPGA se obtiene el resultado en binario como en el PC donde el software B-Nose realiza la clasificación de las muestras. El porcentaje de error obtenido fue del 0%, las once muestras aleatorias que se guardaron como constantes en el algoritmo dieron una clasificación positiva, sin importar el orden ni que se repitió de una misma clase dos muestras.

Tabla 1. Resultados y comparación entre un PC y el sistema diseñado.

#	Tipo de la muestra	Resultado FPGA	Resultado PC
1	Metanol	"100" Metanol	Metanol
2	Benceno	"001" Benceno	Benceno
3	Etanol	"101" Etanol	Etanol
4	Benceno	"001" Benceno	Benceno
5	Tolueno	"010" Tolueno	Tolueno
6	Metanol	"100" Metanol	Metanol
7	Etanol	"101" Etanol	Etanol
8	Xileno	"011" Xileno	Xileno
9	Xileno	"011" Xileno	Xileno
10	Tolueno	"010" Tolueno	Tolueno
11	Benceno	"001" Benceno	Benceno

4. CONCLUSIONES

VHDL al ser un lenguaje de descripción de hardware estándar, permite que los diseños puedan ser migrados a un dispositivo ASIC (Circuito Integrado de Aplicación Específica). Además, a diferencia de los trabajos comúnmente realizados con RNA donde estos algoritmos se implementan en software, en este trabajo se desarrollan en hardware.

Con la implementación de la red neuronal diseñada en el hardware del dispositivo lógico programable permitió niveles muy altos de exigencia.

Por lo general, en implementación de circuitos en la tarjeta SPARTAN 3E se alcanzan porcentajes de utilización de 10, 20 ó 30% de los componentes internos, mientras que en los componentes diseñados se supera la capacidad del dispositivo hasta en un 1000%. Esto permite identificar limitantes del hardware y plantear nuevas investigaciones alrededor de la implementación de técnicas inteligentes en dispositivos lógicos programables.

El lenguaje VHDL permite una vez que se ha probado el proyecto en la tarjeta de desarrollo, la construcción de un circuito integrado según las especificaciones del diseño. Esto abre las puertas a un posible registro de patente del olfato electrónico, pues Xilinx puede fabricar el integrado que realice las funciones de la tarjeta de desarrollo, en el control del módulo y la clasificación de las muestras. El integrado puede ser del tamaño deseado para montaje superficial o para manejo en protoboard. Para posteriores mejoras o desarrollos del B-Nose sería muy interesante incluir, en lugar de una tarjeta de desarrollo, el integrado diseñado específicamente para la aplicación correspondiente.

La red neuronal entrenada e implementada en el dispositivo dio unos resultados increíbles al momento de clasificar los compuestos químicos objetos de estudio. A pesar de los inconvenientes encontrados al momento de realizar el diseño de la implementación, y los diversos métodos utilizados para lograr unir todos los componentes se obtuvo que en todas las pruebas realizadas no se encontró ningún porcentaje de error. Los resultados se comparan con la clasificación realizada por el computador y no se encuentra ninguna diferencia, por lo que el módulo es competente para realizar el trabajo de clasificación y manejo del sistema de olfato electrónico.

Para una investigación futura se deja como recomendación la utilización de un dispositivo que soporte el último estándar VHDL que es el 2008, este permite el uso en la implementación de tipos de números punto fijo y operaciones aritméticas

con variables, el proceso de la extracción de parámetros y de la red neuronal se facilita en un porcentaje muy alto.

REFERENCIAS

- [1] M. Holmberg, F. Winqvist, I. Lundström, F. Davide, C. DiNatale, A. D'Amico, Drift counteraction for an electronic nose, *Sensors and Actuators B: Chemical*, Vol. 36, Issues 1-3, 1996, pp. 528-535.
- [2] J. Rodríguez and C. Durán, Sistema de olfato electrónico para la detección de compuestos volátiles, *Revista Colombiana de Tecnologías de Avanzada*. Vol. 2, No. 12., 2008, pp. 20-26.
- [3] C. Durán. O. Gualdrón and L. Rojas, Sistema de olfato electrónico de bajo costo para la detección de diferentes compuestos químicos contaminantes, *Revista Colombiana de Tecnologías de Avanzada*, Vol. 1, No. 17, 2011, pp. 121-126.
- [4] M. Basualdo, D. Matich and C. Ruiz, (2001, marzo) Redes neuronales artificiales y sus aplicaciones, Universidad Tecnológica Nacional, Facultad Regional Rosario, Departamento de Ingeniería Química, [On-line] pp. 55. Disponible en http://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograias/matich-redesneuronales.pdf.
- [5] J.W. Gardner, P. Bartlett, Performance definition and standardization of electronic noses, *Sensors and Actuators B: Chemical*, Vol. 33, Issues 1-3, 1996, pp. 60-67.
- [6] V. Pedroni, *Circuit Design with VHDL*, MIT Press, 2004, pp. 17-19.
- [7] R. Khalil and S. A. Kazzaz, (2008, diciembre), Digital Hardware Implementation of Artificial Neuron Model Using FPGA, Department of Electrical Engineering, University of Mosul, Mosul, Iraq, [On-line] pp. 12-24. Disponible en <http://www.iasj.net/iasj?func=fulltext&aId=38764>.