

# Implementación de la técnica de Escalado Métrico Multidimensional en FPGAs para posicionamiento relativo en espacios interiores

Santiago Murano, Alfonso González, Fernando Pino,  
Gustavo Guazzone, Rómulo Alcoleas, José Gallardo,  
Jorge Pires

Departamento de Electrónica, Facultad de Ingeniería  
Universidad Nacional de la Patagonia San Juan Bosco  
Comodoro Rivadavia, Argentina  
smurano@unpata.edu.ar

Carlos De Marziani

Departamento de Electrónica, Facultad de Ingeniería  
Universidad Nacional de la Patagonia San Juan Bosco  
Consejo Nacional de Investigaciones Científicas y Técnicas  
Comodoro Rivadavia, Argentina  
marziani@unpata.edu.ar

**Abstract**— La estimación de las posiciones entre nodos a partir de la información recolectada entre pares de éstos se denomina Localización Relativa. La complejidad de los algoritmos es mayor debido a que las posiciones son estimadas utilizando medidas entre nodos que no tienen conocimiento de su ubicación en el entorno. Uno de los algoritmos que frecuentemente se utiliza en sistemas de localización relativa es la técnica de Escalado Métrico Multidimensional (MDS métrico o clásico). Este permite obtener una configuración geométrica de la posición entre nodos, en el mínimo número de dimensiones posibles, empleando las distancias medidas entre ellos. Generalmente es posible obtener una solución analítica aun en aquellos casos donde las distancias están contaminadas con diferentes fuentes de errores que perturban la recolección de datos. Para el empleo de la técnica MDS es necesario realizar operaciones aritméticas sobre el conjunto de datos a analizar organizados de forma matricial, denominado operación de doble centrado. Posteriormente se debe realizar la descomposición en valores singulares de la matriz de datos resultante a fin de obtener las posiciones de los objetos. Así, en este trabajo se analiza la implementación de este algoritmo sobre dispositivos de lógica programable a fin de evaluar la viabilidad de obtener un funcionamiento en tiempo real del sistema bajo estudio utilizando los recursos hardware disponibles y, adicionalmente, implementando dicha técnica en un microprocesador embebido MicroBlaze sobre éste dispositivo.

**Keywords**— Escalado Métrico Multidimensional, Dispositivos de lógica programable, Sistemas de Posicionamiento Relativo

## I. INTRODUCCIÓN

La estimación de las posiciones entre nodos a partir de la información recolectada entre pares de éstos suele llamarse Localización Relativa [1]. Por ello, la complejidad de los algoritmos es mayor debido a que las posiciones son estimadas utilizando medidas entre nodos que, habitualmente, no tienen conocimiento de su ubicación en el entorno. Diversas estrategias de posicionamiento pueden emplearse para determinar la posición relativa de los objetos, pero su empleo depende principalmente de la precisión deseada en la estimación, del número de observaciones y de la capacidad de

cómputo disponible en cada nodo, entre otros factores. En espacios interiores, se utilizan, generalmente, técnicas de multilateración donde los sistemas de ecuaciones a resolver consideran todas las medidas entre los nodos. Uno de los algoritmos que frecuentemente se utiliza en sistemas de localización relativa es la técnica de Escalado Métrico Multidimensional (MDS) [2-3]. Dicho algoritmo permite obtener una configuración geométrica de la posición entre nodos, en el mínimo número de dimensiones posibles, empleando las distancias medidas entre ellos. Generalmente es posible obtener una solución analítica del sistema aun en aquellos casos donde las distancias están contaminadas con ruido o diferentes fuentes de errores que perturban la recolección de datos, tales como: respuesta de los transductores, retardos en los algoritmos de procesamiento, efectos por múltiples caminos, etc. Para el empleo de la técnica MDS es necesario realizar operaciones aritméticas sobre el conjunto de datos a analizar organizados de forma matricial, denominado operación de doble centrado [4]. Posteriormente se debe realizar la descomposición en valores singulares de la matriz de datos resultante a fin de obtener las posiciones de los objetos.

En el sistema propuesto en [5-6], el objetivo final es computar en cada nodo, las posiciones con el resto de objetos del sistema empleando como observaciones los pseudotiempo de vuelo (pTOF) calculados entre las emisiones realizadas por los diferentes nodos según el mecanismo SRTOF (*Simultaneous Round-Trip-Time-of-Flight*). De esta manera, a partir de las relaciones temporales computadas, es posible obtener las distancias entre todos los nodos y posteriormente utilizar esta información para determinar las posiciones entre los objetos empleando la técnica MDS [5]. Así, en este trabajo se analiza la viabilidad de implementación del algoritmo MDS sobre dispositivos de lógica programable a fin de computar las posiciones entre los nodos y obtener un funcionamiento en tiempo real del sistema bajo estudio utilizando los recursos hardware disponibles o bien implementando dicha técnica en un microprocesador embebido MicroBlaze®.

El trabajo se encuentra organizado de la siguiente manera, en la siguiente sección se presenta el sistema de medida utilizado para determinar las distancias entre objetos. En la sección III se presenta la técnica de escalado métrico multidimensional. La sección IV describe las propuestas de implementación sobre FPGAs. Finalmente se describe las pruebas experimentales y las conclusiones del trabajo.

## II. MECANISMO DE MEDIDA SRTOF

De acuerdo al sistema descrito en [6], es posible obtener dos tipos de observaciones sobre las relaciones espaciales entre nodos utilizando el mecanismo de medida SRTOF (*Simultaneous Round Trip Time of Flight*). Por un lado, entre un nodo denominando *Master* y el resto de nodos del sistema que se denominan nodos esclavos y, por el otro, entre los nodos esclavos entre sí. A fin de resumir las relaciones temporales se considera que el nodo *Master* tiene una posición en el entorno con coordenadas  $(x, y, z)$ , que se describe a través de un vector  $\mathbf{p}_{Master}$ ; y que las coordenadas de dos nodos esclavos cualesquiera vienen dadas por los vectores  $\mathbf{p}_q$  y  $\mathbf{p}_l$  respectivamente (véase Fig. 1).

En (1) se describe la relación temporal existente entre las emisiones del nodo *Master* con cada uno de los nodos esclavos para el cómputo de los pTOF. Además de los tiempos de propagación de las emisiones realizadas por cada uno de los nodos y el tiempo de código  $T_{CODE}$ , es necesario considerar el tiempo requerido en las etapas de emisión y recepción para el procesamiento de las señales en el nodo *Master* y el nodo esclavo  $q$  con el cual se establece esta relación temporal:

$$\hat{t}_{Mtr-q} = 2 \cdot \frac{\|\mathbf{p}_{Mtr} - \mathbf{p}_q\|}{c} + tp_{Mtr} + tp_q + T_{CODE} \quad (1)$$

Donde  $\hat{t}_{Mtr-q}$  es el pTOF determinado en el nodo *Master* con el nodo esclavo  $q$ ; y  $T_{CODE}$  corresponde a la duración del código emitido. Además,  $tp_{Mtr}$  y  $tp_q$  representa el tiempo requerido para el procesamiento de las señales que emiten y reciben de cada nodo en el *Master* y el nodo esclavo  $q$ .

Por otra parte, el tiempo medido en las relaciones temporales entre las emisiones de los nodos esclavos viene dado por:

$$\hat{t}_{q-l} = \frac{\|\mathbf{p}_{Mtr} - \mathbf{p}_l\|}{c} + \frac{\|\mathbf{p}_l - \mathbf{p}_q\|}{c} - \frac{\|\mathbf{p}_{Mtr} - \mathbf{p}_q\|}{c} + tp_l + T_{CODE} \quad (2)$$

Donde  $\hat{t}_{q-l}$  es el pTOF determinado en el nodo esclavo  $q$  con el nodo esclavo  $l$  y  $tp_l$  es el tiempo de procesamiento de señales asociado al nodo esclavo  $l$ .

En (2), además de los tiempos de propagación de las emisiones de cada nodo, existe un retardo que corresponde únicamente al proceso de emisión y recepción asociado al nodo esclavo  $l$  con el cual se trata de establecer la relación.

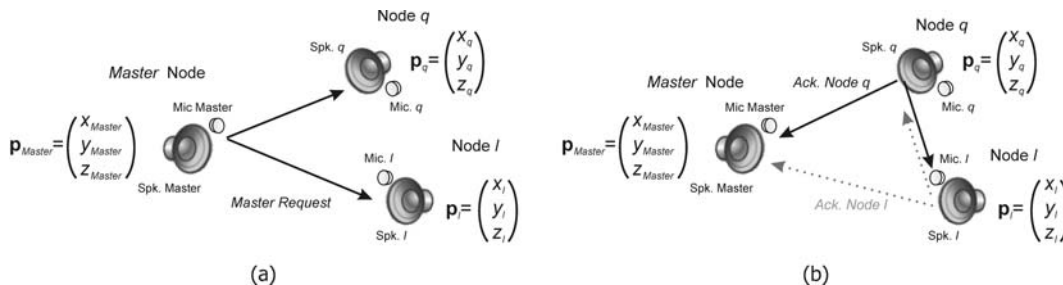


Fig. 1 a) Emisión del Nodo Master. b) Respuesta de los nodos esclavos.

A fin de determinar las distancias entre nodos a partir de los pTOF, se considerará que los tiempos de procesamiento asociados a cada nodo son conocidos (a través de su caracterización *off-line*) y que se suponen invariantes en el tiempo. De esta manera, estos parámetros participan en las relaciones temporales como una constante. Con las relaciones temporales que recolecta el nodo *Master*, asumiendo que la velocidad de desplazamiento de los nodos es mucho menor que la velocidad de propagación de las ondas acústicas, es posible establecer las siguientes distancias entre el nodo *Master* y los esclavos.

$$d_{Mtr-q} = \|\mathbf{p}_{Mtr} - \mathbf{p}_q\| = \frac{[\hat{t}_{Mtr-q} - (tp_{Mtr} + tp_q + T_{CODE})]}{2} \cdot c \quad (3)$$

Por otra parte, con la información recolectada por cada nodo esclavo no es posible establecer la distancia entre ellos. Por lo tanto, utilizando los pTOF recíprocos entre los nodos esclavos  $q$  y  $l$  se obtiene la siguiente relación:

$$\hat{t}_{q-l} + \hat{t}_{l-q} = \frac{\|\mathbf{p}_l - \mathbf{p}_q\|}{c} + \frac{\|\mathbf{p}_q - \mathbf{p}_l\|}{c} + tp_l + tp_q + 2 \cdot T_{CODE} \quad (4)$$

$$[\hat{t}_{q-l} + \hat{t}_{l-q} - (tp_l + tp_q + 2 \cdot T_{CODE})] \cdot c = \|\mathbf{p}_l - \mathbf{p}_q\| + \|\mathbf{p}_q - \mathbf{p}_l\| \quad (5)$$

Asumiendo nuevamente que la velocidad de desplazamiento de los nodos es mucho menor que la velocidad de propagación de las ondas acústicas se obtiene entonces que:

$$d_{ql} = \|\mathbf{p}_l - \mathbf{p}_q\| = \frac{c}{2} \cdot [\hat{t}_{q-l} + \hat{t}_{l-q} - tp_l - tp_q - 2 \cdot T_{CODE}] \quad (6)$$

De esta forma, con (3) y (6) se pueden determinar las distancias entre los nodos del sistema. Posteriormente estos resultados son empleados en cada nodo para la ejecución del algoritmo MDS a fin de realizar el cómputo de la posiciones entre ellos.

## III. ESCALADO MÉTRICO MULTIDIMENSIONAL

La técnica MDS permite obtener una configuración geométrica de los objetos a posicionar, en el menor número de dimensiones posibles, cuando todo lo que se conoce es una medida de la relación existente entre ellos, en este caso la distancia. El procedimiento a seguir se muestra en la Fig. 2 y los pasos pueden resumirse como se detalla a continuación:

Como primer paso, en cada nodo donde se requiera realizar el cómputo de las posiciones utilizando el algoritmo MDS, se deben conocer todas las distancias entre los objetos, lo cual se expresa en una matriz de distancias  $\mathbf{D}_{Q \times Q}$  (Paso 1 en la Fig. 2).

El siguiente paso es la construcción de una matriz  $\mathbf{B}_{Q \times Q}$ , denominada matriz producto de puntos, que considera las distancias entre cada uno de los objetos con respecto a un punto de referencia. Cualquier punto del sistema puede ser seleccionado como referencia, sin embargo se recomienda utilizar el centroide de la figura que forman los objetos.

Esto se debe a que, si las distancias están contaminadas con ruido aleatorio, al elegir este punto se minimizan los errores ya que éstos tienden a cancelarse. Así, a cada elemento de la matriz de distancias  $\mathbf{D}$  se le aplica la siguiente relación:

$$b_{iq} = -\frac{1}{2} \left[ d_{iq}^2 - \frac{1}{Q} \sum_{y=1}^Q d_{iy}^2 - \frac{1}{Q} \sum_{x=1}^Q d_{xq}^2 + \frac{1}{Q^2} \sum_{m=1}^Q \sum_{n=1}^Q d_{mn}^2 \right] \quad (7)$$

Donde  $d_{iq}$  es la distancia entre dos nodos genéricos  $q$  y  $l$  del sistema,  $q, l, x, y, m, n \in \{1, 2, \dots, Q\}$  y  $b_{iq}$  es el producto de puntos asociados a los nodos  $q$  y  $l$  con el centroide de la figura que forman los objetos.

La operación descrita se denomina doble centrado, es decir, a cada elemento distancia se le resta la distancia media de la fila y la distancia media de la columna en la cual se encuentra y se suma la media cuadrática de todas las distancias.

Una vez obtenida la matriz  $\mathbf{B}$ , se procede a su descomposición en valores singulares (SVD, *Singular Value Decomposition*). Considerando las propiedades de las matrices de autovectores  $\mathbf{U}$  y autovalores  $\mathbf{S}$  asociadas a  $\mathbf{B}$ , se determina una matriz de coordenadas de los nodos mediante la siguiente relación:

$$\mathbf{X}^* = (\mathbf{U} \cdot \mathbf{S}^{1/2}) \quad 8$$

Para obtener la matriz de coordenadas  $\mathbf{X}^*$  se eligen las primeras dos o tres columnas, según sea en 2D ó 3D.

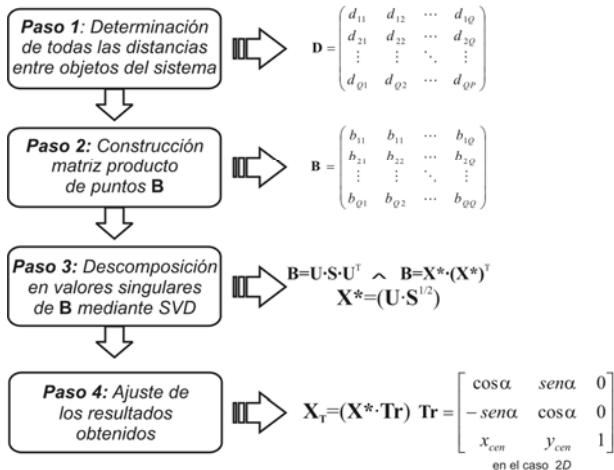


Fig. 2 Esquema del algoritmo de Escalado Métrico Multidimensional para obtener las coordenadas del sistema.

Finalmente, es necesario adaptar los resultados de la estimación como consecuencia de que los resultados obtenidos en la matriz  $\mathbf{X}^*$  utilizan como punto de referencia al centroide de la figura formada por los objetos. A fin de realizar la transformación de las posiciones estimadas al sistema de referencia de cada nodo, se debe contemplar un proceso de rotación y traslación mediante una matriz de transformación  $\mathbf{Tr}$ . La nueva matriz de coordenadas  $\mathbf{X}^T$  resultante de esta transformación contiene las coordenadas de cada nodo ajustadas al sistema de referencia preestablecido para cada uno de ellos.

#### IV. IMPLEMENTACIÓN HARDWARE

Se ha evaluado la viabilidad de implementación de los algoritmos analizados en la sección anterior a partir de una placa de desarrollo Nexys 2 de Digilent Inc. que dispone de una FPGA Xilinx XC3S1200E [7].

De acuerdo a los pasos detallados en la Fig. 2 para el cálculo de la matriz  $\mathbf{B}$ , se ha considerado que ya se dispone de la información de las distancias almacenadas en un bloque de memoria digitalizada en 8 bits. Cada elemento de la matriz  $\mathbf{B}$  de  $Q \times Q$  se obtiene de (7). Se han evaluado dos alternativas de cálculo, por un lado la generación de  $\mathbf{B}$  empleando los recursos hardware disponible en una FPGA programándolos en VHDL; y por otro, la construcción de  $\mathbf{B}$  a partir de la implementación de un microprocesador Microblaze® embebido en la FPGA y su programación en C. Posteriormente la descomposición en valores singulares SVD se implementa sobre el Microblaze® [8] utilizando algoritmos programados en C o bien a través de la interfaz implementada en el ordenador.

1) *Generación de la matriz producto de puntos  $\mathbf{B}$  empleando los recursos hardware disponibles en la FPGA*

Para un sistema de  $Q$  nodos, se ha definido un bloque de  $Q^2$  registros para formar la matriz de distancias  $\mathbf{D}$  en los cuales se encuentran almacenadas las distancias  $d_{ql}$  con  $q, l \in \{1, 2, \dots, Q\}$ . Considerando que cada distancia se almacena en 8 bits, se ha implementado mediante  $Q^2$  registros de 16 bits la matriz  $\mathbf{D}^2$  (véase Fig. 3). Para obtener cada elemento, en este bloque se recorren cada uno de los registros de  $\mathbf{D}$  y se elevan cada uno de sus elementos al cuadrado para posteriormente almacenarlos en  $\mathbf{D}^2$ . Una vez que se obtienen todos los términos, se emite una señal de habilitación que inicia un nuevo proceso encargado de obtener cada uno de los elementos de la matriz  $\mathbf{B}$ .

Para obtener cada elemento  $b_{ij}$ , se utiliza un bloque sumador al que se envían los resultados de los distintos términos requeridos en: suma por filas de las distancias, suma por columnas y suma total de las distancias al cuadrado. Cada uno de estos términos se encuentra dividido por el número total de nodos  $Q$  y se divide la salida por 2 para obtener  $b_{ij}$ , por lo que a la salida se tendrá un nuevo arreglo matricial de  $Q \times Q$  elementos.

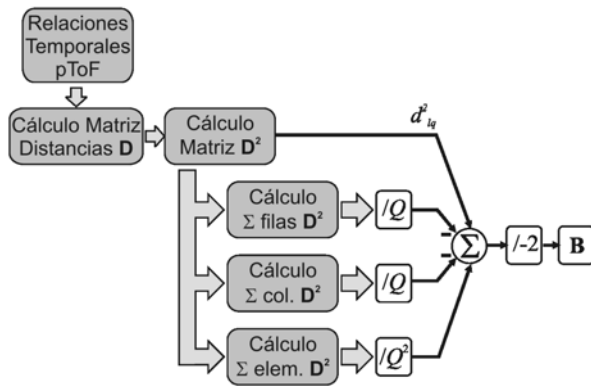


Fig. 3. Diagrama de bloques básico para el cálculo de la matriz producto de puntos **B**

### B. Generación de la matriz producto de puntos **B** mediante Microblaze embebido en FPGA.

En primer lugar se procedió a la implementación del microprocesador Microblaze® en la FPGA siendo programadas las tareas que realizará en lenguaje C. Para su implementación se ha utilizado el software EDK (*Embedded Development Kit*) que es un entorno de desarrollo integrado para el diseño de sistemas embebidos programables de Xilinx. Este kit de preconfiguración contiene la herramienta XPS (*Xilinx Platform Studio*) que posibilita la configuración de la arquitectura, buses y periféricos del sistema embebido. El entorno SDK (*Software Development Kit*) permite agregar la aplicación (C/C++) de Software para la cual el sistema fue diseñado. Se ha configurado un único procesador con una frecuencia de operación a través del oscilador externo de 50 MHz disponible en la placa de desarrollo; y una memoria RAM local de 32 kB. En este caso hay que tener especial cuidado de no exceder los recursos disponibles en la FPGA. Asimismo la memoria chaché de 2kB se ha implementado utilizando los LUTs internos de la FPGA.

En la Fig. 4, se muestra el diagrama de flujo para la generación de **B**. En este caso, se ingresan las coordenadas de los distintos nodos con una interfaz gráfica desarrollada en Matlab, y se procedió a calcular la matriz distancias **D** a partir de dichos datos. Posteriormente se ejecuta cada uno de los pasos de haciendo uso de los pulsadores que dispone la placa de desarrollo a fin de verificar el funcionamiento.

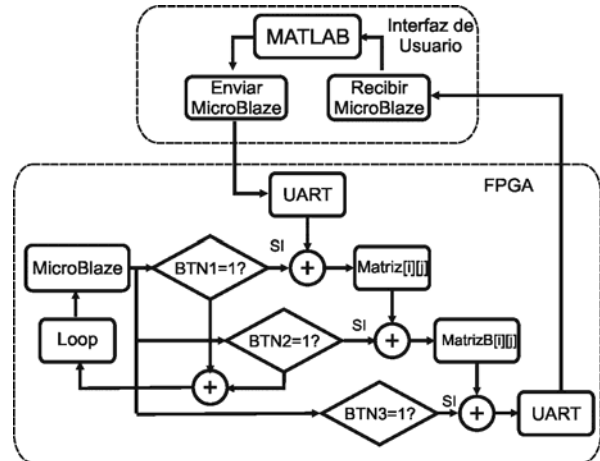


Fig. 4. Diagrama de flujo utilizado para el cálculo de **B** mediante Microblaze®

El microprocesador encuesta el estado de cada uno de los pulsadores denominados como BTN1, BTN 2 y BTN 3 que ejecutarán las siguientes tareas:

A partir de la activación de BTN1, el microprocesador comienza a recorrer un bucle en el cual se esperan los datos recibidos por el puerto serie de la FPGA. Luego los ordena en el mapa de memoria donde se encuentra mapeada la UART, al mismo tiempo que se almacenan estos datos en una variable denominada  $Matriz[i][j]$ . Cuando se activa BTN2, el microprocesador realiza el cálculo de la matriz producto de puntos **B** y se almacenan los valores de cada uno de los elementos en una variable denominada  $MatrizB[i][j]$ .

Finalmente con BTN3, se envían los datos almacenados en el arreglo matricial **B** por el puerto serie de la FPGA hacia el ordenador, donde se ejecuta una interfaz gráfica programada en Matlab. Esta interfaz será la encargada de almacenar dichos datos y de realizar el cálculo de la descomposición en valores singulares de la matriz **B**, obteniendo las matrices **U** de autovectores, **S** de autovalores, y realizando con ellas el cálculo de la matriz de coordenadas estimadas  $\mathbf{X}^*$ .

## V. PRUEBAS EXPERIMENTALES

A fin de realizar las pruebas experimentales, como se explicó anteriormente, se parte de un conjunto de datos que se corresponden a las relaciones temporales existentes entre los distintos nodos de un sistema, y luego de todo un proceso de cálculos matemáticos, se pueden obtener las configuraciones geométricas de los mismos.

Por tanto, de acuerdo a la Fig. 5, se simulará una disposición geométrica específica de un sistema de cuatro nodos a partir del cual se podrán calcular las distancias existentes entre todos los elementos, obteniendo de esa forma la matriz de distancias **D**. Para ello, se emplea una interfaz gráfica creada mediante el software Matlab (véase la Fig. 6).

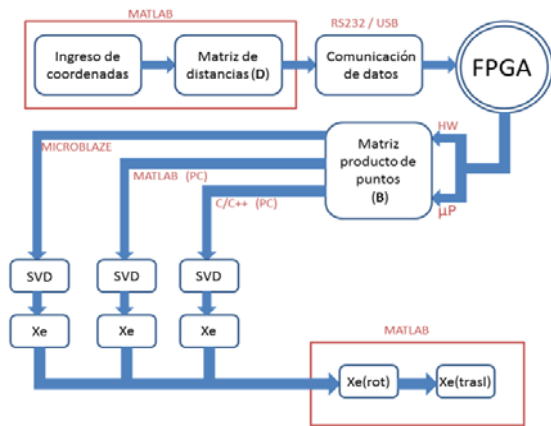


Fig. 5. Fig. 5. Diagrama de bloques básico para la realización de las pruebas experimentales

Luego, se realiza una comunicación por puerto serie entre la interfaz de ingreso de datos y los registros de la FPGA, de forma tal de poder calcular la matriz producto de puntos **B** directamente desde la FPGA.



Fig. 6. Interfaz gráfica implementada en MATLAB para la comunicación de datos en las pruebas experimentales

Por un lado, se ha utilizado directamente la implementación hardware, programando la FPGA mediante el lenguaje VHDL. Por el otro, se ha utilizado el procesador Microblaze previamente embebido en la placa, programándolo mediante el lenguaje C. Una vez que se cuenta con la matriz **B** se ha procedido a computar el algoritmo de descomposición en valores singulares [9], y finalmente se ha llevado a cabo la estimación de las coordenadas. La Fig. 8 muestra los resultados obtenidos en el posicionamiento de una topología de 4 nodos luego del acondicionamiento (rotación y traslación respecto al nodo 1 y nodo 2) de las coordenadas según lo detallado en el Paso 4 de la Fig. 2. En este caso existe un error en la estimación de las posiciones debido a la resolución con la cual se operan los datos digitalizados en **D**.

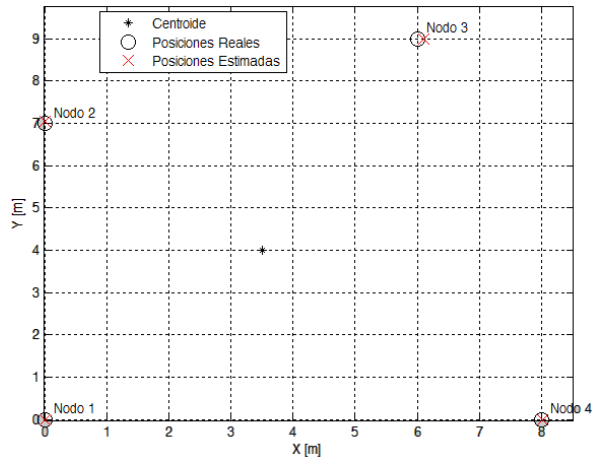


Fig. 7. Resultados en la estimación de las coordenadas de 4 nodos distribuidos en un entorno en 2D.

En la Fig. 8 y la Fig. 9 se presentan los resúmenes de recursos utilizados al implementar las operaciones aritméticas para el cálculo de los elementos de **B** directamente sobre la FPGA y al utilizar el procesador embebido.

Como se puede observar, el número de slices ocupado para el caso de la implementación en hardware es del 25% (véase Fig. 8), mientras que el microprocesador demanda el 43% de recursos (véase Fig. 9).

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	1,022	17,344	5%	
Number of 4 input LUTs	3,312	17,344	19%	
Number of occupied Slices	2,237	8,672	25%	
Number of Slices containing only related logic	2,237	2,237	100%	
Number of Slices containing unrelated logic	0	2,237	0%	
Total Number of 4 input LUTs	3,964	17,344	22%	
Number used as logic	3,312			
Number used as a route-thru	652			
Number of bonded IOBs	25	250	10%	
Number of BUFMUXs	2	24	8%	
Average Fanout of Non-Clock Nets	1.88			

Fig. 8. Resumen de utilización de recursos de la FPGA mediante implementación en Hardware

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	3,551	17,344	20%	
Number of 4 input LUTs	4,769	17,344	27%	
Number of occupied Slices	3,795	8,672	43%	
Number of Slices containing only related logic	3,795	3,795	100%	
Number of Slices containing unrelated logic	0	3,795	0%	
Total Number of 4 input LUTs	4,926	17,344	28%	
Number used as logic	4,167			
Number used as a route-thru	157			
Number used for Dual Port RAMs	256			
Number used as Shift registers	346			
Number of bonded IOBs	101	250	40%	
Number of RAMB16s	23	28	82%	
Number of BUFMUXs	2	24	8%	
Number of DCMs	1	8	12%	
Number of BSCANs	1	1	100%	
Number of MULT18X18SD0s	3	28	10%	
Average Fanout of Non-Clock Nets	3.43			

Fig. 9. Resumen de utilización de recursos de la FPGA mediante implementación con Microblaze

## CONCLUSIONES

En este trabajo se ha evaluado la viabilidad de implementación de la técnica de escalado métrico multidimensional en dispositivos de lógica programable. Un paso importante de este algoritmo es el cálculo de la matriz producto de puntos. Se han evaluado dos variantes, obteniendo dicha matriz por software, y por un diseño de hardware. Se pudo observar que con en el diseño de hardware, se empleó una menor cantidad de bloques lógicos, y no se utilizaron los bloques de memoria RAM interna. Por otro lado, con la implementación del Microblaze, aumentó considerablemente el consumo de recursos. Tal es así, que se utilizó el 82% de los bloques de memoria interna, y un 18% más de bloques lógicos que en el caso del diseño en hardware. Cabe destacar que en la implementación por Hardware, si se desean añadir nuevos procesos o cálculos, esto se verá reflejado en un aumento de los recursos demandados. Sin embargo, se puede aumentar el procesamiento de datos dentro de la programación del Microblaze sin aumentar el consumo de recursos para la ejecución por ejemplo de las técnicas de descomposición en valores singulares.

## REFERENCIAS

- [1] H. Gellersen, P. Lukowicz, M. Beigl, and T. Riedel, "Cooperative Relative Positioning," *Pervasive Computing, IEEE*, vol. 9, pp. 78-89, 2010.
- [2] W. S. Torgerson, "Multidimensional Scaling: Theory and Method," *Psychometrika*, vol. 17, pp. 401-419, 1952.
- [3] S. L. France and J. D. Carroll, "Two-Way Multidimensional Scaling: A Review," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, pp. 644-661, 2011.
- [4] V. Raykar, I. Kozintsev, and R. Lienhart, "Position calibration of microphones and loudspeakers in distributed computing platforms," *IEEE Transactions on Speech and Audio Processing*, vol. 13, pp. 70-83, January 2005.
- [5] C. De Marziani, J. Ureña, A. Hernández, M. Mazo, J. J. García, A. Jiménez, M. C. Perez, F. Alvarez, and J. M. Villadangos, "Acoustic Sensor Network for Relative Positioning of Nodes," *Sensor Basel* vol. 9, pp. 8490-8507, 2009.
- [6] C. De Marziani, J. Urena Urena, A. Hernandez Alonso, J. Garcia, F. Alvarez, A. Jimenez, C. Perez Rubio, J. Villadangos, J. Aparicio, and R. Alcoleas, "Simultaneous Round-Trip Time-of-Flight Measurements with Encoded Acoustic Signals," *Sensors Journal, IEEE*, vol. PP, pp. 1-1, 2012.
- [7] I. Diligent, "Diligent Inc. Digital Design Engineer's Sources." vol. 2012 <http://www.diligentinc.com/Products/Detail.cfm?Prod=NEXYS2>, 2012.
- [8] Xilinx. (01 Marzo 2013). Microblaze Soft Processor. Available: <http://www.xilinx.com/tools/microblaze.htm>
- [9] B. Kollman and D. Hill, *Algebra Lineal*, 2006.