

Propuesta Arquitectónica para el Desarrollo de Sistemas Multi-Sensor Empleando Tecnología Inalámbrica

Walter Mata, Alberto Ochoa, Apolinar González, Ramón Félix
Facultad de Ingeniería Mecánica y Eléctrica, Universidad de Colima, 28400, Coquimatlán (Colima), México

Y

Alfons Crespo
Dpto. de Informática de Sistemas y Computadores, Universidad Politécnica de Valencia, 46022, Valencia, España

Resumen—Los sistemas distribuidos necesitan esquemas de comunicación eficientes para que sus nodos puedan interactuar y funcionar de manera exitosa. Para realizar la comunicación entre estos nodos surge la necesidad de conocer los aspectos relacionados con el dominio de aplicación del sistema, entre ellos podemos mencionar: tipo de sistema operativo, tipo red, protocolos de comunicación, lenguaje de programación, tipos de datos, arquitectura, etc. En este artículo se presenta un sistema multi-sensor que transmite la información codificada al medio siendo recibida por un receptor que procesa dichos datos, todo esto basado en el patrón arquitectónico Publish/Subscribe. De manera interna un middleware se encarga de que la señal recibida sea gestionada para su procesamiento en varios agentes de procesamiento, con los cuales se obtienen tres parámetros importantes (macro-secuencia transmitida, tiempo de emisión y posición de la baliza transmisora publicadora). Con dicha información se crea una matriz de datos que posteriormente puede ser usada para alguna aplicación específica de cómputo ubicuo.

Temas claves—Arquitectura publicar/suscribir, Sistemas multi-sensor, Tecnología inalámbrica.

I. INTRODUCCIÓN

El presente trabajo propone el desarrollo de un Modelo/ Metodología para facilitar el desarrollo de aplicaciones distribuidas embebidas inalámbricas basadas en el patrón arquitectónico Publish/Subscribe, para lo cual fue necesario realizar el análisis, diseño, desarrollo e implementación de una capa intermedia de software (middleware) basada en un subconjunto de la especificación del estándar DDS (Data Distribution Service) realizado por la OMG el cual está basado en el patrón arquitectónico en cuestión, Publish/Subscribe [1]. Dicha capa de software basada en [2] permite facilitar el proceso de desarrollo de aplicaciones embebidas distribuidas inalámbricas, las cuales presentan generalmente restricciones en cuanto al procesamiento de cómputo, poca memoria, periféricos limitados, restricciones energéticas, etc.

El desarrollo y la implementación del middleware están orientados en la difusión de datos a través de tópicos en los sistemas distribuidos inalámbricos. Las aplicaciones

desarrolladas utilizan como mecanismo de comunicación el modelo publish/subscribe, puesto que utiliza características de calidad de servicio (QoS) para el envío/recepción de mensajes entre los diferentes nodos del sistema distribuido.

La arquitectura general de comunicación de los nodos se presenta en la Figura 1, en la cual el middleware corresponde a una capa de software que se encarga de abstraer la complejidad de las capas subyacentes a la aplicación, permitiendo a su vez la capacidad de integración con una variedad de sistemas operativos, dándole la fortaleza para poder interactuar sobre sistemas heterogéneos.

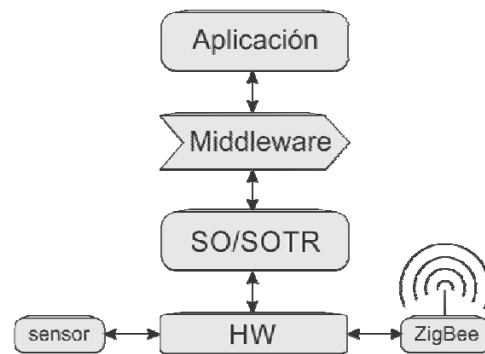


Figura 1. Arquitectura de Comunicación General de los Nodos.

II. INTRODUCCIÓN AL DDS

El modelo de programación según la especificación del DDS está clasificado en dos capas [3].

- DCPS (Data-Centric Publish-Subscribe), la cual provee la funcionalidad requerida por una aplicación para publicar y suscribirse a los tópicos; distribuir los datos; modelar el servicio por medio de las entidades DomainParticipant, Topic, DataWriter, Publisher, DataReader y Subscriber; configurar por medio de los QoS.
- DLRL (Data Local Reconstruction Layer), es una capa opcional, que construye una cache de objetos

locales que permite a las aplicaciones acceder a los datos como si fueran locales.

En las Figuras 2 y 3 se muestra el modelo de publicación (Publisher) y suscripción (Subscriber) respectivamente, dado por la especificación de la OMG. Como se puede observar en el modelo de publicación el tópic, es recibido por el DataWriter y publicado al Espacio Global de Datos, mientras que en el modelo de suscripción el DataWriter toma o lee el dato del dominio a través del subscriber.

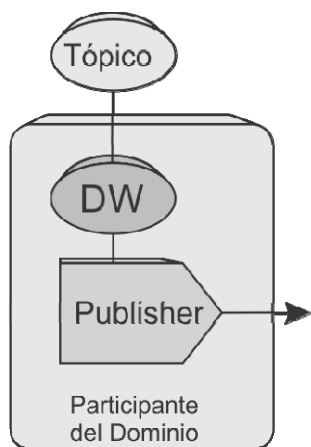


Figura 2. Modelo del Publicador

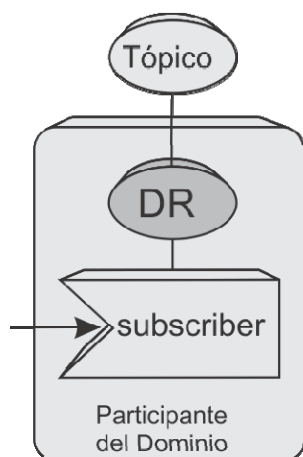


Figura 3. Modelo del Suscriptor

En el trabajo desarrollado, se propone utilizar un modelo tipo Multi-Publicador/Suscriptor (MPS) para la realización de las aplicaciones de los nodos; en las Figuras 4 y 5 se muestran los modelos de los Multi-Publicadores (MP) y los Multi-Suscriptores (MS) respectivamente, mientras que en la Figura 6 se ejemplifica el uso de los Multi-Publicadores/Suscriptores.

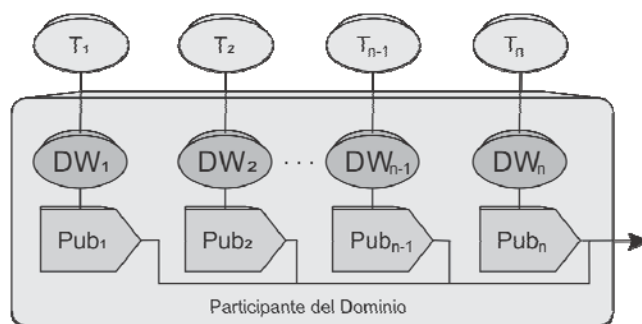


Figura 4. Modelo del Multi-Publicador

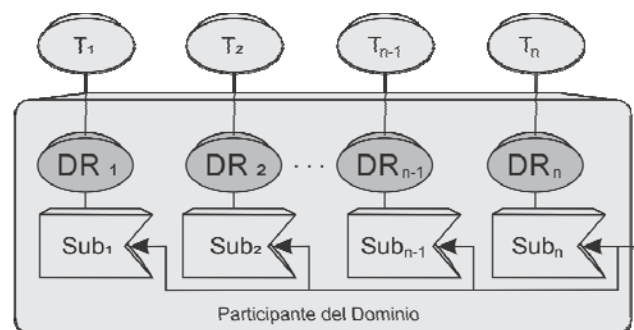


Figura 5. Modelo del Multi-Suscriptor

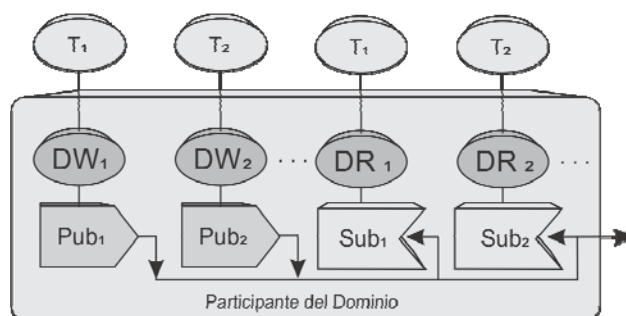


Figura 6. Modelo del Multi-Publicador/ Suscriptor

Puesto que la meta de la especificación del DDS es facilitar la distribución eficiente de datos en un sistema distribuido y que todos los nodos participantes puedan leer o escribir (publicar o suscribirse) los datos de manera eficiente, ofrece un espacio global de datos (Global Data Space) en el que cualquier participante puede leer o escribir. El middleware distribuye los datos para que cada participante suscriptor pueda tener acceso a los valores más recientes. En la Figura 7 se presenta el esquema general de este espacio.

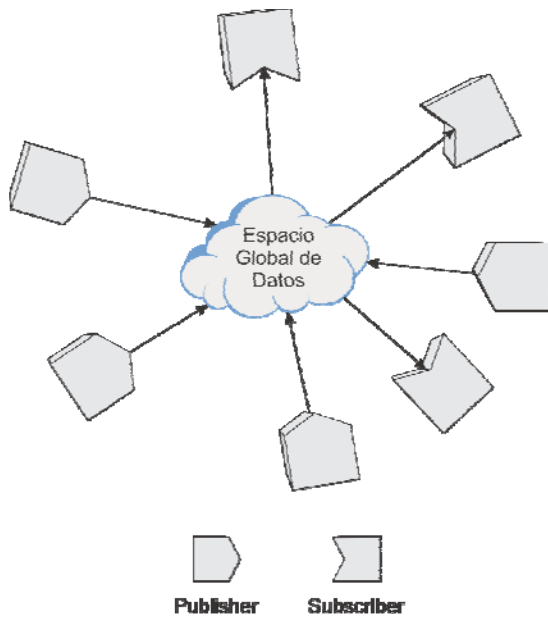


Figura 7. Espacio Global de Datos

Para la implementación de los Multi-Publicadores y Multi-Suscriptores, se utiliza una base de datos en memoria, la cual permite de esta manera que los nodos cuenten con un espacio local de datos, lo que agiliza el proceso de envío si es que los datos se necesitan dentro del mismo nodo para ser procesados por alguna aplicación y no tienen algunos necesidad de ser distribuidos fuera del mismo nodo. En la Figura 8 se presenta el modelo Local de Datos propuesto.

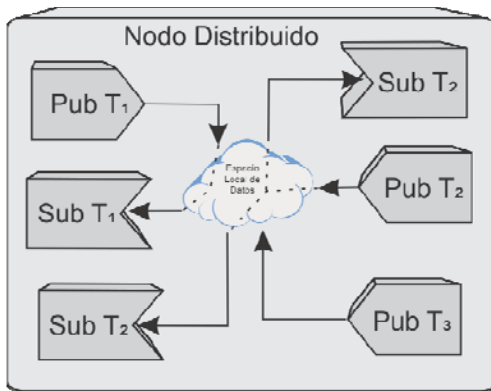


Figura 8. Espacio Local de Datos

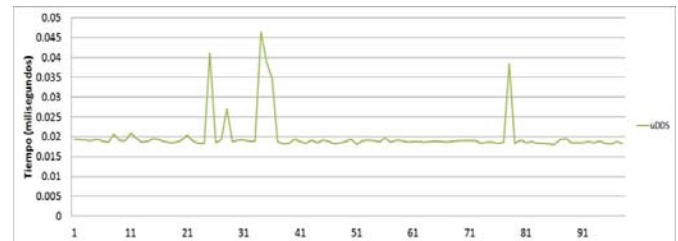
Cada Publicador o Suscriptor del nodo Distribuido se ejecuta en un hilo, el cual mediante un despachador asigna los turnos a cada una de estas entidades, evaluando constantemente cada uno de los parámetros de Calidad de Servicio establecidos para cumplir con ellos.

III. PRUEBAS DE RENDIMIENTO.

Las pruebas de rendimiento realizadas en cada uno de los nodos distribuidos muestran la latencia del comportamiento interno de cada nodo; se realizaron con un sistema operativo Ubuntu 10.04 LTS 32-bits (Núcleo Linux 2.6.32-47-

generic), en una computadora personal HP Compaq 8100 Elite Convertible Minitorre (procesador intel core i7, 8 GB de memoria RAM y disco duro de 1 TB).

En la Gráfica 1 se muestran los resultados obtenidos al medir el tiempo de creación del publicador; se ejecutó un programa publicador con un sólo proceso de creación de publicador, se tomaron 100 muestras y se graficaron con su respectivo tiempo de ejecución. El mismo proceso se realizó con los suscriptores, observándose los resultados en la Gráfica 2. En la tabla 1 se muestran el promedio, los máximos y mínimos de los tiempos de creación de cada una de las entidades.



Gráfica 1. Latencia de creación de publicadores.



Gráfica 2. Latencia de creación de suscriptores

TABLA 1.
DATOS DE CREACIÓN DE PUBLICADORES Y SUSCRIPTORES

	Promedio	Máximo	Mínimo
<i>Publicador</i>	0.01897655	0.055883	0.006842
<i>Suscriptor</i>	0.01881033	0.078937	0.007015

La siguiente prueba de rendimiento ejecuta un programa publicador incrementando el número de procesos de creación de publicadores.

Las Figuras 9 y 10 muestran el tiempo que dura el proceso con un publicador y suscriptor respectivamente, posteriormente dos y así sucesivamente hasta llegar a diez publicadores y suscriptores. La prueba se ejecutó hasta obtener 1000 muestras en cada uno de los procesos de creación.

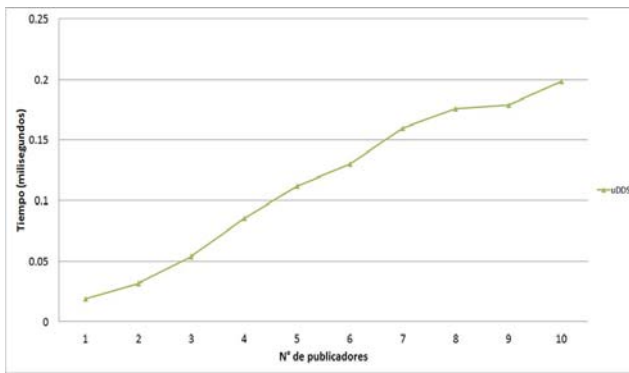


Figura 9. 10 publicadores

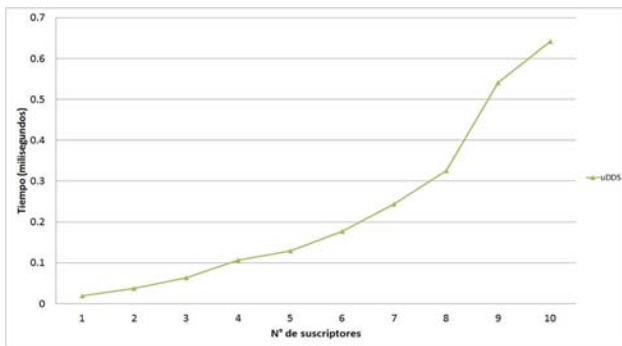


Figura 10. 10 suscriptores

IV. ARQUITECTURA APLICADA A UN SISTEMA MULTI-TRANSDUCTOR ULTRASONICO

Para verificar la eficiencia de la arquitectura de Multi-Publicador/Suscriptor propuesta en este trabajo, se ha seleccionado un caso de estudio que se basa en la comunicación de un grupo de balizas ultrasónicas y un receptor móvil que están ubicadas dentro de una habitación.

El sistema propuesto se caracteriza por la transmisión o publicación simultáneamente de información al medio a través de las balizas ultrasónicas y la recepción o suscripción de una única señal con el receptor móvil. La señal recibida por el receptor está conformada por la suma de las señales transmitidas por las balizas. El escenario de la prueba se muestra en la Figura 11, donde se encuentran ubicadas en un espacio interior 4 balizas (beacons) fijas identificadas por T_{X_i} , con $i = \{1, 2, 3, 4\}$, además de un receptor móvil R_x . La información transmitida por las balizas es la posición P_{X_i} , el tiempo de disparo t_i y la macro-secuencia (MS_i) que identifica la baliza i , esto es: $\{P_{X_i}, t_i, MS_i\}$.

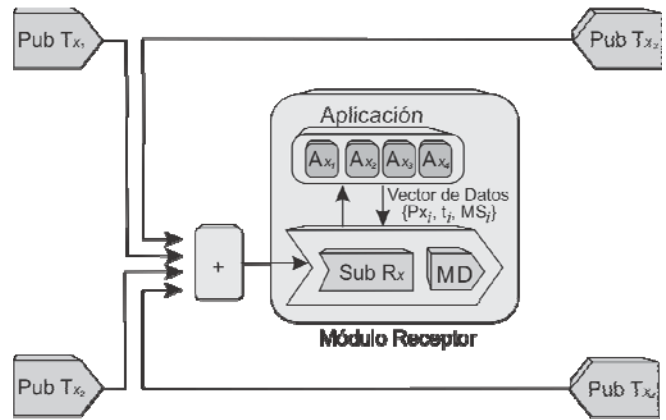


Figura 11. Escenario de Prueba del sistema.

Cada baliza tiene asignada una macro-secuencia única y particular, formada a partir de un conjunto complementario de M secuencias [4], las cuales tienen excelentes propiedades de auto-correlación y correlación cruzada. La ventaja de usar estas macro-secuencias es que pueden transmitirse simultáneamente al medio y la interferencia entre ellas es nula. Para mayor información relacionada con las características de las macro-secuencias puede consultarse las referencias bibliográficas [4], [5] y [6]. Para las macro-secuencias seleccionadas, se logra tener un valor de auto correlación igual a $M \cdot L$ (que es la longitud de la Macro-secuencia) para un desplazamiento nulo, es decir, cuando logras empalmar la secuencia dentro de otra señal. Así mismo, se puede mencionar que en el caso de la correlación cruzada, se logra tener un valor cercano a cero para cualquier otro desplazamiento. En la Figura 12 se muestra una gráfica donde se ejemplifica el valor que se obtiene durante el proceso de correlación de la señal recibida vs la señal transmitida. En esta grafica se pueden observar dos momentos, el instante en que $MS_{1_referencia}$ coincide con el punto donde se encuentra la $MS_{1_transmitida}$ y se genera un pico de correlación; en los demás instantes donde no se logra empalmar la señal da como resultado una señal nula como se observa [4], [5],[6].

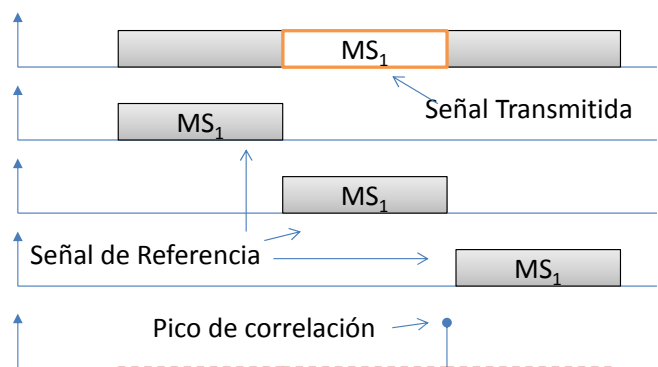


Figura 12. Funciones de AC y CC de una macro-secuencia generada por concatenación a partir de un 32-CSS [4], [5], [6].

El proceso que se lleva a cabo durante la transmisión y recepción de la información se muestra en la Figura 13. El modelo presentado en la figura permite seguir paso a paso la transferencia de información desde que ésta es generada, codificada y modulada por cada uno de los transmisores T_{Xi} .

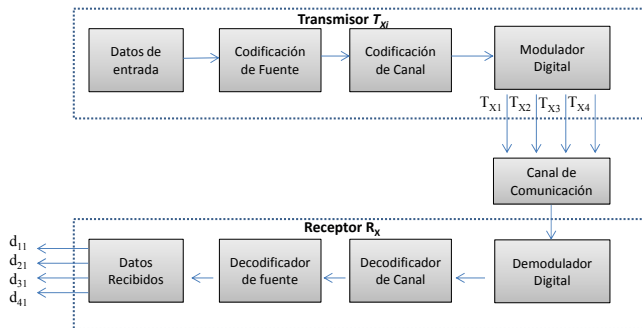


Figura 13. Diagrama de bloques que describe proceso de transmisión y recepción de información.

La señal recibida por el nodo MPS en R_X , que es la suma de todas las señales transmitidas, es demodulada y decodificada tomando como referencia cada una de las señales pre-seleccionadas (macro-secuencias) asignadas a cada transmisor. El proceso que se lleva a cabo en el receptor es que la señal captada es procesada por 4 módulos de correlación dentro del mismo receptor. Cada módulo de correlación, en este caso **Agente X_i** , tiene como dato de entrada la señal captada por el receptor R_X y una señal de referencia única por módulo, la cual es una macro-secuencia MS_i que fue asignada a cada modulo transmisor T_{Xi} . De forma tal que, a la salida de cada Agente se obtendrá como resultado del procesamiento el valor de la función de auto correlación (en el caso de que se encuentre la señal buscada) o de la correlación cruzada.

En la figura 14 se muestra el proceso interno que se lleva a cabo dentro del receptor considerando que información recolectada pasa a través del middleware, observándose cómo la aplicación a través de 4 módulos, denominados agentes A_{Xi} , procesan la información distribuida por el middleware, los cuales tienen como entrada la señal recibida por R_X y la macro-secuencia MS_i . El conjunto de agentes entrega una matriz de información con datos como $\{P_{Xi}, t_i, MS_i\}$. Cabe mencionar que a partir de la detección e identificación de las macro-secuencias por los agentes, se puede estimar la posición de las balizas transmisoras, así como el instante en que se transmitió la señal por la baliza.

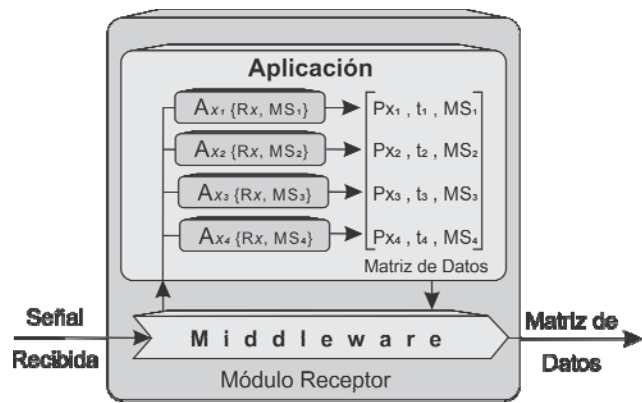


Figura 14. Diagrama a bloques del procesamiento de información llevado a cabo en el receptor para estimar la matriz de datos.

En la Figura 15 se muestran las señales procesadas en un receptor ultrasónico para la transmisión de 4 balizas ultrasónicas, que emiten cada una de ellas una diferente y única macro-secuencia asignada a cada baliza. Así mismo, en esta imagen puede apreciarse como en el receptor, aunque las señales se transmitan simultáneamente y que en el receptor se reciban todas las señales mezcladas, el sistema es capaz de identificar cada MS_i transmitida empleando técnicas de correlación. Para la determinación de la distancia

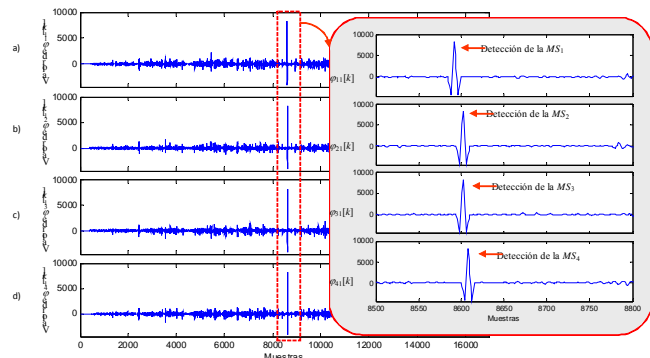


Figura 15. Detección de cada una de las macro-secuencias transmitidas por las balizas en el receptor R_X mediante técnicas de correlación.

En la Figura 16 se muestra la estimación de las distancias entre cada baliza y el receptor R_X . Para el cálculo de dichas distancias se tiene que tomar en cuenta la cantidad de muestras transmitidas, los símbolos de modulación y el periodo de muestreo. En base a estos parámetros se determina el tiempo de vuelo de las señales ultrasónicas, y a partir de estos tiempos pueden calcular las distancias d_{11} , d_{21} , d_{31} , y d_{41} .

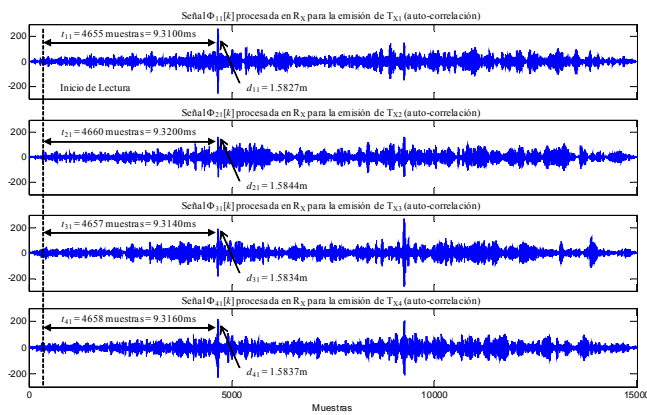


Figura 16. Estimación de las distancias entre cada baliza transmisora y el receptor R_X a partir de las funciones de auto-correlación de las MS.

V. CONCLUSIONES Y TRABAJOS FUTUROS

En el presente artículo se hace uso de un middleware basado en un subconjunto de la especificación DDS de la OMG, para ser utilizado en un Sistema Multi-Transductor Ultrasónico. El sistema propuesto está formado por 4 balizas ultrasónicas que extraen datos del entorno y publican la información sensada, transmitiéndola de manera inalámbrica a un nodo receptor. El nodo (receptor) suscrito a los tópicos de estas procesa la información, mediante el uso de agentes, en la capa de aplicación y obtienen una matriz de datos que indican la posición del objeto de manera local. Los algoritmos desarrollados que detectan la macro-secuencia transmitida permiten obtener dos datos adicionales, la posición 3D del nodo receptor, así como la distancia que existe entre cada baliza y el receptor, a partir del tiempo de transmisión de la señal ultrasónica.

Entre los trabajos futuros se pretende formalizar la interacción interna del middleware, mediante herramientas formales y pruebas de rendimiento en un prototipo real.

VI. AGRADECIMIENTOS

La realización de este proyecto fue gracias al proyecto FRABA (Fondo Ramón Álvarez Buylla de Aldana) "Diseño e Implementación de una Red de Sensores Inalámbricos para Aplicaciones de Monitorización y Posicionamiento en Espacios Interiores" aprobado en la convocatoria 2014.

VII. REFERENCIAS BIBLIOGRAFICAS

- [1] S. Tarkoma, Publish / Subscribe Systems: Design and Principles. Wiley, 2012.
- [2] A. González, W. Mata, L. Villaseñor, R. Aquino, J. Simo, M. Chávez, and A. Crespo, "μDDS: A Middleware for Real-time Wireless Embedded Systems," Journal of Intelligent & Robotic Systems, vol. 64, no. 3–4, pp. 489–503, 2011.
- [3] OMG, Data Distribution Service for Real-Time Systems Version 1.2. OMG Technical Document (2007)

[4] C. De Marziani, J. Ureña, A. Hernández, M. Mazo, F. Alvarez, J. Garcia, and P. Donato, "Modular architecture for efficient generation and correlation of complementary set of sequences," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 2323–2337, May 2007.

[5] Ochoa, A.; Ureña, J.; Hernández, A.; Mazo, M.; De Marziani, C.; Pérez, M.C., "Processing Algorithm for obtaining the Impulse Response in a MIMO Ultrasonic System," Emerging Technologies and Factory Automation, 2006. ETFA '06. IEEE Conference on , vol., no., pp.977,980, 20-22 Sept. 2006.

[6] Pérez Rubio, M.C.; Sanz Serrano, R.; Ureña Ureña, J.; Hernández Alonso, A.; De Marziani, C.; Álvarez Franco, F.J., "Correlator Implementation for Orthogonal CSS Used in an Ultrasonic LPS," *Sensors Journal, IEEE* , vol.12, no.9, pp.2807,2816, Sept. 2012

Currículo corto de los autores

Walter Mata es Profesor de Tiempo Completo, adscrito a la Facultad de Ingeniería Mecánica y Eléctrica de la Universidad de Colima; imparte cátedra en la Carrera de Ingeniero en Sistemas Computacionales y en la Maestría en Ingeniería. Su línea de investigación está relacionada con la automatización y los sistemas embebidos.

Alberto Ochoa es Profesor de Tiempo Completo, adscrito a la Facultad de Ingeniería Mecánica y Eléctrica de la Universidad de Colima. El Dr. Ochoa imparte cátedra en los programas educativos Ingeniería en Sistemas Electrónicos y Telecomunicaciones y la Maestría en Ingeniería. Sus líneas de investigación son procesamiento digital de señales, estructuras sensoriales para aplicaciones robóticas, y clasificación y localización de objetos empleando algoritmos inteligentes.

Apolinar González es Profesor de Tiempo Completo, adscrito a la Facultad de Ingeniería Mecánica y Eléctrica de la Universidad de Colima; imparte cátedra en la Carrera de Ingeniero en Sistemas Computacionales y en la Maestría en Ingeniería de Procesos. Su línea de investigación está relacionada con la automatización y los sistemas embebidos.

Ramón Félix es Profesor de Tiempo Completo, adscrito a la Facultad de Ingeniería Mecánica y Eléctrica de la Universidad de Colima. El Dr. Félix imparte cátedra en los programas educativos Ingeniería en Sistemas Electrónicos y Telecomunicaciones y la Maestría en Ingeniería. Su línea de investigación es el control no lineal para sistemas dinámicos.

Alfons Crespo es Catedrático del Dpto. de Informática de Sistemas y Computadores de la Universidad Politécnica de Valencia; Sus líneas de investigación incluyen sistemas de tiempo real (planificación, hardware, control, etc.) y técnicas de virtualización.