

Estimación de distancia con sensores ópticos reflexivos usando redes neuronales con funciones de base radial para aplicaciones embebidas¹

Distance Estimation with Reflective Optical Sensors Using Neural Networks with Radial Basis Functions for Embedded Applications²

Estimação de distância com sensores óticos reflexivos usando redes neurais com funções de base radial para aplicações embutidas³

*Juan Sebastián Botero-Valencia⁴
Luis Javier Morantes-Guzmán⁵*

SICI: SICI: 0123-2126(201301)17:1<27:EDDSOR>2.0.TX;2-L

¹ Fecha de recepción: 27 de enero de 2012. Fecha de aceptación: 27 de julio de 2012. Este artículo se deriva de un proyecto de investigación denominado *Medición de velocidad en nadadores usando acelerómetros triaxiales* (código P11104), desarrollado por el grupo de investigación Automática y Electrónica, del Instituto Tecnológico Metropolitano, Medellín, Colombia, en convenio con la Liga de Natación de Antioquia.

² Reception date: January 27th 2012. Admission date: July 27th 2012. This paper originates from a research project called *Medición de velocidad en nadadores usando acelerómetros triaxiales* (Code P11104), carried out by the Automatic and Electronic research group of the Instituto Tecnológico Metropolitano in Medellín, Colombia, together with the Antioquia Swimming League.

³ Data de recepção: 27 de janeiro de 2012. Data de aprovação: 27 de julho de 2012. Este artigo origina-se de um projeto de pesquisa denominado *Medición de velocidad en nadadores usando acelerómetros triaxiales* (*Medição de velocidade em nadadores usando-se acelerômetros triaxiais*) (código P11104), desenvolvido pelo grupo de pesquisa *Automática y Electrónica* [Automática e Eletrônica], do Instituto Tecnológico Metropolitano, Medellín, Colômbia, em convênio com a *Liga de Natación* de Antioquia.

⁴ Ingeniero electrónico, Universidad Nacional de Colombia. Magíster en Automatización y Control Industrial, Instituto Tecnológico Metropolitano, Medellín, Colombia. Docente tiempo completo-modalidad ocasional, Instituto Tecnológico Metropolitano. Correo electrónico: juanbotero@itm.edu.co.

⁵ Ingeniero electrónico, Universidad Nacional de Colombia. Magíster en Ingeniería-Automatización Industrial, Universidad Nacional de Colombia. Profesor asistente, Facultad de Ingenierías, Instituto Tecnológico Metropolitano, Medellín, Colombia. Correo electrónico: luismorantes@itm.edu.co.

Resumen

Los sensores de reflexión ópticos para medir distancia se caracterizan por la no linealidad de la salida. Esta condición deben asumirla las unidades de procesamiento donde son utilizados. El presente artículo tiene por objeto reducir el costo computacional, en términos de espacio de memoria y tiempo de procesamiento, en la linealización de la salida de los sensores para ser usados en sistemas embebidos de bajo costo. Los resultados de dos diferentes sensores indican que la curva exponencial estimada se puede ajustar usando redes neuronales con funciones de base radial con una reducción de más del 50% en el tiempo de procesamiento. Para el diseño de la red se estudió la eficiencia respecto al tamaño de la red y a la distribución de los centroides con resultados importantes. La reducción del tiempo y de espacio lograda permite aprovechar los recursos del sistema en otras tareas, al mismo tiempo que posibilita aumentar la frecuencia de muestreo en la adquisición.

Palabras clave

Linealización de sensores, sensores ópticos de distancia, redes neuronales con funciones de base radial.

Abstract

Reflective optical sensors that measure distances are characterized by the non-linearity of their output. Processing units in which these are used must allow for such conditions. This paper aims to reduce computational costs regarding storage capacity and processing times by linearizing sensor outputs to be used in low-budget embedded systems. Results from two different sensors show that the esteemed exponential curve can be adjusted using neural networks with radial basis functions, with a reduction of over 50% in processing time. Regarding the design of the network, the paper focused on its efficiency in terms of the size of the network and the distribution of centroids, yielding important results. The achieved reduction in time and space allows for the use of the system's resources in other tasks, while it also allows for an increase of the sampling frequency in data acquisition.

Keywords

Sensor linearization, distance optical sensors, neural networks with radial basis functions.

Resumo

Os sensores de reflexão óticos para medir distância caracterizam-se pela não linearidade da saída. Essa condição deve ser assumida pelas unidades de processamento nas quais os sensores são utilizados. O presente artigo tem como objetivo reduzir o custo computacional, em termos de espaço de memória e tempo de processamento, na linearização da saída dos sensores que podem ser usados em sistemas embebidos de baixo custo. Os resultados de dois diferentes sensores indicam que a curva exponencial estimada pode ser ajustada usando-se redes neuronais com funções de base radial com uma redução de mais de 50% no tempo de processamento. Para projetar a rede, estudou-se a eficiência quanto ao tamanho da rede e da distribuição dos centroides com resultados importantes. A redução do tempo e do espaço atingida permite o aproveitamento dos recursos do sistema em outras tarefas e, ao mesmo tempo, possibilita aumentar a frequência da amostra na aquisição.

Palavras chave

Linearização de sensores, sensores óticos de distância, redes neurais com funções de base radial.

Introducción

Diferentes procesos industriales de los sectores manufacturero, automotor, aeronáutico, médico, entre otros, requieren desarrollar sistemas embebidos dentro de sus soluciones tecnológicas. Estos sistemas especializados contienen unidades centrales de procesamiento (CPU) específicas, de acuerdo con los requisitos del problema, que demandan algoritmos extensos que cumplen, pero ocupan completamente su memoria. Por lo tanto, la interacción con los procesos en el entorno debe llevarse a cabo de manera eficiente, haciendo la caracterización de medición una tarea bien importante.

Los sensores transforman cualquier tipo de señal a señales eléctricas que un sistema basado en microcomputadores emplea para monitorear y controlar. La salida de los sensores, muchas veces no lineal, puede ser acondicionada con *hardware* adicional como en (Khachab e Ismail, 1991; Volpi, Nizza y Bruschi, 2007); sin embargo, esto puede traer más dificultades. Una solución interna es usar tablas de búsqueda, pero la memoria requerida llega a ser considerable (Boris, Hocenski y Cvitas, 2006). Por otra parte, las no linealidades pueden tratarse mediante operaciones aritméticas, pues tienen un modelo preciso del sensor. En ausencia de modelos teóricos, las redes neuronales muestran una alternativa para la aproximación de funciones (Sánchez y Alanís, 2006). La linealización del sensor es considerada una tarea de estimación de funciones, donde la salida del sensor es el conjunto de datos y la respuesta linealizada es el objetivo de la estimación (Medrano-Marques y Martín-del-Brio, 2001).

En esta experiencia se escogieron sensores ópticos medidores de distancia. De acuerdo con el fabricante, se presenta una curva de naturaleza exponencial, que es no lineal. Algunos trabajos recientes todavía prefieren trabajar sobre el *hardware* como en (Mohan, Kumar y Sankaran, 2011; Nenova y Nenov, 2009) para linealizar. Otros sistemas, basados en microcontroladores, utilizan redes neuronales de tipo perceptrón multicapa (*multilayer perceptron* [MLP]) con circuitos integrados de bajo costo (8 bits) en diferentes sensores (Cotton y Wilamowski, 2010; Medrano-Marques *et al.*, 2005), con el inconveniente que hacen uso del

punto flotante. Para evitar la carga excesiva al microcontrolador, ocasionada por la función exponencial, se emplean redes neuronales con funciones de base radial (RBF) en la capa oculta, similar a (Chatterjee *et al.*, 2000).

La habilidad de una red neuronal para aproximar un mapeo entrada-salida puede verse de dos formas: identificación de sistemas y modelado inverso. En el caso de las RBF se espera que el espacio oculto sea linealmente separable, ya que la capacidad de la capa de salida es limitada y eficiente solo en esta condición. Esta conclusión está basada en el teorema de Cover, descrito en (Haykin, 1999).

Las RBF se han aplicado para aproximar funciones como en (Qu, Chen y Ji, 2010; Zhou y Lin, 2010), donde se realiza un esfuerzo por encontrar una técnica en la escogencia de los centroides. En (Erdem, 2010) se estudiaron seis técnicas basadas en *software* para linealizar un sensor GP2D12, de las estudiadas una de las más destacadas es la *Programmable Gain Amplifier* (PGA) aunque requiere el uso del dispositivos externos; sin embargo, la implementación con *Artificial Neural Network* (ANN) no tuvo buenos resultados.

El uso de las redes neuronales con RBF demostró reducir los requerimientos del sistema de procesamiento sin perder calidad en la estimación de la medida. Adicionalmente, la implementación con RBF permite al sistema la realización de otras tareas. En la primera parte del documento se presenta el desarrollo experimental y el sustento matemático del algoritmo. En la segunda parte se presentan los resultados más importantes y las conclusiones generales del trabajo desarrollado.

1. Descripción del trabajo

1.1. Materiales

Los sensores GP2Y0A21YK0F y GP2Y0A02YK son sensores de distancia de propósito general, desarrollado por la empresa SHARP. El sensor es una combinación de un *Position Sensitive Detector* (PSD) y de un *Infrared Emitting Diode* (IRED), combinado con un circuito procesador. El principio de funcionamiento reflexivo (por triangulación) reduce la influencia de variables ambientales sobre la medida. En la figura 1 se muestran los sensores implementados. En la tabla 1 se presentan las características técnicas más relevantes.

En la figura 2 se presenta la curva estimada por los fabricantes en condiciones ambientales controladas. Se resalta que la zona de operación que se puede estimar exponencial es confiable después de 10 cm para el GP2Y0A21YK, y de 20 cm para el GP2Y0A02YK, y se comprobó experimentalmente que funciona hasta 80 y 150 cm, respectivamente, reflejando el haz sobre papel blanco. De igual

forma, no existe una influencia relevante del material utilizado sobre la medida, aunque sí del color.

Figura 1. Sensores: a) GP2Y0A21YK y b) GP2Y0A02YK



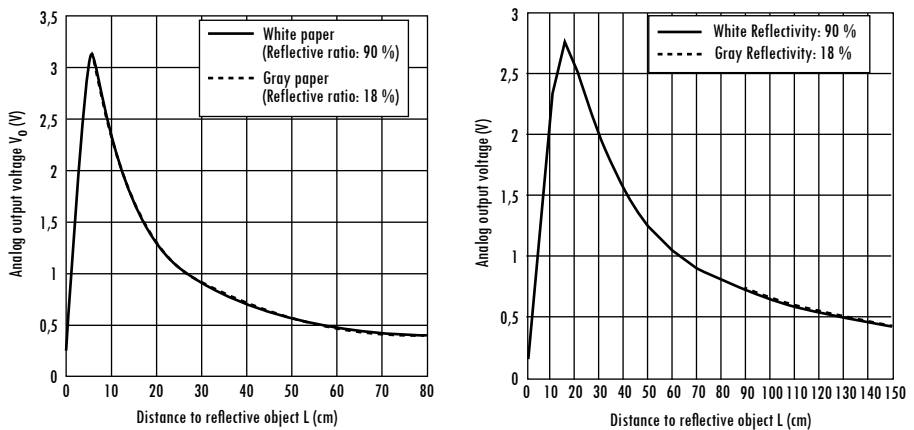
Fuente: a) (SHARP, 2005); b) (SHARP, 2006).

Tabla 1. Información técnica de los sensores

Referencia	GP2Y0A21YK	GP2Y0A02YK
Consumo de corriente	30 mA	33 mA
Voltaje de operación	4,5-5,5 V	4,5-5,5 V
Interfaz	Análoga	Análoga
Distancia sensada	10-80 cm	20-150 cm

Fuente: (SHARP, 2005 y 2006).

Figura 2. Característica del sensor: a) GP2Y0A21YK y b) GP2Y0A02YK



Fuente: a) (SHARP, 2005); b) (SHARP, 2006).

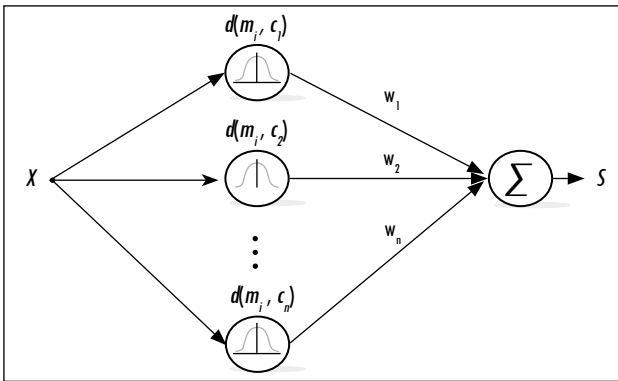
1.2. *Arquitectura de las redes RBF*

Las redes neuronales de tipo RBF están comúnmente constituidas por dos capas. Una capa oculta, formada por funciones del tipo radial, y una capa de salida, que realiza una suma ponderada a partir de las salidas de la capa oculta. La salida de la capa oculta tiene la forma mostrada en la ecuación (1).

$$D = [d(m_i, c_1), d(m_i, c_2), d(m_i, c_3), \dots, d(m_i, c_n)] \tag{1}$$

Donde m_i es el conjunto de variables de entrada, y c_1, c_2, \dots, c_n son los centroides estimados en la fase de entrenamiento, que son equidistantes en el rango de operación del sensor, lineal sobre la variable independiente o sobre la variable dependiente con el método presentado en la sección “estimación de centroides”. En la ecuación (1), $d(m_i, c_j)$ denota una función de distancia radial entre la muestra m_i y el centroide c_j . Nótese que D , el vector resultante de la capa oculta, es de dimensión $[1 \times n]$, donde n es el número de centroides como se aprecia en la figura 3.

Figura 3. Arquitectura de la RBF



Fuente: presentación propia de los autores.

La capa de salida se puede expresar como se muestra en la ecuación (2).

$$S = D \cdot W = [d(m_i, c_1), d(m_i, c_2), d(m_i, c_3), \dots, d(m_i, c_n)] \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} \tag{2}$$

Donde D es el vector de distancias obtenido de la capa oculta, W es el vector de pesos de la capa de salida obtenido en el algoritmo de entrenamiento y S es la única salida de la red $[1 \times 1]$.

1.3. Pesos de la capa de salida

Para realizar el entrenamiento de la capa de salida en la red RBF es necesario encontrar la solución que mejor se ajuste al sistema mostrado en la ecuación (3):

$$Gx = b \tag{3}$$

Donde:

$$G = \begin{bmatrix} d(m_1, c_1) & d(m_1, c_2) & d(m_1, c_3) & \cdots & d(m_1, c_n) \\ d(m_2, c_1) & d(m_2, c_2) & d(m_2, c_3) & \cdots & d(m_2, c_n) \\ d(m_3, c_1) & d(m_3, c_2) & d(m_3, c_3) & \cdots & d(m_3, c_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d(m_m, c_1) & d(m_m, c_2) & d(m_m, c_3) & \cdots & d(m_m, c_n) \end{bmatrix} \tag{4}$$

Las filas de G en la ecuación (4) están construidas con la salida de la capa oculta, donde el número de filas es definido por el número de muestras para realizar el entrenamiento. El número de centroides n se estima iterando para obtener un valor de error requerido por la aplicación.

El vector de salida b deseado se construye a partir de los vectores calculados con la función que se desea estimar como se muestra en la ecuación (5).

$$F(m_i) = a_i$$

$$b = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n-1} \\ a_n \end{bmatrix} \tag{5}$$

Para resolver este sistema debemos usar la pseudoinversa. Premultiplicando la traspuesta de la ecuación (4) en la ecuación (3), se tiene la ecuación (6).

$$G^T Gx = G^T b \tag{6}$$

Despejando x , se obtiene la proyección del vector w en el espacio columna de G . Se muestra en la ecuación (7). Ya que la matriz $G^T G$ es simétrica, el sistema es invertible, luego siempre tiene solución.

$$\tilde{x} = p = (G^T G)^{-1} G^T b \tag{7}$$

1.4. Estimación de centroides

Uno de los problemas comunes al utilizar métodos basados en agrupamiento, como son las RBF, es el de la escogencia del número ideal de centroides. Para muchas técnicas usadas, esta escogencia se realiza de forma heurística, pues reduce la flexibilidad y la adaptabilidad del algoritmo.

En este trabajo se presenta un método para determinar un número adecuado de núcleos. Inicialmente se consideró escoger los centroides separados uniformemente en el rango de operación de la señal de entrada. La separación está ligada a la concentración de valores de la función alrededor del centroide, es decir, la selección ideal en la separación implica valores de centroides cercanos cuando los cambios en la curva son fuertes, y lejanos cuando los cambios son suaves. Esta transformación beneficia el comportamiento de la capa de salida (que es una operación lineal) y, por consiguiente, el rendimiento de la estimación.

En primer lugar se implementó una separación lineal de los centroides (método centroides 1) sobre el eje de la variable independiente usando la ecuación (8).

$$c_i = (i - 0,5) \left(\frac{\max(X) - \min(X)}{n} \right) + \min(X) \quad (8)$$

$$i = 1, 2, 3, \dots, n$$

Teniendo en cuenta la percepción de que la ubicación de los centroides debería ser proporcional al cambio de la función, se optó por desarrollar una técnica que separara uniformemente sobre el eje de la variable dependiente y proyectara los centroides sobre el eje de la variable independiente usando la función objetivo. Para esto, inicialmente, se encuentran los valores que separan de manera equidistante la variable sobre el eje de la variable dependiente, como se muestra en la ecuación (9).

$$de_i = (i - 0,5) \left(\frac{\max(Y) - \min(Y)}{n} \right) + \min(Y) \quad (9)$$

$$i = 1, 2, 3, \dots, n$$

Donde de es el vector de distancias equidistantes sobre la variable dependiente Y , n es el número de centroides y Y es el vector de la función que se va a estimar disponible para entrenamiento. Luego se ubica la posición correspondiente utilizando la ecuación (10); p_{0_i} es la posición del vector Y donde se encuentra el mínimo indicado en la operación.

$$p_{o_i} = \min (abs(Y - de_i)) \tag{10}$$

$$i = 1,2,3, \dots n$$

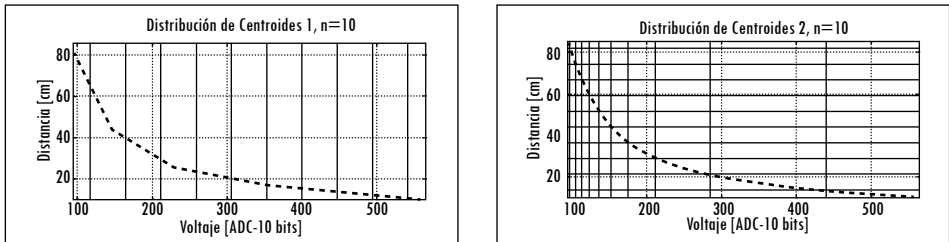
Los centroides se obtienen tomando los valores del vector de la variable independiente disponibles para entrenamiento (método centroides 2), como se muestra en la ecuación (11).

$$c_i = X(p_{o_i})$$

$$i = 1,2,3, \dots n \tag{11}$$

En la figura 4a las líneas verticales muestran la proyección de los centroides sobre la variable independiente, los centroides se separan uniformemente (centroides 1). En la figura 4b se utiliza el método descrito en las ecuaciones (9), (10) y (11), separación no uniforme (centroides 2). Se observa que la separación en el eje de la variable dependiente es uniforme y que la proyección que se realiza sobre el eje de la variable independiente distribuye los centroides en función de la tasa de cambio de la función que se va a estimar.

Figura 4. Distribución para $n = 10$: a) centroides 1; b) centroides 2



Fuente: presentación propia de los autores.

2. Métodos

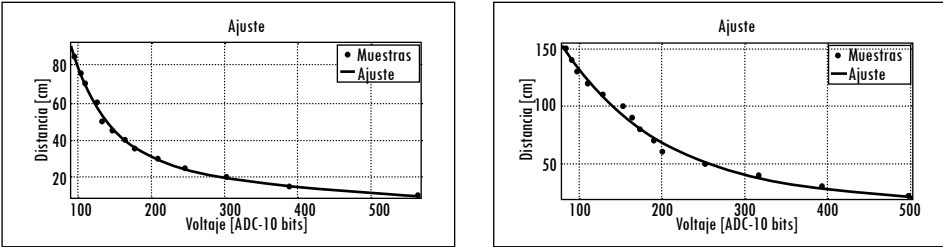
Se seleccionaron dos sensores: GP2Y0A21YK0F y GP2Y0A02YK, como los presentados en la figura 1, para realizar las medidas de distancia. Se tomaron diez puntos de referencia, y los datos medidos se promediaron para obtener un vector de distancia y un vector de voltajes que permitiera hacer un ajuste matemático de las curvas características de medición de distancia.

Teniendo en cuenta las curvas suministradas por el fabricante, vistas en la figura 2, y el error de estimación obtenido con diferentes funciones de aproximación, se decidió emplear una función exponencial doble, como se muestra en la figura 5. La función que se va a aproximar y los parámetros que se van a estimar para los dos sensores se muestran en la ecuación (12):

$$d = a_1 e^{b_1 V} + a_2 e^{b_2 V} \tag{12}$$

Donde V es el voltaje de salida expresado en proporción al valor entero en un conversor análogo digital de 10 bits, d es la distancia correspondiente en cm y a_1, a_2, b_1 y b_2 son los parámetros de ajuste que se van a estimar. Los valores obtenidos se muestran en la tabla 2.

Figura 5. Valores de ajuste: a) GP2Y0A21YK; b) GP2Y0A02YK



Fuente: presentación propia de los autores.

Tabla 2. Coeficientes estimados

	a_1	b_1	a_2	b_2
GP2Y0A21YK0F	381,5	-0,02111	42,40	-0,002594
GP2Y0A02YK	251,8	-0,008107	20,36	-0,0003681

Fuente: presentación propia de los autores.

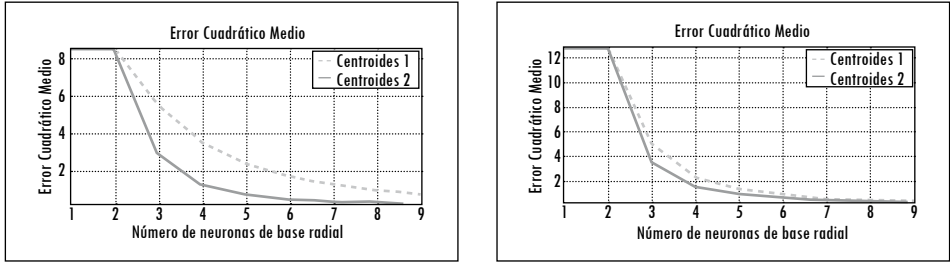
3. Resultados

3.1. Medidas de error

Se implementó la red RBF variando el número de neuronas en la capa oculta entre dos y nueve neuronas para ambos sensores. Se detuvo el crecimiento de la red por la reducción del error de estimación. En la figura 6 se presenta el error cuadrático medio respecto al crecimiento de la red y a las dos posibilidades de centroides presentadas en la subsección 1.4.

El error decae más rápidamente si la separación entre los centroides no es uniforme (centroides 2); de igual forma, la separación lineal no alcanza la misma reducción en error de la implementación. Ello implica la necesidad de usar un mayor número de centroides con la técnica lineal.

Figura 6. Error cuadrático medio: a) GP2Y0A21YK; b) RMS GP2Y0A02YK

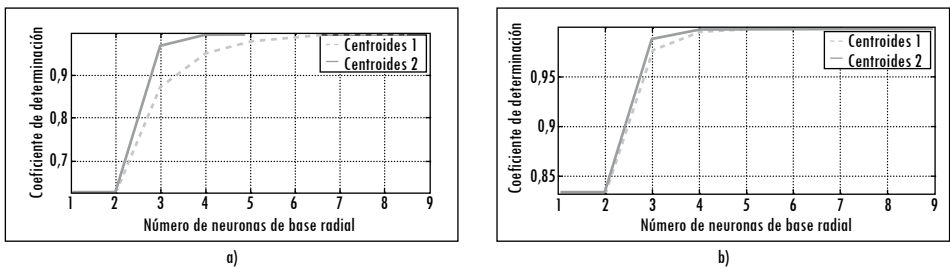


Fuente: presentación propia de los autores.

Es necesario aclarar que la escogencia de los centroides se realiza fuera de línea y, por consiguiente, no afecta el tiempo de procesamiento. El modelo presentado contempla que el entrenamiento fuera de línea evite adicionar carga de procesamiento del sensor al microcontrolador. Con las muestras adquiridas para calibración se lleva a cabo el entrenamiento, la estimación de centroides (tamaño y ubicación) y el cálculo de los pesos de salida. Esta tarea se realiza una sola vez para el sensor y los datos son posteriormente descargados como constantes al microcontrolador encargado de la tarea de muestreo.

En la figura 7 se presenta el coeficiente de determinación para la implementación con ambos métodos de selección de centroides y para la red RBF con 2, 3, 4, 5, 6, 7, 8 y 9.

Figura 7. Coeficiente de determinación: a) GP2Y0A21YK; b) GP2Y0A02YK



Fuente: presentación propia de los autores.

3.2. Medidas de tiempo

Como se mencionó, la red RBF se entrena fuera de línea. Se usó un computador con un procesador Intel® Core™ E7300 @ 2,66 GHz a 64 bits con memoria RAM de 8 GB. La carga de la fase de entrenamiento no es significativa. Poste-

riormente se descargó en dos microcontroladores Microchip®, usando rutinas de programación C++. Las características de los microcontroladores se presentan en la tabla 3.

Tabla 3. Especificaciones de los microcontroladores

	Memoria de programa (KB)	Memoria RAM (B)	Puertos análogos (cantidad)	No de pines (cantidad)
PIC18F2550	32	2048	10	28
PIC16F88	7	368	7	18

Fuente: hojas de características.

Finalmente, para comparar el rendimiento de la implementación directa, es decir, realizar la operación descrita en la ecuación (12) o implementar la operación en la ecuación (2), que permite obtener el valor de salida con la red RBF, se calculó el tiempo y se obtuvo el porcentaje de memoria RAM y ROM usado por ambos métodos. Los resultados se muestran en la tabla 4. ALG RBF corresponde a la implementación que se ha presentado en este documento (el rendimiento computacional en el microcontrolador es indiferente a la escogencia de los centroides) y ALG DIR corresponde a la implementación directa, es decir, a ingresar el valor de la variable independiente directamente a la ecuación (12).

Para los dos microcontroladores evaluados el tiempo de procesamiento se reduce más del 50 % con la implementación RBF y el porcentaje de memoria ROM también es mucho menor. Los casos marcados como NA en la tabla implican que no soporta el algoritmo.

Tabla 4. Uso de memoria y tiempo de procesamiento

	18F2550 INT 32	18F2550 FLOAT	16F88 INT 32	16F88 FLOAT	
ALG RBF	2	4	8	18	ROM [%]
	2	2	17	17	RAM [%]
	0,0031	0,0051	0,0067	0,0091	Tiempo [s]
ALG DIR	NA	9	NA	35	ROM [%]
	NA	1	NA	16	RAM [%]
	NA	0,0119	NA	0,0218	Tiempo [s]

Fuente: presentación propia de los autores.

Conclusiones

Es fundamental la adquisición en sistemas automáticos de control; por esto se deben buscar técnicas que reduzcan la capacidad de procesamiento necesaria para procesar las señales de los sensores, como las redes neuronales con funciones de base radial. La reducción del tiempo y del espacio lograda con la red RBF, comparada con la implementación directa, permite aprovechar los recursos del sistema en otras tareas, al mismo tiempo que posibilita aumentar la frecuencia de muestreo en la adquisición. El marco experimental se diseñó sobre microcontroladores de bajo costo; por lo tanto, es indispensable entrenar la red neuronal previa al funcionamiento del sensor. Para microcontroladores más robustos se puede pensar en extrapolar el algoritmo de tal manera que se llegue a realizar en línea.

Para los sensores ópticos estudiados, la curva característica se puede ajustar usando redes neuronales con RBF con nueve neuronas en la capa oculta y con los centroides distanciados de manera no uniforme. Ello consigue una reducción de más del 50 % en tiempo de procesamiento y memoria de programa.

Referencias

- BORIS, B.; HOCENSKI, Z. y CVITAS, L. Optimal approximation parameters of temperature sensor transfer characteristic for implementation in low cost microcontroller systems. *IEEE International Symposium on Industrial Electronics*. 2006, núm. 4, pp. 2784-2787. DOI: 10.1109/ISIE.2006.296055.
- CHATTERJEE, A.; MUNSHI, S.; DUTTA, M. y RAKSHIT, A. An artificial neural linearizer for capacitive humidity sensor. *Proceedings Of The 17th IEEE Instrumentation And Measurement Technology Conference* [Cat. No. 00CH37066]. 2000, núm. 1, pp. 313-317. DOI: 10.1109/IMTC.2000.846876.
- COTTON, N. J. y WILAMOWSKI, B. M. Compensation of sensors nonlinearity with neural networks. *24th IEEE International Conference on Advanced Information Networking and Applications*. 2010, pp. 1210-1217. DOI: 10.1109/AINA.2010.170.
- ERDEM, H. Implementation of software-based sensor linearization algorithms on low-cost microcontrollers. *ISA Transactions*. 2010, vol. 49, núm. 4, pp. 552-558. DOI: 10.1016/J. Isatra.2010.04.004.
- HAYKIN, S. *Neural networks and learning machines*. 3rd ed. New York: Pearson, 2009.
- KHACHAB, N. I. e ISMAIL, M. (1991). Linearization techniques for nth-order sensor models in MOS VLSI technology. *IEEE Transactions on Circuits and Systems*. 1991, vol. 38, núm. 12, pp. 1439-1450. DOI: 10.1109/31.108498.
- MEDRANO-MARQUES, N. J. y MARTÍN-DEL-BRIO, B. Sensor linearization with neural networks. *IEEE Transactions on Industrial Electronics*. 2001, vol. 48, núm. 6, pp. 1288-1290. DOI: 10.1109/41.969414.

- MEDRANO-MARQUES, N. J.; MARTÍN-DEL-BRIO, B.; BONO-NUEZ, A. y BERNAL-RUIZ, C. Implementing neural networks onto standard low-cost microcontrollers for sensor signal processing. *IEEE Conference on Emerging Technologies and Factory Automation*. 2005, vol. 2, pp. 967-972. DOI: 10.1109/ETFA.2005.1612776.
- MOHAN, N. M.; KUMAR, V. J. y SANKARAN, P. Linearizing dual-slope digital converter suitable for a thermistor. *IEEE Transactions on Instrumentation and Measurement*. 2011, vol. 60, núm. 5, pp. 1515-1521.
- NENOVA, Z. P y NENOV, T. G. Linearization circuit of the thermistor connection. *IEEE Transactions on Instrumentation and Measurement*. 2009, vol. 58, núm. 2, pp. 441-449. DOI: 10.1109/TIM.2008.2003320.
- QU, L.; CHEN, Y. y JI, Y. Regularized RBF-FA neural network to improve the generalization performance of function approximation: Computer application and system modeling (ICCSM) [artículo en línea]. *IEEE International Conference on*. 2010, vol. 11, pp. V11-267. <http://Ieeexplore.Iee.org/Xpls/Abs_All.Jsp?Arnumber=5623211>.
- SHARP. *Optoelectronic device GP2Y0A21YK* [Datasheet]. 2005. <http://Www.Sharpsma.Com/Webfm_Send/1208>.
- SHARP. *Optoelectronic device GP2Y0A02YK* [Datasheet]. 2006. <<http://Www.Datasheetarchive.Com/GP2Y0A02YK-Datasheet.Html>>.
- SÁNCHEZ, E. N. y ALANÍS, A. Y. *Redes neuronales: conceptos fundamentales y aplicaciones a control automático*. Madrid: Pearson; 2006.
- VOLPI, E.; NIZZA, N. y BRUSCHI, P. *A non linear ADC for sensor linearization*. PhD Research in Microelectronics and Electronics Conference. 2007, pp. 5-8. DOI: 10.1109/RME.2007.4401797.
- ZHOU, S. y LIN, H. Function approximation based on self-adaptive RBF neural network with combined clustering algorithm [document en línea]. *International Conference on Intelligent Control and Information Processing (ICICIP)*. 2010, pp. 435-438. <http://Ieeexplore.Iee.org/Xpls/Abs_All.Jsp?Arnumber=5565283>.