

SIMULADOR DE REDES DE SENSORES ACÚSTICOS PARA APLICACIONES DE CONTROL DE CAMPO SONORO

REFERENCIA PACS: 43.60.Qv

Antoñanzas, Christian; Estreder, Juan; Ferrer, Miguel; Gonzalez, Alberto
Inst. de Telecom. y Aplicaciones Multimedia (iTEAM), Univ. Politécnica de València (UPV), Camino de
Vera, s/n, 46022, Valencia, Tel:+34963877007, ext. 73008, Fax:+34963877309, chanma@iteam.upv.es

ABSTRACT

In this work, a tool implemented in MATLAB to develop sound field control applications using acoustic sensor networks is proposed. This tool allows define acoustic nodes, that is, devices capable of obtaining information of the sound field from one or more sensors, capable of processing that information and modifying the sound field via one or more actuators. Moreover, it should possess the ability to communicate with other network nodes. The tool simulates the data acquisition and generation performed in real-time applications, the distributed processing and the exchanging information among the nodes allowing us to model the existing communication problems between the network and the nodes.

RESUMEN

En este trabajo se describe una herramienta implementada en MATLAB para desarrollar aplicaciones de control de campo sonoro usando redes de sensores acústicos. Dicha herramienta permite definir nodos acústicos, es decir, dispositivos capaces de obtener información del campo sonoro mediante uno o varios sensores, procesarla, y modificar dicho campo sonoro mediante uno o varios actuadores, además de comunicarse con otros nodos. La herramienta simula la adquisición y generación de datos en tiempo real, el procesado distribuido y el intercambio de información entre nodos permitiendo modelar los problemas existentes en dicho intercambio.

1. INTRODUCCIÓN

Las redes inalámbricas de sensores acústicas (Wireless Acoustic Sensor Networks, WASN) están consideradas con una solución barata, flexible y eficiente para diferentes aplicaciones acústicas [1], [2]. En estas aplicaciones, los nodos acústicos están compuestos normalmente por micrófonos conectados a un procesador, con algún tipo de capacidad computacional y comunicativa. Existen numerosas aplicaciones para este tipo de nodo acústico [3] pero la mayoría se suelen centrar en la estimación de una señal o un parámetro que puede ser medido por todos los nodos y no interactúan con el medio, es decir, no generan señales. Para aplicaciones de control de campo sonoro, como control activo de ruido o cancelación de crosstalk, estos nodos deben tener la capacidad de actuar sobre el entorno emitiendo sonidos a través de actuadores o altavoces. Además, la red debiera centrarse en el diseño de señales que alimente a los altavoces y por tanto, controlen el campo sonoro. Por lo tanto, definimos nodo acústico como un dispositivo capaz de obtener información de uno o más sensores (micrófonos) y capaz de generar señales vía uno o más actuadores (altavoces) para controlar el campo sonoro. Además, debe tener la habilidad de comunicarse con otros nodos de la red. Smartphones, tablets o portátiles son algunos ejemplos de nodos acústicos.

Basado en esto, en este trabajo se presenta una herramienta para modelar y simular sistemas de control de sonido así como su comportamiento. Es importante diferenciar tres procesos en estos sistemas (Figura 1) para entender el funcionamiento de la herramienta:

- 1) Algoritmos: definen las operaciones a ejecutar entre las señales del sistema y su interacción con los dispositivos de entrada/salida (micrófonos/altavoces) así como con otros elementos de la red.
- 2) Sistema de comunicación: permite el intercambio de información entre los nodos y permite simular los problemas reales en las comunicaciones como latencias, errores de transmisión, pérdida de información,...
- 3) Sistema acústico: simula la propagación por un medio acústico de las señales emitidas por los altavoces de los nodos acústicos hasta que son recogidas por los micrófonos de los mismos. Para ello, se vale de las respuestas al impulso que caracterizan los caminos acústicos (generalmente medidos en un recinto que se quiere recrear o sintetizados para simular espacios virtuales) y que enlazan los diferentes altavoces con cada uno de los micrófonos del sistema.

A pesar de que pueden intercambiar información, los tres procesos trabajan de manera independiente. Este concepto es importante para entender cómo funciona la herramienta de simulación.

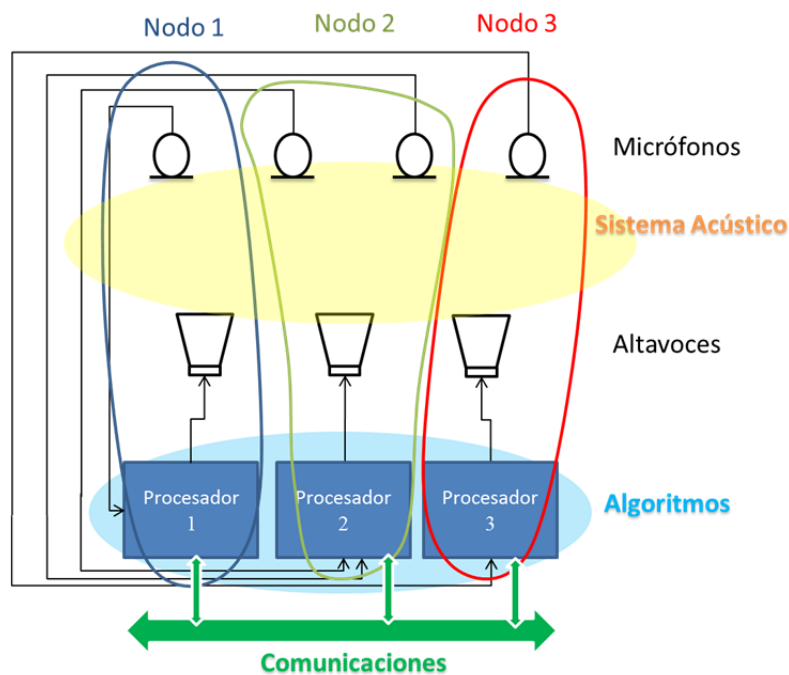


Figura 1: Esquema de sistema de control sonoro con nodos multicanal

El simulador acústico es una herramienta implementada en MATLAB que permite desarrollar aplicaciones de control de campo sonoro destinadas a implementarse en WASN, sin necesidad de disponer físicamente de los nodos acústicos, sino definiendo solamente sus características y los algoritmos que ejecutaría cada uno de ellos, caracterizando a su vez, la red de comunicaciones que emplearían los nodos en la transferencia de información y el sistema acústico (recinto o sala) en el que los nodos estarían ubicados.

El principal objetivo de esta herramienta es poder desarrollar y simular algoritmos de procesamiento de audio multicanal destinados a ser ejecutados en DSP o cualquier otro sistema de adquisición y generación de datos en tiempo real, así como poder analizar el comportamiento de los mismos, siendo una de las principales características del simulador que cada nodo puede usar procesamiento de señal distribuido si la red de comunicaciones lo permite. Este procesamiento tiene como objetivo obtener una señal de salida en cada nodo como resultado de procesar una serie de señales propias en cada nodo junto con cierta información compartida proveniente de otros nodos de la red. Cada nodo procesa las señales independientemente y cada uno es relevante para el funcionamiento global del sistema de acuerdo con la aplicación deseada. Además, la herramienta puede simular los problemas existentes en las comunicaciones entre red y nodos así como una adquisición de datos no ideal y muy próxima a la realizada en aplicaciones en tiempo real.

El simulador presenta otras virtudes como el almacenamiento, reproducción y análisis de todas las señales de audio o parámetros de la simulación así como herramientas para la definición, modificación o estudio del sistema acústico. Por lo tanto, la herramienta permite el paso entre la idea matemática inicial de cada algoritmo y su programación final sobre plataformas digitales. A continuación, se presenta la descripción del funcionamiento del simulador.

2. DESCRIPCIÓN DEL SIMULADOR

2.1 ¿Cómo está formado el Simulador?

El simulador está basado en un tipo de dato de MATLAB llamado estructura (struct). Una estructura es un conjunto de datos organizado por grupos llamados campos. Cada campo puede contener cualquier tipo de dato con cualquier tamaño.

Se puede construir estructuras de dos maneras:

- Mediante la función struct donde los argumentos de entrada son los nombres de los campos y sus respectivos valores.
Ejemplo: `structArray = struct('campo1',valor1,'campo2',valor2,...)`
- Mediante asignación creando los distintos campos uno a uno y estableciendo un valor específico para cada uno.
Ejemplo: `(structArray.campo1='valor1'; structArray.campo2=valor2,...)`

El acceso a datos se realiza mediante el operador punto seguido del nombre del campo al que queramos acceder (ej: `structNombre.campoNombre`).

Es posible crear arrays de estructuras pero todas deben tener el mismo número de campos. A su vez, todos los campos de los mismos deben compartir el mismo nombre aunque pueden contener diferente tipo y tamaño de datos. Además, una estructura permite mecanismos de almacenamiento jerárquico para contener diferentes tipos de datos. Se permiten estructuras anidadas y crear vectores o matrices de estructuras. También, es posible añadir nuevos campos a una estructura en cualquier momento. Además, existen funciones optimizadas para realizar operaciones específicas [4].

2.2 Descripción del Modelo

En esta sección se describen los diferentes pasos para realizar la simulación de un modelo de red acústica definido como el modelo basado en la configuración de todos los parámetros necesarios para simular una aplicación de control de campo sonoro.

El primer paso es inicializar el modelo. Para esto, es necesario configurar todos los procesos del sistema:

- Configuración del sistema acústico: Se carga el archivo que contiene la respuesta al impulso del canal acústico entre cada altavoz y cada micrófono situados en el recinto que se va a simular o sintetizar. En esta parte, se define el número de altavoces y micrófonos que gestionará cada nodo. Es posible crear nodos multicanal formados por un número arbitrario de altavoces y micrófonos.
- Configuración de los algoritmos: El algoritmo a ejecutar por cada nodo se elige en función de la aplicación a simular. Además, se definen los diferentes parámetros necesarios para el funcionamiento de cada algoritmo. Por lo tanto, varios nodos pueden compartir el mismo algoritmo, pero pueden tener diferentes parámetros de configuración.
- Configuración del sistema de comunicaciones: Se define qué tipo de topología tiene la red de comunicaciones. Esto explica como colaboran los nodos. Por ejemplo, un nodo puede intercambiar información con un pequeño grupo de nodos o sólo con sus nodos vecinos. Por otra parte, la red de comunicaciones puede introducir latencias en la transmisión de información a los nodos por lo que es posible simular este retardo jugando con el número de muestras en la recepción o transmisión de datos de cada nodo.
- Configuración del sistema de adquisición de datos: La herramienta intenta simular la adquisición de datos realizada en aplicaciones en tiempo real a través de redes asíncronas. Cada nodo tiene su propio reloj de muestreo y almacena, procesa y envía los datos de forma independiente al resto de nodos. Intentando simular el comportamiento de un DSP, los nodos están en un estado permanente de bajo consumo hasta que se produzca una interrupción debido a la llegada de datos. En ese momento, el nodo realiza el procesamiento correspondiente. Tal procesamiento termina con otra interrupción para volver al estado inicial de bajo consumo del nodo. Así, definimos dos flags de interrupción: lectura/escritura (R/W) y el tiempo de procesado. La interrupción R/W indica el instante de tiempo en el que el nodo puede leer los nuevos datos procedentes de sus micrófonos y puede escribir los datos procesados en la iteración anterior para ser reproducida por los altavoces. El tiempo de procesado indica el tiempo que se necesita para procesar un bloque de muestras. Si el nodo puede leer nuevos datos (interrupción R/W activa), pero el nodo aún está procesando datos anteriores (interrupción tiempo de procesamiento activa también), no se adquirirán ni procesarán datos nuevos. El tiempo de muestras será la unidad mínima de tiempo a la que el simulador funciona. Así, el sistema acústico trabaja muestra a muestra en el dominio del tiempo (para aproximarnos lo más posible al dominio continuo), pero al mismo tiempo, los nodos pueden ejecutar algoritmos implementados por bloques de muestras (en el dominio del tiempo o de la frecuencia). Del mismo modo, la red de comunicaciones trabaja muestra a muestra independientemente de los otros procesos. En resumen, al mismo tiempo cada nodo puede funcionar con un bloque de tamaño diferente, una frecuencia de muestreo diferente y un tiempo de procesado diferente.

A partir de todos los datos de configuración, se crea la estructura de la red y se caracteriza el modelo de red acústica. En resumen, en esta primera parte, se definen todos los datos necesarios para crear una estructura de red en función de la aplicación a simular.

La estructura de la red direcciona la información pero no la procesa. Por lo tanto, la estructura debe tener campos importantes que contengan las variables que van a utilizar los algoritmos, el sistema acústico y el sistema de comunicación para el procesamiento de datos.

Algunos de estos campos son:

- Estado: Campo que contiene las variables de entrada y de salida de cada proceso en el sistema.
- Estado inicial: Campo que contiene los valores iniciales de las variables del campo anterior.
- Histórico: Campo para almacenar las variables que desea analizar.

➔ Ejemplo de estructura

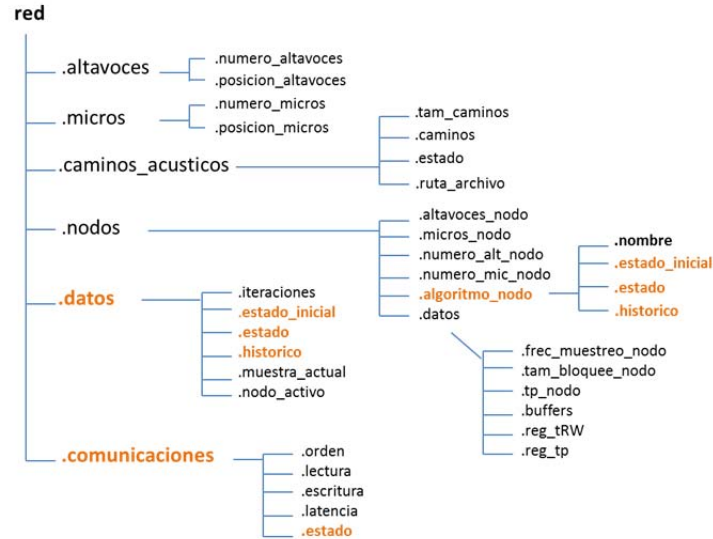


Figura 2: Ejemplo del tipo de estructura usada en el simulador.

Los campos importantes para el funcionamiento de la herramienta están marcados en naranja.

El siguiente paso es realizar la simulación. La herramienta ejecuta los algoritmos en cada nodo y simula si hay latencias en las comunicaciones y/o intercambio de información entre los nodos. Una vez las señales han sido procesadas por los algoritmos y se han enviado a los altavoces, el sistema acústico transforma las señales reproducidas por los altavoces que a su vez son captadas por los micrófonos.

Por tanto, solamente modificando ciertos parámetros de la configuración, se pueden realizar rápidamente diferentes simulaciones para diferentes aplicaciones.

➔ Pasos de la simulación:

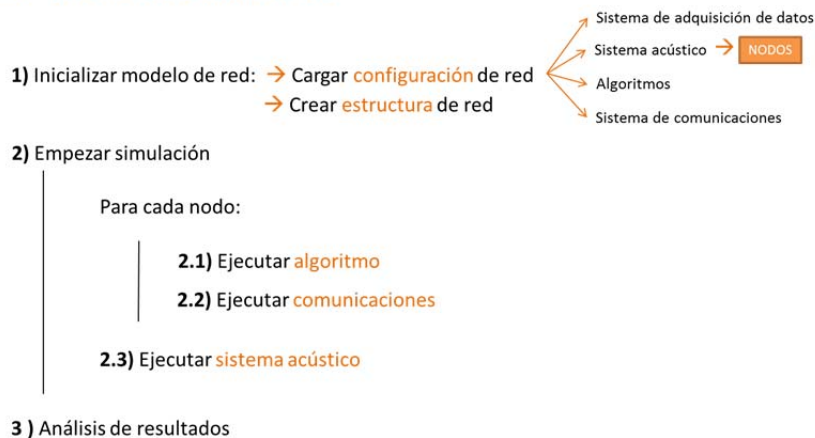


Figura 3: Pasos para realizar la simulación de un modelo de red acústica

2.3 Simulación de los Algoritmos

Cada nodo puede ejecutar cualquier algoritmo compatible con el código de direccionamiento de la herramienta. Los algoritmos pueden trabajar muestra a muestra o por bloques de muestras.

La idea es adaptar el código del algoritmo al formato definido por la estructura para hacer más flexible la herramienta sin importar el tipo de algoritmo. El algoritmo lee valores en ciertos campos de la estructura, ejecuta las operaciones pertinentes y escribe en determinados campos de la estructura.

La definición del algoritmo consta de cuatro partes:

- 1) Declaración: Se declara un nombre para la función que acepta entradas de la estructura y devuelve las salidas a la estructura.
- 2) Lectura: El proceso de lectura se lleva a cabo en el campo estado del algoritmo y en el campo estado del sistema de comunicaciones (si existe cualquier intercambio de información entre los nodos), donde se encuentran las variables necesarias para el funcionamiento del algoritmo.
- 3) Procesado: En esta sección, se ejecutan las operaciones necesarias para la correcta ejecución del algoritmo. Dependiendo del tipo, el algoritmo puede procesar una o varias variables de entrada (correspondientes a las señales de entrada o a las señales captadas por los micrófonos o a parámetros enviados por otros nodos, etc...) y uno o varias variables de salida (correspondientes a las señales que los altavoces tienen que reproducir o a parámetros que necesitan otros nodos, etc...).
- 4) Escritura: Después del procesado y de manera similar al proceso de lectura, los nuevos valores de las variables deben almacenarse en el campo estado tanto del algoritmo de cada nodo como de la red de comunicaciones.

➔ **Ejemplo algoritmo:**

```

%% ALGORITMO FILTRADO + GENERADOR
function [red]=Filtrado_Gen(red)

% Leer campos constantes de la red
nodo=red.datos.nodo_activo;
B=red.nodos(nodo).datos.tam_b_nodo;
% Leer campos constantes del algoritmo
H=red.nodos(nodo).algoritmo_nodo.estado.H;
P=red.nodos(nodo).algoritmo_nodo.estado.P;
tam_fft=red.nodos(nodo).algoritmo_nodo.estado.tam_fft;
% Leer campos variables del algoritmo
buffx=red.nodos(nodo).algoritmo_nodo.estado.buffx;
x_p=red.nodos(nodo).algoritmo_nodo.estado.x_p;
% Leer señal de entrada al nodo
IN=red.nodos(nodo).datos.buf_fentrada;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

buffx=[buffx(B+1:end);IN];
x_p(:,2:end)=x_p(:,1:P-1);
x_p(:,1)=buffx;
X=fft(x_p,tam_fft);
% Filtrado
y=ifft(X.*H);
y=sum(y,2);
y=real(y(B+1:end,1));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Dentro del campo datos guardo la salida del algoritmo
red.nodos(nodo).datos.buf_salida=OUT;

% Almacenar estados del nodo en estructura
red.nodos(nodo).algoritmo_nodo.estado.buffx=buffx;
red.nodos(nodo).algoritmo_nodo.estado.x_p=x_p;

```

Cabecera de la función

Lectura

Procesado

Escritura

Figura 4: Ejemplo de función que modela un algoritmo

2.4 Simulación del Sistema de Comunicaciones

El sistema de comunicación realiza dos tipos de operaciones en el simulador. Por un lado, y como se mencionó anteriormente, es posible simular las latencias (u otros efectos) en la transmisión de información a los nodos retrasando las señales de entrada un cierto número de muestras en cada nodo respectivamente. Por otro lado, si se utilizan algoritmos de colaboración, el sistema permite el intercambio de información entre los nodos. En este caso, la topología de la red explica cómo deben colaborar los nodos. Es decir, para cada nodo, de qué otros nodos puede leer datos y en cuáles otros nodos puede escribir datos.

Esta información, junto con los nodos que colaboran y su orden en la red, debe ser introducida en la configuración de la red de comunicaciones. Luego, cada nodo lee y escribe datos en el ámbito estatal en la red de comunicaciones. Por lo tanto, dependiendo de la topología, la red será responsable de enviar esa información al nodo correspondiente así como de introducir errores o retardos, en su caso.

2.5 Simulación del Sistema Acústico

El sistema acústico está formado por un número de actuadores (altavoces) y un número de sensores (micrófonos), junto con las respuestas al impulso de los caminos acústicos entre ellos. Por lo tanto, el sistema acústico caracteriza un recinto y la posición donde están situados los altavoces y micrófonos dentro de él.

Como se ha mencionado anteriormente, el sistema acústico debe trabajar muestra a muestra en el dominio del tiempo. La forma de programarlo es idéntica a la definición de los algoritmos. En primer lugar, la declaración de la función seguida por una parte en la que se leen las variables necesarias a partir de los campos de la estructura. A continuación, la parte de procesado donde las señales reproducidas por los altavoces son filtradas por los caminos acústicos. Finalmente, en la parte de escritura, las señales captadas por los micrófonos se almacenan en la estructura.

➔ Sistema acústico:

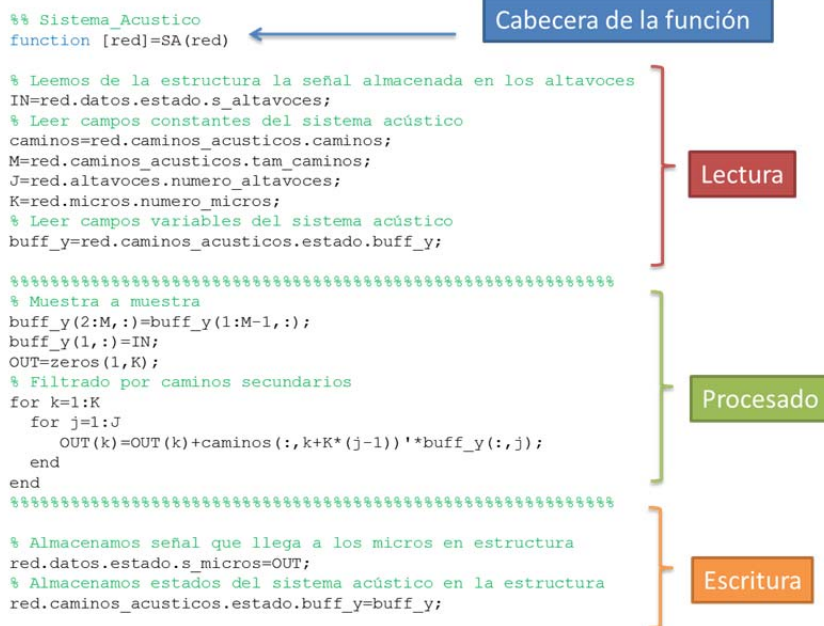


Figura 5: Función que modela el sistema acústico

2.6 Análisis de Resultados

La herramienta nos permite analizar cualquier señal o parámetro existente en el sistema. Usando una función específica, la herramienta almacena en la estructura sólo aquellas señales captadas por los micrófonos o reproducidas por los altavoces seleccionados previamente. Esto reduce la carga computacional y la memoria almacenada, sobre todo en el caso de sistemas multicanal con un gran número de micrófonos o altavoces donde sólo algunas señales capturadas o reproducidas son necesarias.

2.7 Limitaciones

La principal limitación es la falta de interacción entre el usuario y la herramienta. Por lo tanto, la implementación de una interfaz gráfica de usuario es necesaria. Esto mejoraría, sobre todo, la capacidad de análisis de los resultados. Para trabajar con frecuencias de muestreo retardadas que no son múltiplo de la unidad mínima de tiempo, es necesario crear una unidad de tiempo inferior a la muestra. Desde el punto de vista de la programación en Matlab, se ha tratado de vectorizar nuestro código. A pesar de esto, el uso de bucles es inevitable. Por lo que, cuando se utiliza un gran número de nodos, tanto el tiempo de ejecución como el almacenamiento en memoria son considerables. Además, trabajar con diferentes unidades de tiempo aumenta el tiempo de ejecución. Actualmente no se ha implementado ninguna simulación con una resolución numérica diferente de las proporcionadas por Matlab ya que no se han simulado los efectos de la precisión finita. Tampoco se ha simulado el efecto de la cuantificación en los Conversores AD/DA en la entrada y salida de audio, por la que no se han tenido en cuenta efectos tales como la saturación y los errores de cuantificación. Y, por último, la herramienta no es compatible con sistemas acústicos dinámicos o variables.

Todas estas limitaciones son las presentes en la versión actual de la herramienta presentada y actualmente son líneas de desarrollo futuro para su mejora.

3. CONCLUSIONES

En este trabajo se ha presentado una herramienta de simulación que permite el desarrollo de aplicaciones de control de campo sonoro en redes distribuidas cuya principal ventaja es la facilidad y rapidez para simular diferentes sistemas.

La herramienta permite crear nodos con un número arbitrario de micros y de altavoces además de poder ejecutar cualquier algoritmo tanto en el dominio del tiempo, trabajando muestra a muestra, como algoritmos que trabajan por bloques, independientes del tamaño del bloque y trabajando en el dominio del tiempo o de la frecuencia. El simulador ejecuta el sistema acústico muestra a muestra de forma independiente y permite modelar tanto el comportamiento del sistema de comunicaciones simulando los posibles problemas existentes, como la adquisición de datos realizada en aplicaciones en tiempo real. Por último, la herramienta propuesta permite analizar cualquier señal o parámetro existente en el sistema acústico, en los nodos o en la red de comunicaciones, es decir, en cualquier punto del sistema.

4. REFERENCIAS BIBLIOGRÁFICAS

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci. A survey on sensor networks. Communications Magazine, IEEE, 40 (8) 102–114, 2002.
- [2] D. Puccinelli and M. Haenggi: Wireless sensor networks: applications and challenges of ubiquitous sensing. Circuits and Systems Magazine, IEEE, vol. 5, no. 3, pp. 19–31, 2005.
- [3] A. Bertrand: Applications and trends in wireless acoustic sensor networks: A signal processing perspective. Communications and Vehicular Technology in the Benelux (SCVT), 2011 18th IEEE Symposium on, pp.1–6, 2011.
- [4] MathWorks help, on line at:
http://es.mathworks.com/help/matlab/ref/struct.html?s_cid=srchtitle