

UNIVERSIDAD POLITECNICA DE VALENCIA

ESCUELA POLITECNICA SUPERIOR DE GANDIA

Master en Ingeniería Acústica



UNIVERSIDAD
POLITECNICA
DE VALENCIA



ESCUELA POLITECNICA
SUPERIOR DE GANDIA

“Simulador de sistema genérico de reproducción multicanal adaptativo”

TESIS DE MASTER

Autor:

Christian Antoñanzas

Director:

D. Miguel Ferrer

GANDIA, 2011

ÍNDICE

Resumen tesina	3
I. Objeto de la tesina	4
II. Introducción teórica	5
II.1. Sistemas transaurales	5
II.2. Cancelador de crosstalk	6
II.2.1. Soluciones en el dominio frecuencial	7
II.2.2. Soluciones en el dominio temporal.....	10
II.3. Filtrado adaptativo.....	11
II.3.1. El algoritmo LMS.....	12
II.3.2. El algoritmo LMS de filtrado x-LMS	15
III. Implementación práctica	17
III.1. Simulador de sistema de reproducción transaural estático.....	18
III.2. Simulador de sistema de reproducción transaural adaptativo	21
III.3. Simulador de sistema genérico de reproducción multicanal adaptativo.....	23
III.4. Creación del interfaz gráfico del simulador	25
IV. Análisis de resultados	29
IV.1. Comparación de resultados de los simuladores de reproducción transaural	30
IV.2. Verificación de la robustez de los algoritmos	32
V. Conclusiones	40
VI. Líneas futuras	41
Bibliografía	42
ANEXO	43

Simulador de sistema genérico de reproducción multicanal adaptativo

Autor: Christian Antoñanzas

Director: Miguel Ferrer Contreras

RESUMEN

Dentro de la reproducción de audio envolvente, una de las técnicas más empleadas es la reproducción binaural. Al reproducir señales binaurales a través de altavoces, se produce el conocido efecto de crosstalk, donde a cada oído llega información de la señal deseada más una señal interferente proveniente de los demás altavoces. Para evitar eso, se pueden diseñar unos filtros estáticos canceladores que permitan al oyente tener presente en cada oído sólo la señal deseada. El problema está si el oyente, en vez de estar en una posición fija, en un determinado momento cambia de posición o si además queremos cambiar la información de las señales deseadas haciendo que lo que antes se escuchaba en el oído derecho pase a escucharse en el izquierdo y viceversa. De ahí, la necesidad de utilizar filtros que se adapten a situaciones que varían conforme pasa el tiempo. Lo que se pretende con este trabajo es comprobar la robustez y flexibilidad de un simulador multicanal desarrollado usando métodos adaptativos, así como el estudio de diferentes configuraciones de señal de entrada, altavoz y micrófono, ante cambios en el entorno acústico.

I. OBJETO DE LA TESIS

Se pretende desarrollar un sistema genérico de reproducción multicanal para estudiar su robustez frente al desplazamiento relativo de los puntos de control (generalmente micrófonos) respecto a las posiciones virtuales de las fuentes y frente a los cambios en la información deseada en cada punto de control. Por tanto, el sistema de reproducción ha de ser adaptativo. Se desarrollará un entorno de simulación que facilite el análisis y obtención de resultados con el fin de poder implementarse en un futuro en un sistema real.

Para ello comenzaremos con un caso particular, el cancelador de crosstalk y para identificar su problemática, primeramente lo implementaremos de forma estática. Después, realizaremos un sistema de reproducción transaural estéreo (cancelador de crosstalk) adaptativo. Se usará un tipo de algoritmo adaptativo conocido como filtrado-XLMS para obtener las señales deseadas en los puntos de control. A continuación, extrapolaremos el caso anterior a un sistema genérico multicanal y verificaremos mediante simulación la validez de los sistemas de reproducción implementados introduciendo movimientos o variaciones simuladas de las posiciones relativas de las fuentes y puntos de control.

Para la realización de las simulaciones se ha usado el entorno de programación *Matlab*.

II. INTRODUCCIÓN TEÓRICA

II.1. SISTEMAS TRANSAURALES

Un sistema transaural es aquel que pretende reproducir en ciertos puntos del espacio el audio generado mediante altavoces deshaciendo el efecto de la propagación en la sala (reflexiones) y la interferencia entre altavoces. Frecuentemente puede tratarse de un sistema estéreo formado por dos señales a emitir y dos altavoces que reproducen las señales para que un oyente pueda recibirlas en cada oído. El objetivo de estos últimos es que la señal emitida por el altavoz izquierdo sea recibida por el oído izquierdo y que la señal emitida por el altavoz derecho sea recibida por el oído derecho.

Pero la realidad es que aparecen ciertos efectos no deseados que hacen que en cada oído recibamos ambas señales perdiéndose los aspectos relativos a la localización contenidos en la grabación binaural. A este efecto, conocido como diafonía o crosstalk, hay que añadirle el efecto que producen las reflexiones de la sala en la señal emitida por los altavoces.

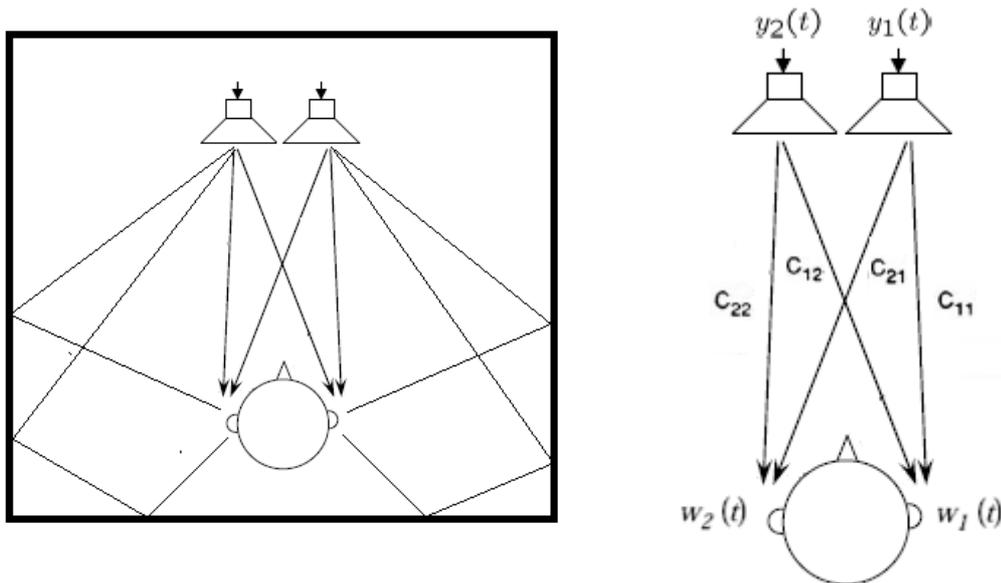


Fig.1 y 2. Sistemas de reproducción estéreo habitual. Efectos de las reflexiones de la sala y diafonía

Emitimos dos señales, $y_1(t)$ e $y_2(t)$ por los altavoces y obtenemos dos señales, $w_1(t)$ y $w_2(t)$ en su oído correspondiente. Los términos c_{11} , c_{12} , c_{21} y c_{22} representan los canales acústicos por los que viajan las señales desde cada altavoz a cada oído. Por lo tanto, representan las respuestas al impulso medidas entre cada par altavoz-oído.

Podemos obtener las ecuaciones que describen este sistema definiendo las señales que se recogen en los oídos como la convolución de las señales emitidas por los caminos acústicos:

$$w_1(t) = y_1(t) * c_{11} + y_2(t) * c_{12}$$

(1)

$$w_2(t) = y_2(t) * c_{22} + y_1(t) * c_{21}$$

Nuestro objetivo es que las señales emitidas sean las mismas que las señales que se captan en los oídos. Es decir, que $w_1(t) = y_1(t)$ y que $w_2(t) = y_2(t)$. Por lo tanto, debemos eliminar o al menos minimizar los términos de crosstalk que producen los caminos secundarios además de la influencia de los caminos directos. Una de las formas de minimizar todos estos efectos es prefiltrar adecuadamente las señales a emitir, invirtiendo el efecto de los caminos acústicos a través de un cancelador de crosstalk.

II.2 . CANCELADOR DE CROSSTALK

El diagrama de bloques de un cancelador de crosstalk para un sistema de reproducción estéreo de dos altavoces y dos micrófonos se representa en la siguiente figura:

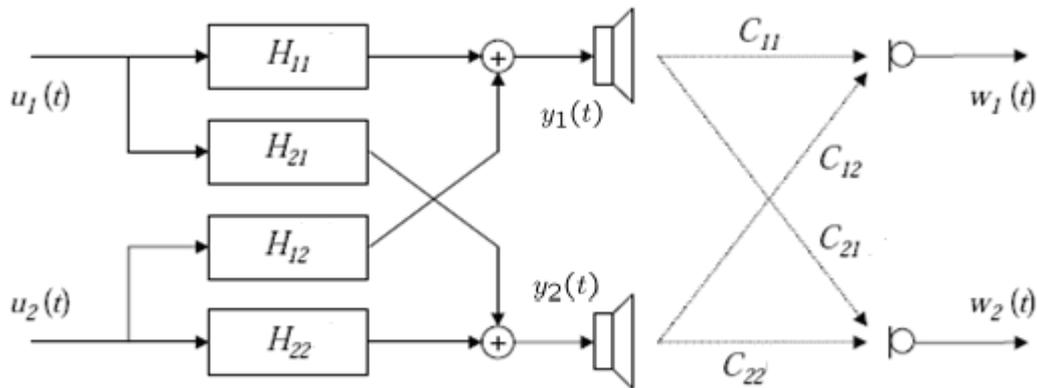


Fig.3. Cancelador de crosstalk estéreo

Los micrófonos representan los oídos del oyente. De esta manera, en un sistema real podemos captar las señales emitidas y compararlas con las señales deseadas.

Para invertir el efecto de los caminos acústicos, utilizamos el banco de filtros H que representa el cancelador de crosstalk. Las señales $y_1(t)$ y $y_2(t)$ son las que debe emitir cada altavoz para que las señales que alcanzan los oídos del oyente $w_1(t)$ y $w_2(t)$, sean idénticas a $u_1(t)$ y $u_2(t)$, las señales binaurales originales. Además, el esquema incluye los canales acústicos, que son los que hay que compensar.

Por lo tanto, debemos encontrar un banco de filtros H que minimice el error que se produce entre la señal original y la señal que recibimos:

$$e_i(t) = u_i(t) - w_i(t) \quad \text{con } i = \{1, 2\} \quad (2)$$

Este sistema se diseña en el dominio discreto y para que sea implementable en tiempo real, debe ser causal y estable. Por lo tanto, utilizaremos filtros FIR ya que son no recursivos y al no tener realimentaciones, son siempre estables. El problema de estos filtros es que para conseguir una buena resolución a bajas frecuencias se requieren muchos coeficientes, lo que produce un retardo grande y por consiguiente, a veces no son aptos para aplicaciones de tiempo real.

Si intentamos resolver el sistema para c_{11} , c_{12} , c_{21} y c_{22} conocidos no se obtienen soluciones precisas puesto que los sistemas acústicos son sistemas con retardo de fase grande a altas frecuencias (sistemas de fase no mínima) y sus inversos pueden no ser causales. Por lo que hay que añadir un retardo resultado de convolucionar los filtros C de los caminos acústicos con los filtros H que estamos diseñando. Si estos filtros fuesen causales bastaría con introducir el retardo de los caminos acústicos pero como los inversos de estos caminos no pueden serlo, para que puedan implementarse, deben retardarse y es este retardo el que hay que considerar. En una primera aproximación suele funcionar el considerar el retardo como la mitad del número de muestras que se elijan para los filtros que estamos diseñando.

Este problema puede resolverse de forma estática o adaptativa siendo la primera opción más fácil ya que sólo debemos invertir de alguna manera el efecto de los caminos acústicos y tener en cuenta algunos retardos. En la práctica, se suele utilizar el dominio temporal ya que en frecuencia se pueden encontrar sistemas inestables. Aún así, de forma teórica, vamos a calcular estos filtros inversos en ambos dominios.

II.2.1 *SOLUCIONES EN DOMINIO FRECUENCIAL*

La siguiente figura describe el modelo discreto equivalente de un sistema de reproducción de audio estéreo con preprocesado digital:

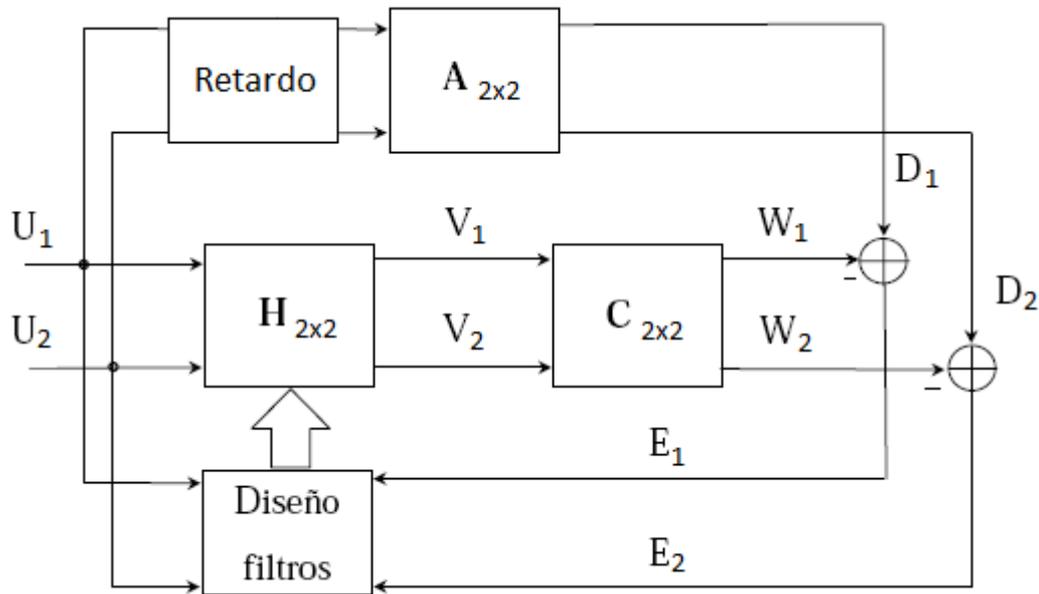


Fig.4. Diagrama de bloques simplificado de un sistema de reproducción estéreo.

Una forma útil y sencilla de ver las relaciones entre las señales es utilizar la forma matricial. Tenemos dos señales de entrada representadas por el vector \mathbf{u} , dos señales a emitir en el vector \mathbf{v} , dos señales captadas en los puntos de interés en el vector \mathbf{w} , así como 2 señales objetivo (las que se desearía obtener y que coinciden con las de entrada con un cierto retardo) en el vector \mathbf{d} . La diferencia entre las señales deseadas y captadas, forma el vector \mathbf{e} de dos señales de error.

Debemos compensar el efecto de los canales acústicos representados por la matriz \mathbf{C} mediante una matriz de filtros \mathbf{H} para conseguir una matriz deseada \mathbf{A} . Al ser un sistema estéreo, la matriz de caminos acústicos es 2×2 con lo que la matriz de los filtros que vamos a diseñar y la matriz de respuestas deseadas también lo serán.

Por ejemplo, las ecuaciones del sistema para el dominio frecuencial serán:

$$\text{-Para las señales deseadas: } \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} = \text{Retardo} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \quad (3)$$

$$\text{-Para las señales de control: } \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \quad (4)$$

$$\text{-Para las señales captadas: } \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad (5)$$

$$\text{-Para las señales de error: } \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} - \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} \quad (6)$$

La solución pasa por encontrar un banco de filtros \mathbf{H} que minimice el error que se produce entre la señal deseada y la señal que recibimos. Esta solución podría aproximarse mediante un banco de filtros FIR si se añade un retardo Δ a las respuestas deseadas.

$$\text{Consideremos el vector de error: } E_i(z) = D_i(z) - W_i(z) \quad \text{con } i,j=\{1, 2\} \quad (7)$$

$$\text{Como } D_i(z) = z^{-\Delta} A_{ij}(z)U_i(z) \text{ y } W_i(z) = C_{ij}(z)V_i(z) \text{ siendo } V_i(z) = H_{ij}(z)U_i(z) \quad (8)$$

$$\text{obtenemos que: } E_i(z) = [z^{-\Delta} A_{ij}(z) - C_{ij}(z)H_{ij}(z)] U_i(z) \quad (9)$$

Definimos la función de coste J como la esperanza matemática del error cuadrático:

$$J_i(z) = E [|E_i(z)|^2] = E [|D_i(z) - W_i(z)|^2] \quad (10)$$

Por lo tanto, debemos hallar los filtros H que minimicen esta función de coste $J_i(z)$.

$$\text{El error será mínimo cuando: } C_{ij}(z)H_{ij}(z) \rightarrow z^{-\Delta} A_{ij}(z) \quad (11)$$

$$\text{Por lo tanto: } H_{ij}(z) = z^{-\Delta} C_{ij}^{-1}(z) A_{ij}(z) \quad (12)$$

Cada $H_{ij}(z)$ tendrá la información de todos los caminos acústicos para cada punto frecuencial. Como la matriz \mathbf{C} es cuadrada, podemos hallar su inversa mediante inversión matricial. Si no lo fuera, deberíamos hallar la matriz generalizada o pseudoinversa y añadirle un término de regularización para producir una inversa más estable.

Como buscamos la cancelación ideal, la respuesta deseada \mathbf{A} será equivalente a la unidad entre cada señal y su oído correspondiente y cero en los opuestos. Es decir, a la matriz identidad:

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (13)$$

Teniendo en cuenta estas consideraciones e igualando la respuesta del sistema a la respuesta deseada, los filtros han de cumplir las siguientes condiciones:

$$\begin{array}{l}
 H_{11}C_{11}+H_{21}C_{12}= A_{11} \\
 H_{11}C_{21}+H_{21}C_{22}= A_{12} \\
 H_{22}C_{12}+H_{12}C_{11}= A_{21} \\
 H_{22}C_{22}+H_{12}C_{21}= A_{22}
 \end{array}
 \quad \longrightarrow \quad
 \begin{array}{l}
 H_{11}C_{11}+H_{21}C_{12}= z^{-\Delta} \\
 H_{11}C_{21}+H_{21}C_{22}=0 \\
 H_{22}C_{12}+H_{12}C_{11}=0 \\
 H_{22}C_{22}+H_{12}C_{21}= z^{-\Delta}
 \end{array}
 \quad (14)$$

II.2.2 SOLUCIONES EN DOMINIO TEMPORAL

Para el dominio temporal, la ecuaciones serán las mismas que (3)(4)(5)(6), cambiando la operación multiplicación por convolución. La ventaja de elegir esta estrategia es que podemos considerar del banco de filtros inversos H y el conjunto de canales acústicos como filtros FIR causales.

Siguiendo el mismo criterio que en el dominio frecuencial, definimos una función de coste $J_i(n)$ que debemos minimizar:

$$J_i(n) = E[|e_i(n)|^2] = E[|d_i(n)-w_i(n)|^2] \quad (15)$$

$$\text{El error será mínimo cuando: } c_{ij}(n) * h_{ij}(n) \rightarrow \delta(n-\Delta) * a_{ij}(n) \quad (16)$$

En este caso, la respuesta deseada A será equivalente al impulso unidad retardado entre cada señal y su oído correspondiente y cero en los opuestos:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} \delta(n-\Delta) & 0 \\ 0 & \delta(n-\Delta) \end{bmatrix} \quad (17)$$

Los filtros han de cumplir las mismas condiciones que en la ecuación (14) cambiando solamente la multiplicación por la convolución y tomando el retardo como una delta desplazada:

$$\begin{array}{l}
 h_{11}*c_{11}+h_{21}*c_{12}= a_{11} \\
 h_{11}*c_{21}+h_{21}*c_{22}= a_{12} \\
 h_{22}*c_{12}+h_{12}*c_{11}= a_{21} \\
 h_{22}*c_{22}+h_{12}*c_{21}= a_{22}
 \end{array}
 \quad \longrightarrow \quad
 \begin{array}{l}
 h_{11}*c_{11}+h_{21}*c_{12}= \delta(n-\Delta) \\
 h_{11}*c_{21}+h_{21}*c_{22}=0 \\
 h_{22}*c_{12}+h_{12}*c_{11}=0 \\
 h_{22}*c_{22}+h_{12}*c_{21}= \delta(n-\Delta)
 \end{array}
 \quad (18)$$

Este sistema de ecuaciones expresando la convolución como un producto de matrices queda como:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \quad (19)$$

siendo c_{11} , c_{12} , c_{21} y c_{22} matrices con estructura de matriz Toeplitz no simétrica, con una considerable cantidad de términos a cero de la forma:

$$\begin{bmatrix} c_{ij}(0) & & & & 0 \\ c_{ij}(1) & c_{ij}(0) & & & \\ \vdots & c_{ij}(1) & \ddots & & \\ c_{ij}(M) & \vdots & & c_{ij}(0) & \\ & c_{ij}(M) & & c_{ij}(1) & \\ & & & \vdots & \\ 0 & & & & c_{ij}(M) \end{bmatrix} \quad (20)$$

La matriz C no puede ser cuadrada ya que siempre tendremos más ecuaciones que incógnitas. Si L es la longitud de los filtros h , a diseñar y M es la longitud de los caminos acústicos, el número de filas de C será $2(L+M-1)$ mientras que su número de columnas será $2L$.

Las desventajas de estos sistemas surgen en el caso en que se desplacen los sensores o cuando las señales deseadas cambien dinámicamente. En estos casos, el sistema pierde eficiencia puesto que los filtros están calculados para unos puntos en concreto.

II.3. FILTRADO ADAPTATIVO

Los sistemas adaptativos son otra solución posible para resolver el problema de encontrar el banco de filtros óptimos para un sistema de reproducción estéreo. Son útiles cuando a priori no se sabe qué filtro utilizar o cuando el filtro óptimo deba variar con el tiempo (sistemas no estacionarios). Cuando los sistemas de adaptan muestra a muestra se suele usar el dominio temporal para este tipo de filtrado.

En este caso, los filtros digitales adaptan continuamente sus coeficientes dependiendo de una determinada función de coste que defina un algoritmo que los actualiza. Los coeficientes varían con el tiempo, se adaptan en función del entorno cambiante y pueden ser entrenados. No hay métodos de síntesis, solamente hay que diseñar la regla de variación de los coeficientes.

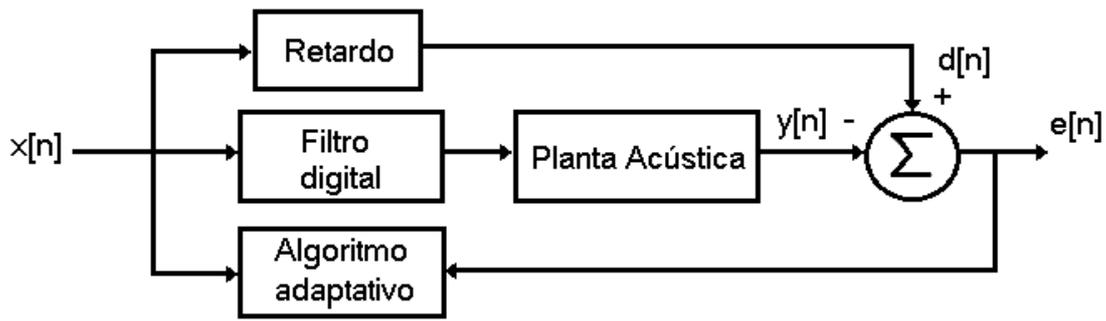


Fig.5. Diagrama de bloques simplificado de un sistema de reproducción transaural usando un algoritmo adaptativo.

Según vemos en la figura anterior $x(n)$ es la señal de entrada o de referencia, $d(n)$ es la señal deseada, $y(n)$ es la señal generada y $e(n)$ la señal de error.

El objetivo sigue siendo el mismo, diseñar el filtro FIR digital H de manera que la señal de error $e[n]$ sea lo menor posible. Para ello se busca que la señal $y[n]$ se parezca a la señal de referencia $d[n]$ ya que el error es la diferencia entre ambas:

$$e(n) = d(n) - y(n) \quad (21)$$

Los coeficientes de H deben ajustarse para que la salida del sistema $y[n]$ sea una aproximación por mínimos cuadrados de la señal deseada $d[n]$. Para ello se minimiza la potencia media del error cometido:

$$\text{MSE} = E[|e(n)|^2] = E[|d(n) - y(n)|^2] \quad (22)$$

Los algoritmos adaptativos intentan encontrar el mínimo mediante técnicas iterativas. Se parte de un vector de coeficientes inicial $\mathbf{h}(0)$ que provoca un cierto exceso de error ΔMSE , y a continuación, utilizando un criterio adecuado, se calcula un nuevo vector $\mathbf{h}(1)$ que dé lugar a un ΔMSE menor. Siguiendo este procedimiento se puede llegar al vector de pesos óptimo.

II.3.2 El algoritmo LMS

Este algoritmo iterativo se basa en el gradiente para calcular el mínimo de MSE. Se desplaza por la función de error según la dirección de máxima variación, es decir la inversa de la dirección del gradiente, o mejor dicho una estima de ésta.

En la siguiente figura se muestra la función MSE en una dimensión (parábola):

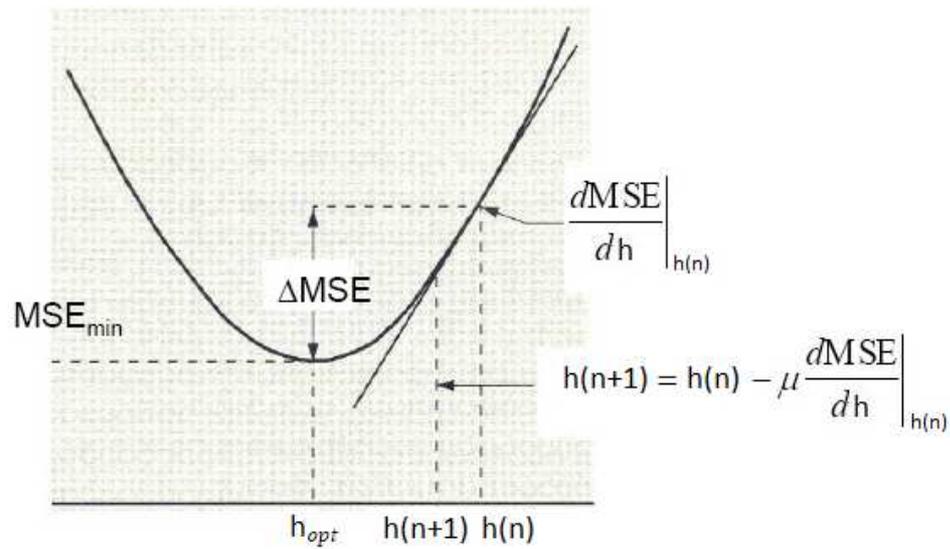


Fig.6. Función MSE (error cuadrático medio)

Se parte de $h(n)$ y para mejorarlo se debe ir en dirección a h_{opt} usando la información de la derivada. El gradiente indica la dirección de máxima pendiente y su módulo es el valor de la pendiente. Por tanto el mínimo que buscamos tendrá pendiente cero, es decir, su gradiente será cero.

$$\text{Gradiente: } \nabla_h = \frac{dMSE}{dh} = 0 \tag{23}$$

El algoritmo LMS (Least Mean Squares) usa el valor instantáneo del error cuadrático como aproximación del error cuadrático medio, en el cálculo de la estimación del gradiente:

$$\mathbf{g}(n) = \nabla_h \{ [e^2(n)] \} = 2e(n) \nabla_h \{ e(n) \} \tag{24}$$

Como sabemos que: $e(n) = d(n) - y(n)$ $y(n) = \mathbf{h}^T \cdot \mathbf{x}(n)$ (25)

sustituimos en la ecuación (24) y nos queda: $\mathbf{g}(n) = 2e(n) \nabla_h \{ d(n) - \mathbf{h}^T \cdot \mathbf{x}(n) \}$ (26)

Como el gradiente sólo depende de los coeficientes h : $= -2e(n) \nabla_h \{ \mathbf{h}^T \cdot \mathbf{x}(n) \}$ (27)

Y como hemos dicho que el mínimo que buscamos tendrá gradiente igual a cero, obtenemos que:

$$\mathbf{g}(n) = -2e(n) \mathbf{x}(n) \tag{28}$$

A este término $\mathbf{g}(n)$ se le llama gradiente instantáneo pues se basa en los valores de e y \mathbf{x} en el instante n . Si se calcula su media se obtiene el gradiente verdadero

Por tanto, la ecuación de actualización de los coeficientes, base del algoritmo LMS, queda de la siguiente manera:

$$\begin{aligned}
 h(n+1) &= h(n) - \mu \left. \frac{dMSE}{dh} \right|_{h(n)} \\
 &\Downarrow \\
 h(n+1) &= h(n) - \mu \nabla_h \{ [e^2(n)] \} \\
 &\Downarrow \\
 h(n+1) &= h(n) - \mu \mathbf{g}(n) \\
 &\Downarrow \\
 \boxed{h(n+1) = h(n) + 2\mu e(n) \mathbf{x}(n)} & \quad (29)
 \end{aligned}$$

donde μ es una constante denominada parámetro de convergencia que debe estar acotada superiormente de forma conveniente para evitar la inestabilidad del algoritmo:

$$0 < \mu < \frac{1}{(N+1)P_x} \quad (30)$$

donde P_x es la potencia de la señal $x(n)$ y N el orden del filtro.

A modo de resumen, los pasos que describen este algoritmo son:

- 1) Se actualiza el vector de muestras de entrada:

$$\mathbf{x}^T(n) = [x(n) \ x(n-1) \ \dots \ x(n-N)] \quad (31)$$

- 2) Se calcula la salida del filtro transversal:

$$y(n) = \mathbf{h}^T \cdot \mathbf{x}(n) = \mathbf{x}^T(n) \cdot \mathbf{h} \quad (32)$$

- 3) Se calcula la señal de error para ese instante:

$$e(n) = d(n) - y(n) \quad (33)$$

- 4) Se actualizan los coeficientes del filtro transversal:

$$h(n+1) = h(n) + 2\mu e(n) \mathbf{x}(n) \quad (34)$$

III.3.2 El algoritmo de filtrado x-LMS

Este algoritmo, basándose en las mismas técnicas que el anterior, corrige el retardo que existe cuando la señal que se genera no se corresponde con la que se usa en el cálculo de la señal de error.

Para ello, utiliza una estima de los caminos acústicos para actualizar los coeficientes del filtro adaptativo, tal y como se describe en [1]:

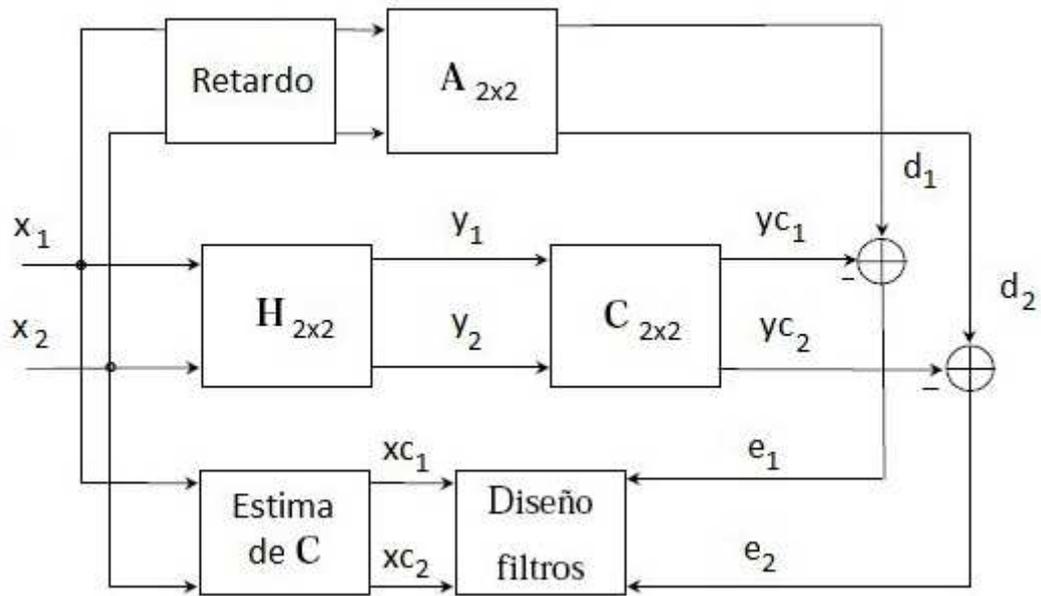


Fig.7. Sistema de reproducción estéreo adaptativo usando el algoritmo de filtrado x-LMS.

Así, las señales xc_1 y xc_2 que utiliza el algoritmo adaptativo y las señales yc_1 e yc_2 a partir de la que se obtiene las señales de error, están relacionadas adecuadamente.

Siendo L la longitud de los filtros H adaptativos, M la longitud de los filtros C que describen los caminos acústicos, I el número de señales de entrada (en nuestro caso, $I=2$), J el número de altavoces ($J=2$) y K el número de micrófonos ($K=2$), entonces:

-definimos el vector de coeficientes \mathbf{h} como:

$$\mathbf{h}_{ij}(n) = [h_{ij1}(n) \quad h_{ij2}(n) \quad \dots \quad h_{ijL}(n)]^T \tag{35}$$

-definimos el vector de coeficientes \mathbf{c} como:

$$\mathbf{c}_{jk} = [c_{jk1} \quad c_{jk2} \quad \dots \quad c_{jk1M}]^T \tag{36}$$

-actualizamos las señales que actúan en el sistema con una determinada longitud:

$$\begin{aligned}
 \mathbf{x}c_{ijk}(n) &= [xc_{ijk}(n) \quad xc_{ijk}(n-1) \quad \dots \quad xc_{ijk}(n-L+1)]^T \\
 \mathbf{x}_i(n) &= [x_i(n) \quad x_i(n-1) \quad \dots \quad x_i(n-L+1)]^T \\
 \mathbf{x}'_i(n) &= [x_i(n) \quad x_i(n-1) \quad \dots \quad x_i(n-M+1)]^T \\
 \mathbf{y}_j(n) &= [y_j(n) \quad y_j(n-1) \quad \dots \quad y_j(n-M+1)]^T
 \end{aligned} \tag{37}$$

Entonces, el filtrado XLMS puede ser descrito por las siguientes ecuaciones:

$$y_j(n) = \sum_{k=0}^I \mathbf{h}_{ij}^T(n) \mathbf{x}_i(n) \quad \mathbf{x}c_{ijk}(n) = \mathbf{c}_{jk}^T(n) \mathbf{x}'_i(n) \tag{38}$$

donde la ecuación de actualización del algoritmo es:

$$\mathbf{h}_{ij}(n+1) = \mathbf{h}_{ij}(n) + \mu \sum_{k=1}^K \mathbf{x}c_{ijk}(n) e_k(n) \tag{39}$$

mientras que la señal error será:

$$e_k(n) = d_k(n) - \sum_{j=1}^J \mathbf{c}_{jk}^T(n) \mathbf{y}_j(n) \tag{40}$$

III. IMPLEMENTACIÓN PRÁCTICA

El objetivo final que se pretende alcanzar con estas técnicas es un sistema de reproducción multicanal robusto y escalable en número de fuentes y sensores (siempre dentro de las capacidades máximas de cálculo del hardware utilizado). Hay que tener claro que estamos intentando programar un sistema simulado con el fin de poder implementarse en un sistema real en un futuro. Para todas las simulaciones, usaremos Matlab como entorno de programación.

Como vamos a partir de caminos acústicos conocidos, para calcular las respuestas al impulso entre los altavoces y los micrófonos, utilizaremos una herramienta simple y de libre distribución: ROOMSIM.

Antes de comenzar, hay que tener en cuenta que para, por ejemplo, un sistema real de reproducción transaural, tenemos dos señales de entrada que simulan una señal estéreo, un sistema de procesado donde se realizan las operaciones necesarias para conseguir la cancelación de crosstalk, dos altavoces por donde se emiten dichas señales procesadas y un oyente que las escucha a través del espacio acústico que lo separa de los altavoces. Para poder tener acceso a las señales de escucha, se suponen dos micrófonos cercanos a los oídos del oyente y que captan dichas señales. Pero en un sistema simulado, las señales que recogen los micrófonos las tenemos que calcular a partir de las señales que emitirían los altavoces filtrándolas por los caminos acústicos. Esa parte que en los sistemas reales ya está implícitamente, en los sistemas de simulación hay que implementarla.

Por tanto, como idea principal básica, hemos de diseñar unos filtros inversos de tal manera que las señales de entrada filtradas por estos filtros generen unas nuevas señales que, tras ser filtradas por los caminos acústicos, den como resultado las señales originales.

Conviene tener en cuenta que el resultado que obtendremos no serán las mismas señales originales sino una solución que minimiza la función de coste elegida (que para el caso del algoritmo LMS es la potencia instantánea de la señal de error). En el caso de los sistemas que calculan de forma estática el prefiltrado, puesto que el sistema de ecuaciones es de rango incompleto (infradeterminado), no tendrá solución exacta y por tanto la solución escogida es la que minimiza el error cuadrático medio.

Teniendo clara esta idea y como ya hemos comentado con anterioridad, empezaremos implementando el caso más sencillo, el cancelador de crosstalk estático. Veremos su problemática y eso nos servirá para avanzar hasta alcanzar el objetivo final.

III.1. SIMULADOR DE SISTEMA DE REPRODUCCIÓN TRANSAURAL ESTÁTICO

Recordando la figura 3, que representa el cancelador de crosstalk estático, vemos que los filtros a diseñar son h_{11} , h_{12} , h_{21} y h_{22} . Se pueden calcular tanto en tiempo como en frecuencia. Nosotros hemos optado por el dominio temporal puesto que en frecuencia se pueden encontrar sistemas inestables mientras que en tiempo, a lo sumo, encontraremos inversas un poco imprecisas.

Partimos de las respuestas al impulso de los caminos acústicos, es decir, conocemos c_{11} , c_{12} , c_{21} y c_{22} . y también conocemos las señales de entrada. De primeras, utilizaremos dos señales de banda ancha como ruido blanco. Por otro lado, si queremos comprobar que el algoritmo funciona escuchando las señales captadas por los micrófonos, no es recomendable usar señales de ruido porque son difíciles de diferenciar una de otra. Es mejor usar dos tonos de frecuencias diferentes, o dos archivos de sonido con contenido claramente diferenciado para comprobar si se produce crosstalk de una manera clara. De todos modos, nuestro sistema ha de diseñarse para que sea cual sea la señal de entrada, consigamos una cancelación de crosstalk aceptable entre los dos canales de reproducción.

Primero tenemos que determinar los coeficientes que van a tener los filtros y el retardo asociado. Hay que recordar que introducimos este retardo para que el sistema inverso sea estable y causal. Cuantos más coeficientes, más grados de libertad para poder resolver las ecuaciones de cada rama. Primero se suele estimar un número doble o mayor de los coeficientes que tienen los caminos acústicos que queremos invertir y si luego vemos que la solución alcanzada tiene muchos valores relativamente pequeños por los extremos, se desprecian dichos coeficientes, truncando la respuesta al impulso para obtener respuestas lo más cortas posibles.

Aunque al reducir el número de coeficientes se ajusta peor la solución (ya que los filtros en teoría tienen una respuesta de duración infinita), nos interesa que sean cortos para que los cálculos a realizar no saturen la capacidad del procesador usado y por eso se desprecian aquellos coeficientes que cuyo valor esté próximo a cero puesto que añadirlos no mejorará la respuesta del filtrado ya que también se añaden errores de imprecisión numérica, y sí que añadirán coste computacional.

En nuestro caso, los caminos acústicos que utilizamos tienen 350 coeficientes ($M=350$) por lo que la longitud de los filtros H será: $N=2M+1$ o mejor, la potencia de 2 más cercana por arriba: $N=2 \times 350 + 1 = 700 \rightarrow 1024$. Por lo tanto, los filtros inversos a diseñar tendrán 1024 coeficientes.

Una primera aproximación, que suele funcionar, es considerar el retardo Δ como la mitad del número de muestras de los filtros $H \rightarrow \Delta = 512$ muestras. En nuestro caso, adelantamos que el retardo elegido es igual a la mitad de $N+M-1$ muestras por razones que veremos más adelante.

Los filtros han de cumplir el sistema de ecuaciones (18) y para resolverlo, expresaremos cada ecuación utilizando la forma matricial de la forma expuesta en (19) que podemos agrupar de la manera:

$$\mathbf{a} = \mathbf{c} \mathbf{h}$$

La matriz \mathbf{c} está formada por los coeficientes de los caminos acústicos estratégicamente colocados. Tiene estructura de matriz de bloques de Toeplitz no simétrica con términos a cero como vimos en (20). Como N es la longitud de los filtros h y M es la longitud de los caminos acústicos, el número de filas de \mathbf{c} será $2(N+M-1)$ mientras que su número de columnas será $2N$. Podemos obtener esta matriz mediante sencillas operaciones matriciales en Matlab. La matriz \mathbf{h} tendrá dimensión $2N$ filas por 2 columnas y el vector \mathbf{a} $2(N+M-1)$ filas por 2 columnas.

En la práctica, las respuestas deseadas a_{11} y a_{12} se modelan como un retardo $\delta(n - \Delta)$. En forma matricial, esto se traduce en un vector de longitud $M+N-1$ formado por ceros y un uno en la muestra Δ , que será la mitad de la longitud del vector. Por su parte, las respuestas deseadas para a_{21} y a_{22} serán vectores de ceros de longitud $M+N-1$.

Teniendo en cuenta todos estos aspectos, la ecuación $\mathbf{a}=\mathbf{c}\mathbf{h}$ tendrá la forma:

$$\begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ 1 & \vdots \\ \vdots & \vdots \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ \vdots & \vdots \\ \vdots & 1 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} = \begin{matrix} \text{2(N+M-1)} \\ \text{filas} \end{matrix} \begin{bmatrix} c_{11}(0) & & 0 & & c_{12}(0) & & 0 \\ c_{11}(1) & c_{11}(0) & & & c_{12}(1) & c_{12}(0) & & & 0 \\ \vdots & c_{11}(1) & \ddots & & \vdots & c_{12}(1) & \ddots & & \vdots \\ c_{11}(M) & \vdots & & c_{11}(0) & c_{12}(M) & \vdots & & c_{12}(0) \\ & c_{11}(M) & & c_{11}(1) & & c_{12}(M) & & c_{12}(1) \\ & & & \vdots & & & & \vdots \\ 0 & & & c_{11}(M) & 0 & & & c_{12}(M) \\ c_{21}(0) & & & 0 & c_{22}(0) & & & 0 \\ c_{21}(1) & c_{21}(0) & & & c_{22}(1) & c_{22}(0) & & & \\ \vdots & c_{21}(1) & \ddots & & \vdots & c_{22}(1) & \ddots & & \vdots \\ c_{21}(M) & \vdots & & c_{21}(0) & c_{22}(M) & \vdots & & c_{22}(0) \\ & c_{21}(M) & & c_{21}(1) & & c_{22}(M) & & c_{22}(1) \\ & & & \vdots & & & & \vdots \\ 0 & & & c_{21}(M) & 0 & & & c_{22}(M) \end{bmatrix} \begin{matrix} \text{2N} \\ \text{columnas} \end{matrix} \begin{bmatrix} h_{11}(0) & h_{12}(0) \\ h_{11}(1) & h_{12}(1) \\ \vdots & \vdots \\ h_{11}(N) & h_{12}(N) \\ h_{21}(0) & h_{22}(0) \\ h_{21}(1) & h_{22}(1) \\ \vdots & \vdots \\ h_{21}(N) & h_{22}(N) \end{bmatrix} \begin{matrix} \text{2N} \\ \text{filas} \end{matrix}$$

Fig.8. Representación matricial de la ecuación $\mathbf{a}=\mathbf{c}\mathbf{h}$.

A continuación, debemos despejar \mathbf{h} . Como es un sistema infradeterminado, donde tenemos más ecuaciones que incógnitas, no admite solución exacta sino que hay que usar algún método de mínimos cuadrados o similar. Matlab lo resuelve con la instrucción $\mathbf{h}=\mathbf{c}\backslash\mathbf{a}$ que realiza la multiplicación de la inversa de \mathbf{c} con \mathbf{a} por la izquierda. Es equivalente a $\mathbf{c}^{-1}\mathbf{a}$ si existe la inversa de \mathbf{c} . Si la matriz \mathbf{c} no es cuadrada, automáticamente se realiza el cálculo de su pseudoinversa, lo que equivaldría a resolver el sistema de ecuaciones correspondiente por mínimos cuadrados.

Para comprobar que los filtros inversos calculados son los correctos, se tienen que cumplir que la convolución entre estos filtros y los caminos acústicos nos den una delta desplazada o un cero, según las ecuaciones (18). Implementado el código según los pasos comentados, creamos la función:

```
>>XTC(N,u1,u2);
```

donde N es la longitud de los filtros inversos y $u1$ y $u2$ son las señales de entrada.

En las siguientes figuras, vemos que se cumplen aceptablemente y por tanto, los filtros H están hallados correctamente:

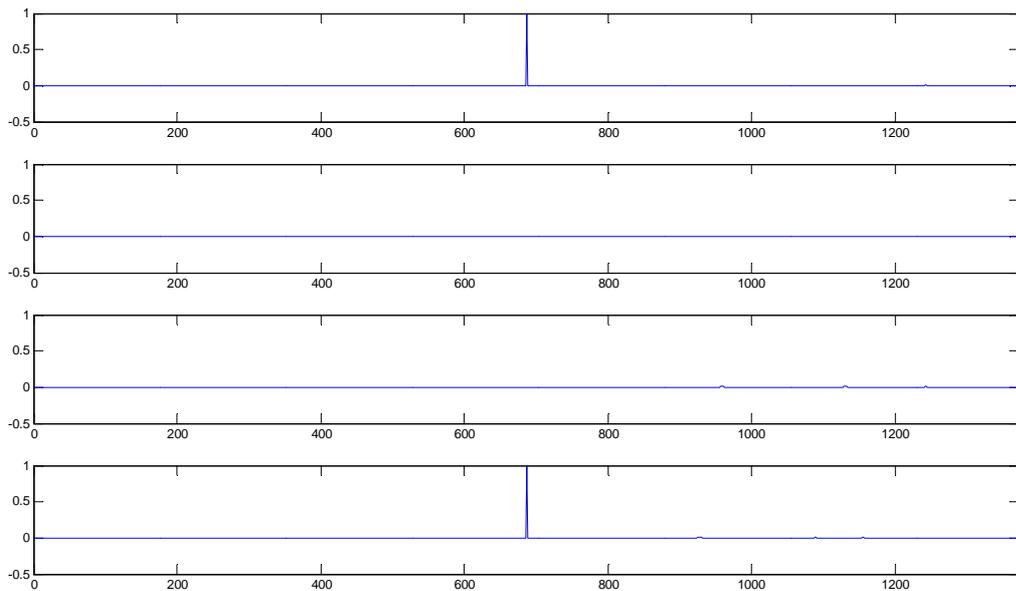


Fig.9. Señales resultantes de la convolución entre los filtros inversos y los caminos acústicos.

Una vez tenemos los filtros inversos, los introducimos en el sistema y en el apartado de análisis de resultados, observaremos que el error que se produce entre las señales deseadas y las señales captadas es mínimo. El problema de este método es que es estático, es decir, que no se adapta a un entorno acústico variable, como por ejemplo, cambios de los caminos acústicos debidos a que el oyente se está moviendo o si queremos cambiar la información que llega a cada micrófono. De ahí la necesidad de utilizar algoritmos adaptativos.

III.2. SIMULADOR DE SISTEMA DE REPRODUCCIÓN TRANSAURAL ADAPTATIVO

Una vez realizado el sistema de reproducción transaural calculando los filtros inversos de forma estática, ahora vamos a realizarlo de forma adaptativa. Para ello, utilizaremos un tipo de algoritmo adaptativo conocido como algoritmo de filtrado x -LMS que ya hemos explicado en el apartado anterior. La razón por la que hemos elegido este tipo de algoritmo adaptativo es porque es fácil de implementar, se consigue una adaptación aceptable y es uno de los más utilizados en sistemas TSR [1].

El objetivo de este tipo de algoritmo es minimizar las señales de error calculadas a partir de la señal que se recibe en cada uno de los micrófonos menos la que debiera recibirse (tal y como vimos tanto en la figura 5 como en la ecuación (21) con la particularidad que utiliza técnicas iterativas para conseguirlo.

Como queremos implementar un sistema de reproducción transaural estéreo adaptativo, debemos fijarnos en el diagrama de bloques de la figura 7 así como en las ecuaciones que se derivan, tal y como vimos en el apartado anterior y según la referencia [1]. Por lo tanto, tendremos 2 señales de entrada, 2 altavoces y 2 micrófonos.

Ahora lo que debemos hacer es traducir ese diagrama de bloques con sus pertinentes ecuaciones a código de nuestro entorno de programación para poder obtener el simulador.

Para ello, definiremos una función en Matlab llamada *filtrado_xLMS(u1,u2,L,mu)* cuyos parámetros de entrada son:

- $u1$ y $u2$ = señales de entrada,
- L = longitud en muestras de los filtros adaptativos,
- μ = constante llamada paso de convergencia (μ).

Es importante introducir un valor de N adecuado ya que si elegimos un valor pequeño es posible que no de tiempo a que los filtros se adapten puesto que la idea es utilizar este valor como límite superior en el bucle de la iteración.

El sistema es bastante crítico con el número de coeficientes L y con el paso de convergencia elegidos, por lo que hay que ir probando mediante prueba y error. Los 4 caminos acústicos los hemos calculado gracias a la herramienta ROOMSIM. Tienen una longitud de 350 muestras y se han guardado en la variable *stereo1.mat*.

Respecto a las señales de entrada, elegir un tipo de señal u otra condiciona la elección de otros parámetros (μ por ejemplo). Para conseguir la adaptación de los filtros en toda la banda de trabajo hay que usar señales de banda ancha como ruido blanco. La duración de las mismas (N) debe ser lo suficiente para alcanzar el estado estacionario del sistema y por tanto, para obtener los coeficientes de dichos filtros que consiguen minimizar la función de coste elegida.

Como vemos en la figura 7, debemos retrasar las señales de entrada un determinado número de muestras para poder calcular las señales de error correctamente. A estas señales las llamaremos deseadas y los retardos elegidos han de coincidir con los retardos de los caminos acústicos, que en nuestro caso han sido estimados a partir de la muestra donde incide la propagación directa (determinada por el valor máximo de la respuesta al impulso). En caso de que este retardo no estuviera bien estimado, podemos retrasar o adelantar una o dos muestras estos retardos mediante prueba y error.

El vector de coeficientes iniciales de los cuatro filtros adaptativos lo inicializamos a ceros. Utilizamos buffers circulares y definimos todas las variables de forma matricial/vectorial para ahorrarnos la creación dinámica de variables.

Como ya hemos explicado, este algoritmo trabaja muestra a muestra y para que los coeficientes se vayan adaptando a sus valores óptimos y conseguir minimizar el error, necesitamos técnicas iterativas. Por lo que utilizaremos un bucle *for* para recorrer N muestras que es la duración de las señales de entrada. Este método considera la estima de los caminos acústicos en su ecuación de actualización de los filtros adaptativos. En nuestro caso, vamos a utilizar una estimación perfecta puesto que dichas estimas son iguales a nuestros caminos acústicos.

Los coeficientes se hallan adaptativamente y se actualizan en cada iteración. Para lo cual, utilizamos una estima de los caminos acústicos igual a las señales de entrada retrasadas los retardos que introducen cada uno de los caminos acústicos multiplicada por las dos señales de error.

Después de implementar la parte eléctrica, nos quedaría la parte acústica, es decir, calcular las señales que recibiríamos en la posición de los micrófonos, filtrando las señales que llegan a los altavoces por los caminos acústicos.

Para poder ver la evolución del error cometido, creamos dos vectores error (uno para cada canal) que nos servirán inicialmente para comprobar si el sistema diverge o no

En la figura 10, podemos comprobar que el algoritmo no diverge y que se consigue reducir el error:

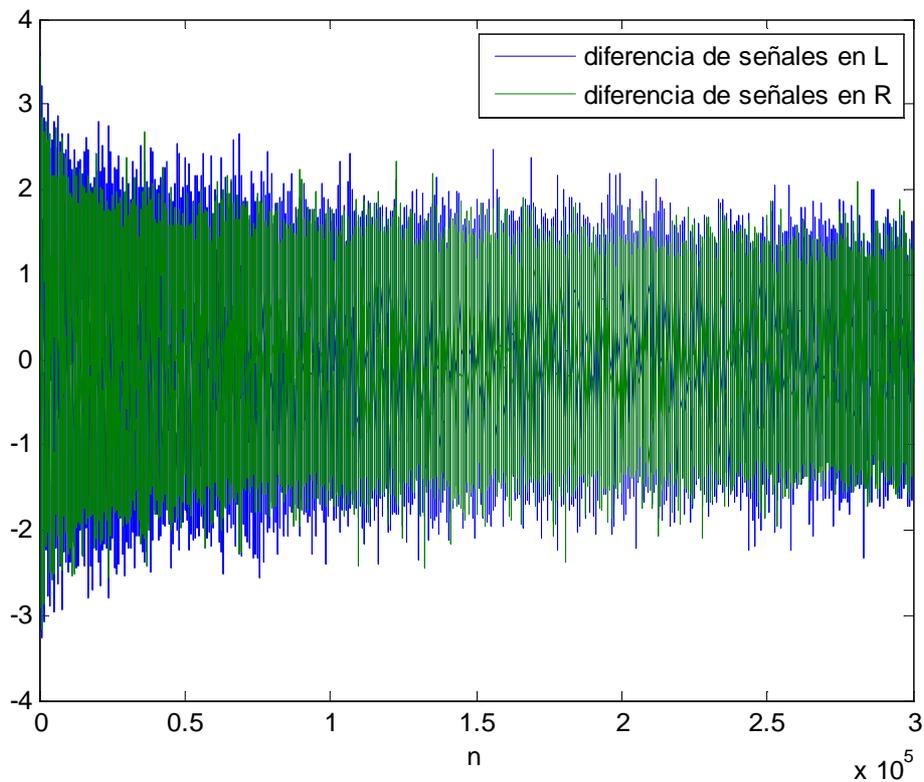


Fig.10. Señales de error resultantes del sistema de reproducción transaural estéreo con filtrado x-LMS.

III.3. SIMULADOR DE SISTEMA GENÉRICO DE REPRODUCCIÓN MULTICANAL ADAPTATIVO.

Una vez obtenido el sistema de reproducción transaural adaptativo, es decir un sistema 2:2:2 (2 señales de entrada, 2 altavoces y 2 micrófonos), ahora debemos escalar el código para conseguir un sistema genérico multicanal I:J:K (I señales de entrada, J altavoces y K micrófonos). Lo ideal es que el código soporte cualquier sistema genérico y aunque haya algunas configuraciones que no tienen sentido, el código debe funcionar con mejores o peores resultados.

Definiremos una función en Matlab llamada: *filtrado_xLMS_generico* (*entradas,altavoces,micros,L,mu*) cuyos parámetros de entrada son los mismos que en el caso transaural añadiendo la posibilidad de elegir el número de entradas, altavoces y micrófonos que se desee.

Igual que en el caso anterior, el número de coeficientes L y el paso de convergencia deben ser elegidos probando mediante prueba y error. Si bien L puede mantenerse para todas las configuraciones, μ si ha de cambiarse. El criterio escogido es que μ debe ser el mismo que se use en un sistema monocanal dividido por el número de micrófonos. En este caso, las variables estarán dimensionadas en función del número de fuentes, altavoces y micrófonos, así como tamaño de los filtros adaptativos y de las estimaciones de los caminos acústicos. Aunque en un principio se creyó conveniente cargar los caminos por pares para simular que el objetivo es siempre un oyente, para hacerlo más genérico, se decidió cargar los caminos por separado.

A la hora de almacenar los caminos, utilizaremos dos matrices:

- f_estima , para el cálculo de las estimas,
- f_acus , para el cálculo de los caminos acústicos.

La razón es que según el caso, los caminos están ordenados de manera diferente y además, se de la posibilidad en un futuro, de poder utilizar f_estima para cargar estimas que no sean perfectas, que no sean los mismos caminos acústicos.

Como novedad respecto al sistema anterior, permitiremos al usuario elegir qué información desea tener en cada sensor, es decir, lo que se desea escuchar a partir de las señales de entrada que haya o incluso combinación de ellas (suma o resta).

Por lo demás, las instrucciones son similares al caso transaural, teniendo en cuenta de que las variables se han de redimensionar dependiendo del número de entradas, altavoces y micrófonos.

La figura 11 muestra las señales de error para ver que el sistema no diverge:

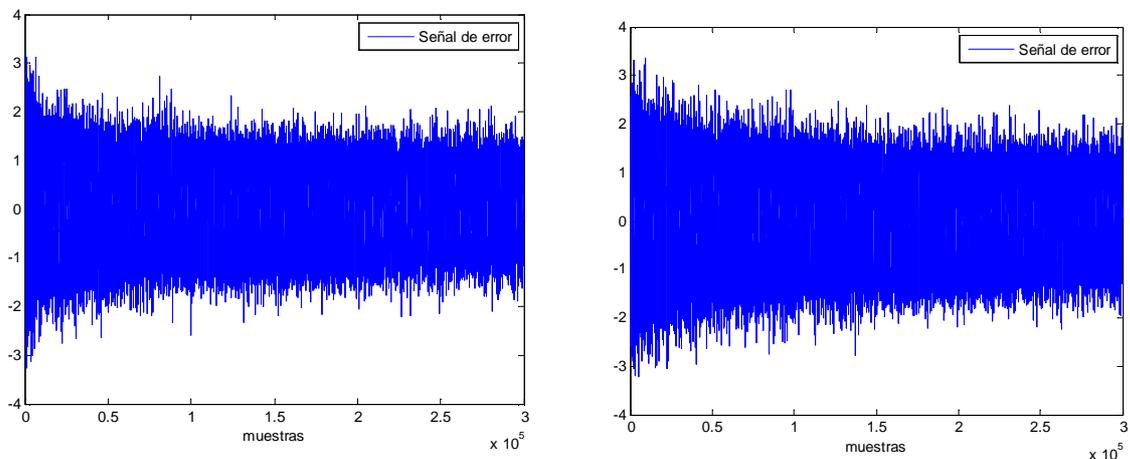


Fig.11. Señales de error del resultantes del sistema de reproducción multicanal para la configuración 2:2:2.

Como vemos, los resultados son idénticos a la figura 10 por lo que podemos afirmar que el simulador multicanal se ha implementado correctamente.

Ahora nos queda demostrar que es un sistema robusto frente a todo tipo de configuraciones (I:J:K) y sobre todo, frente a desplazamientos del oyente y cambios en la información de los sensores.

Para ello, vamos a diseñar una interfaz gráfica que nos ayude y facilite el trabajo de análisis y comparación de resultados así como de verificación de la robustez del algoritmo implementado.

III.4. CREACIÓN DEL INTERFAZ GRÁFICO

Siendo fiel al objetivo de la tesina, una vez desarrollado un sistema adaptativo gobernado por el algoritmo LMS hemos de demostrar que es un sistema de TSR multicanal robusto frente a los desplazamientos relativos de las posiciones oyente-fuente y al cambio de información en los sensores.

Para ello, vamos a diseñar una interfaz gráfica que nos ayude a realizar las simulaciones más fácilmente configurando el simulador de manera visual y así poder comparar los resultados de manera rápida y eficiente.

Como parámetros de entrada, hay que introducir en el programa:

- El número de señales de entrada y el tipo,
 - El número de altavoces y sensores (por lo tanto, quedaría definido el sistema acústico),
 - Qué información se quiere recibir en cada sensor, es decir, qué señales de entrada (o combinación de ellas) se quiere escuchar en cada punto,
 - Además, si queremos que pueda evolucionar con el tiempo, debemos ofrecer la posibilidad de que las señales que se desean escuchar cambien con el tiempo, es decir, si en el sensor 1 se está escuchando la fuente 1, que a partir de un instante determinado se escuchase la fuente 2, reajustándose los filtros adaptativos.
 - De la misma forma, debe permitir cambiar la posición de los sensores para simular que el oyente se mueve.
 - El resto de parámetros de entrada del programa serían los propios del algoritmo (número de coeficientes de los filtros, paso de convergencia, etc).
-

Una vez configurado estos parámetros, el programa deberá adaptar los filtros que necesite para que en los puntos de control se escuchen las señales deseadas y deberá mostrar las señales de error como en los casos anteriores para comprobar que se reduce el error.

Hay que tener en cuenta a la hora de programar el código que, si el usuario cambia de posición, el camino acústico real difiere más de la estima. Pero no es necesario readaptar la estima sino que debe ser el filtro adaptativo el que se readapte para compensar esa mayor diferencia.

Con todas estas premisas y mediante la aplicación GUIDE de Matlab, presentamos en la figura 12 la interfaz gráfica propuesta:



Fig.12. Interfaz gráfica del simulador multicanal adaptativo

El diseño es muy básico y como vemos, se divide en dos partes. La parte de la derecha es la destinada a introducir los parámetros de entrada del simulador y la parte izquierda hace la veces de lienzo donde se irá plasmando la configuración elegida. Además, en la parte izquierda podemos elegir el tipo de señal de entrada elegida y la información que queremos escuchar en cada sensor (como vemos en la figura 14).

Vamos a ilustrar lo explicado con un ejemplo. Si elegimos una configuración propia de un sistema transaural, la parte derecha quedará tal y como se muestra en la figura 13:

Fig.13. Parámetros de entrada de un sistema de reproducción transaural estándar

Y por consiguiente, la parte izquierda quedará de la siguiente manera:

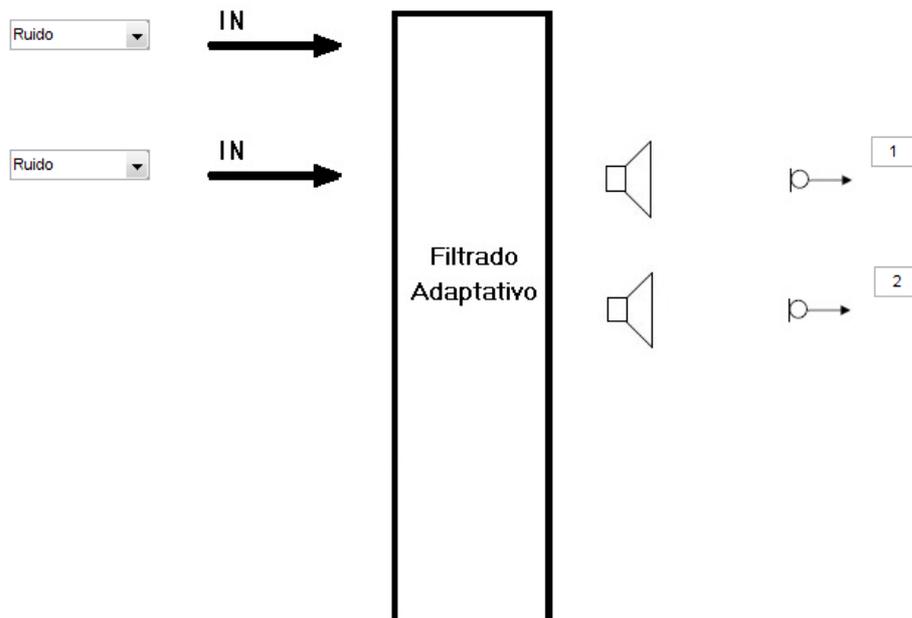


Fig.14. Diagrama de bloques de un sistema de reproducción transaural estándar

Por defecto, el tipo de señal de entrada es ruido blanco pero también podemos elegir entre una señal senoidal o un archivo de audio. A la hora de introducir las señales deseadas en cada sensor, debemos elegir la señal de entrada que queramos escuchar o una combinación de ellas (suma o resta).

Si queremos introducir cambios en la posición de los sensores, pulsaremos el botón correspondiente e introduciremos el instante en segundos en el que queremos que suceda el cambio. El programa nos avisará si el valor introducido es correcto o no. Lo mismo ocurre si queremos cambiar la información en los sensores con el añadido de que en la parte izquierda nos dará la opción de introducir las nuevas señales deseadas.

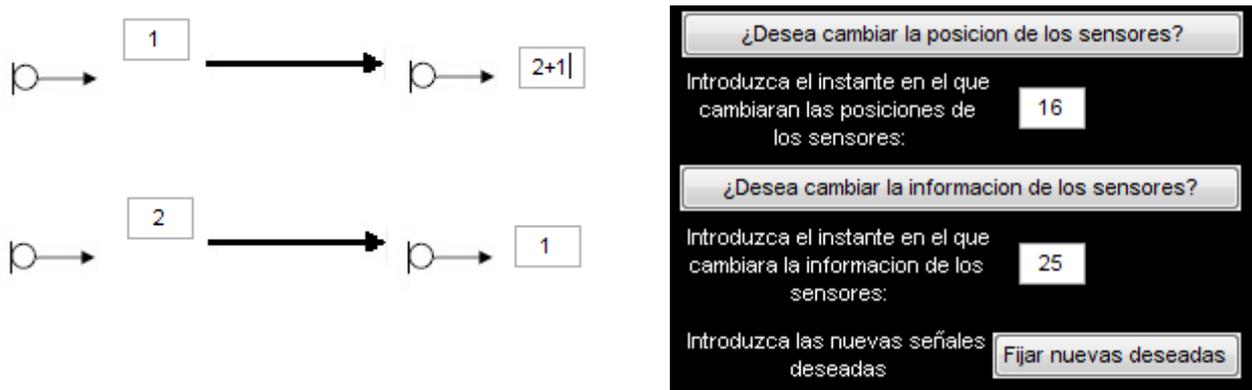


Fig.15. Parámetros de entrada y diagrama de bloques al introducir cambios en el simulador

Una vez decidido el valor de cada parámetro, debemos pulsar el botón *Fijar* para guardar la configuración. Si queremos borrar algún dato, hay que pulsar el botón *Reset* y se volverá a la configuración inicial. Una vez rellenados todos los campos y con todos los datos fijados, pulsaremos el botón *Iniciar* para comenzar la simulación y obtener los resultados.

En el siguiente apartado, vamos a realizar el análisis y comparación de los resultados obtenidos en los simuladores implementados y sobre todo, analizaremos cómo se comportan ante cambios en el entorno acústico.

IV. ANÁLISIS DE RESULTADOS

Una vez implementado el simulador multicanal adaptativo, tenemos que demostrar que es un sistema TSR robusto frente a cambios en el entorno acústico tanto a desplazamientos relativos de las posiciones del oyente como a cambios en la información que llega a cada sensor. Para ello, compararemos resultados con el simulador transaural estático y veremos la necesidad de este tipo de algoritmos.

Además, vamos a definir una serie de parámetros que nos ayudarán a determinar si la cancelación es aceptable o no y a facilitar la comparación de los métodos. Llamamos nivel de cancelación de error en dB, a la relación entre las potencias de la señal de error y la señal deseada:

$$\text{Error (dB)} = 10\log_{10} (\text{error} / \text{señal deseada})$$

Si llamamos K1 al cociente de potencias entre la señal que existe en el sensor y la que realmente se desea:

$$K1 = m1/d1$$

como sabemos que la señal que llega al sensor está compuesta por parte de señal útil más parte de señal indeseada (interferencias debidas al canal cruzado, imprecisiones en la inversión del canal directo, etc):

$$m1 = d1 + I$$

llamaremos relación señal útil a interferente (S/I) será igual a:

$$S/I = 1/(K1-1)$$

$$S/I \text{ (dB)} = 10\log_{10} 1/(K1-1)$$

Por tanto, con ayuda de estos parámetros podremos comparar resultados y determinar qué método es más útil según las circunstancias.

Hay que tener en cuenta que la mejor manera de ver la cancelación de crosstalk no es con las gráficas o parámetros que estamos analizando, sino reproduciendo los archivos de audio provenientes de las simulaciones comparando lo que se escucharía con y sin los algoritmos funcionando ya que subjetivamente se puede apreciar bastante bien si se “cuela” información del otro canal o no. Aunque para ello, debemos usar señales de tonos diferentes o señales de audio porque con las señales de ruido no se notaría la cancelación. Como esto es difícil de mostrar de manera escrita, se propone realizar estas demostraciones en la posterior defensa de esta Tesina.

Primero, vamos a comparar los resultados obtenidos en el sistemas estático y en los adaptativos para un configuración estéreo y luego pasaremos a verificar la robustez de los algoritmos para ver la necesidad de los métodos adaptativos en entornos cambiantes.

IV.2. COMPARACIÓN DE RESULTADOS DE LOS SIMULADORES DE REPRODUCCIÓN TRANSAURAL

Con la configuración mostrada en la figura 3 y con el simulador del cancelador de crosstalk estático implementado, introducimos como parámetros de entrada en dicho simulador los siguientes:

```
>>XTC(1024,randn(300000,1), randn(300000,1));
```

En las figuras 16 y 17, podemos ver las señales de error de los canales 1 y 2 como la diferencia entre las señales originales retardadas y las señales captadas respectivamente:

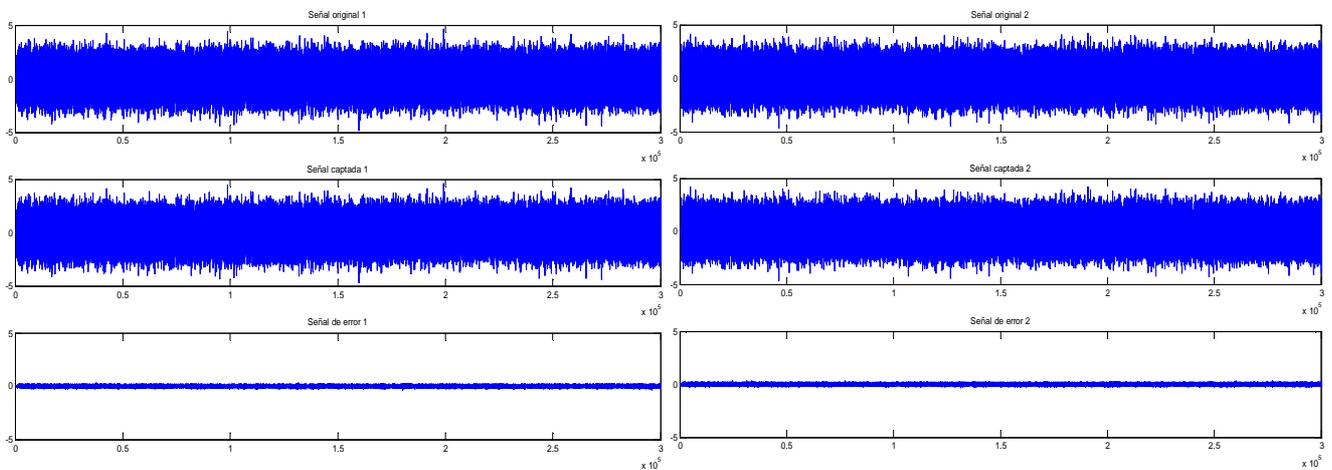


Fig.16.y 17. (derecha) Señal original 1 retardada, señal captada 1 y la señal de error 1.
(izquierda) Señal original 2 retardada, señal captada 2 y señal de error 2.

El nivel de cancelación obtenido para el canal 1 es de -22,1374dB y para el canal 2 es de -22.0328dB. Por lo tanto, se ha conseguido reducir el error unos 22dB. La relación señal útil frente a interferente (S/I), en el canal 1 es de 24,98dB y para el canal 2 es de 20.5297dB.

Como vemos, la diferencia entre las señales originales y las señales que captarían los micrófonos es muy pequeña. Además, donde más error se produce es en partes que se pueden eliminar ya que se corresponden a ceros añadidos en la señal original. Por lo que podemos decir que hemos conseguido diseñar un cancelador de crosstalk aceptable que calcula los filtros inversos de manera sencilla y con un error mínimo.

Mientras, para el simulador de sistema de reproducción transaural adaptativo, una vez implementado el código fijándonos en la figura 7, iniciamos la función en Matlab con la siguiente

configuración, tras realizar pruebas e ir acercándonos a los parámetros que provocan mayor reducción en las señales de error:

```
>>filtrado_xLMS(randn(300000,1), randn(300000,1),60,0.00006);
```

En las siguientes figuras, podemos ver las señales de error de los canales 1 y 2 como la diferencia entre las señales originales retardadas y las señales captadas respectivamente:

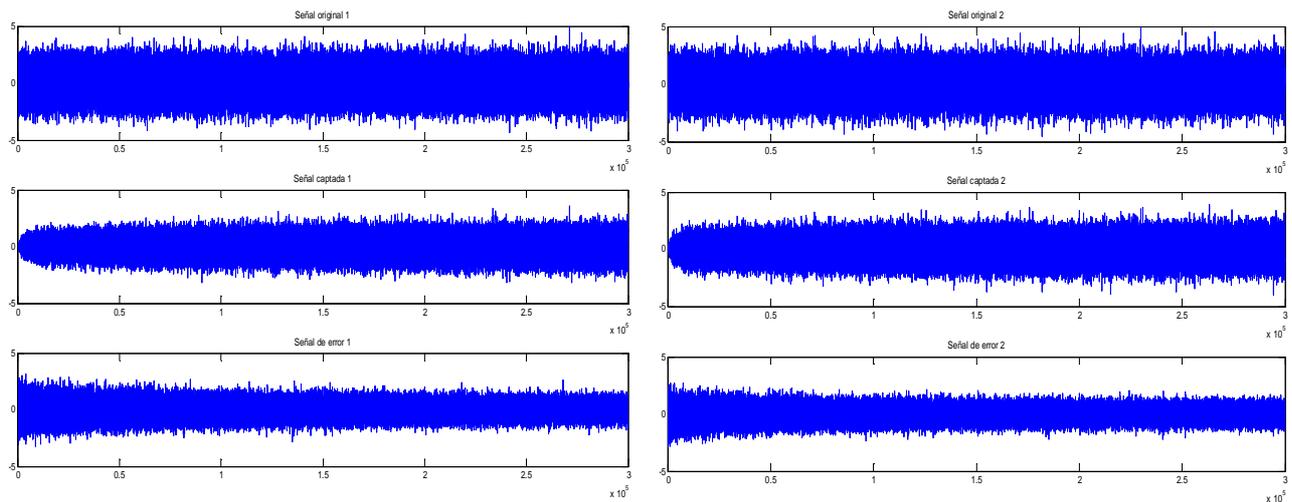


Fig.18.y 19. (derecha) Señal original 1 retardada, señal captada 1 y la señal de error 1.

(izquierda) Señal original 2 retardada, señal captada 2 y señal de error 2.

Si comparamos estos resultados con los obtenidos en el sistema anterior, vemos que el nivel de cancelación obtenido para el canal 1 es de -4,56dB frente a -22,1374dB del método estático y para el canal 2 es de -5,58dB frente a -22.0328dB. Por lo tanto, aunque se ha conseguido reducir el error, el método estático consigue un nivel de cancelación mayor. Respecto a la relación (S/I), en el canal 1 es de 3,0405dB frente a 24,98dB del método estático y para el canal 2 es de 4,3485dB frente a 20.5297dB. Con lo que vemos que la interferencia es mucho mayor en los canales del sistema adaptativo.

Las diferencias son notables y eso muestra que para sistemas de reproducción estéreo sin cambios en el entorno acústico, para posiciones fijas de los puntos de control, el método estático obtiene mejores resultados que el método adaptativo.

Comparar estos resultados con el sistema genérico multicanal no tiene sentido pues en configuración 2:2:2 se obtendrían idénticos que el sistema de reproducción transaural adaptativo.

IV.1. VERIFICACIÓN DE LA ROBUSTEZ DE LOS ALGORITMOS

Ahora vamos a probar porqué hemos elegido algoritmo adaptativos. Vamos a comparar los sistemas estáticos y adaptativos más habituales con y sin cambios en los caminos acústicos y en la información que se desea recibir en cada sensor y veremos la robustez y flexibilidad del algoritmo implementado frente a los sistemas estáticos.

Empezaremos comparando los sistemas de reproducción más habituales:

Caso A.1) Sistemas estático y adaptativo con configuración 1:1:1 sin cambios

Si simulamos el sistema estático con la siguiente configuración:

```
>>XTC_1(1024,randn(300000,1));
```

y el simulador multicanal, mediante la configuración mostrada en el interfaz:

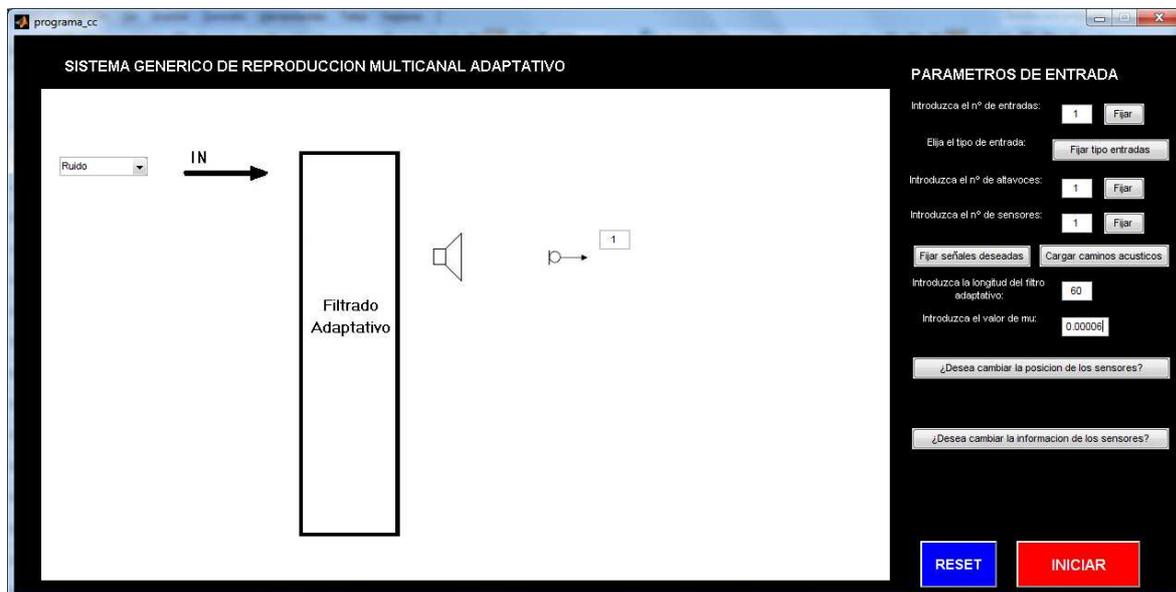


Fig.20. Configuración 1:1:1 del interfaz del simulador multicanal adaptativo sin cambios

Los resultados obtenidos son los siguientes:

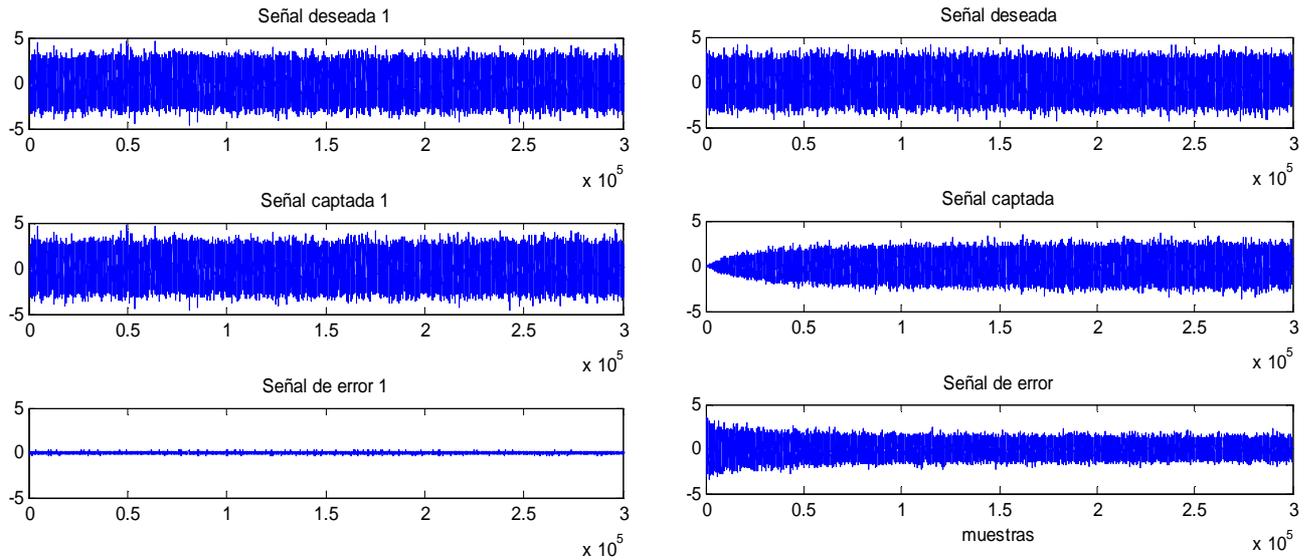


Fig.21 y 22. (derecha) Resultados del sistema estático en configuración 1:1:1 sin cambios. (izquierda) Resultados del sistema adaptativo en configuración 1:1:1 sin cambios

El nivel de cancelación obtenido para el simulador estático es de $-22,2297\text{dB}$ mientras que para el adaptativo es de -5.37dB . Por lo tanto, el método estático consigue un nivel de cancelación mayor. Respecto a la relación (S/I), no es posible calcularla ya que no hay otro altavoz para crear interferencia.

Caso A.1) Sistemas estático y adaptativo con configuración 1:1:1 con cambios

Simular cambios o movimientos en un sistema acústico del método no es más que en un instante dado de la simulación, cambiar los caminos acústicos por otros. Esto no es del todo realista porque nunca cambian los caminos acústicos tan bruscamente, pero nos sirve para estudiar la robustez del sistema.

Simulamos el sistema estático con la siguiente configuración:

```
>>XTC_1(1024,randn(300000,1));
```

En este método, hemos realizado cambios en los caminos acústicos realizando un bucle de dos iteraciones en el que en la primera iteración tenemos los caminos acústicos originales y en la segunda, cambiamos esos caminos por otros. En realidad, tenemos que calcular los filtros inversos dos veces por lo que la carga computacional es el doble. Si en vez de cambiar los caminos una vez, lo queremos hacer varias veces, el tiempo de cálculo sería eterno. Además, estos cambios no son posibles si queremos implementar este método en tiempo real ya que los coeficientes de los filtros son fijos, invariantes en el tiempo y no se adaptarían a cambios de posición de los micrófonos.

Si además, queremos añadir cambios en la información de las señales captadas por los micros, hace que este método sea inviable para cambios en el entorno acústico.

Sin embargo, con el simulador multicanal, podemos cambiar la posición del micrófono, con lo que los caminos acústicos serán diferentes a partir del tiempo que se introduzca en el interfaz.

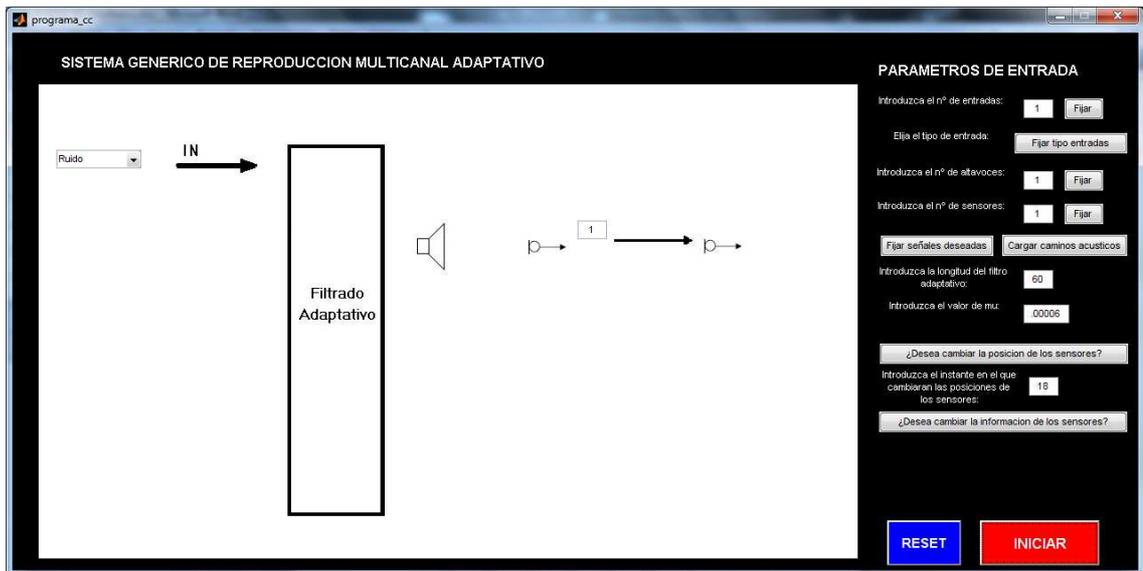


Fig.23. Configuración 1:1:1 del interfaz del simulador multicanal adaptativo con cambios

Los resultados obtenidos son los siguientes:

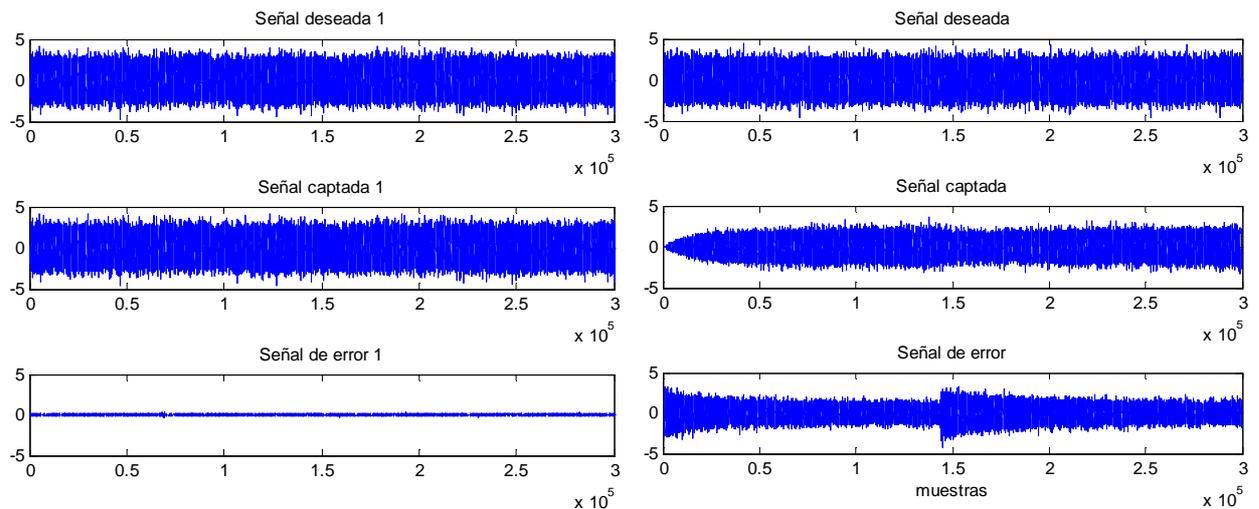


Fig.23 y 24. (derecha) Resultados del sistema estático en configuración 1:1:1 con cambios. (izquierda) Resultados del sistema adaptativo en configuración 1:1:1 con cambios

Podemos ver en las gráficas, como mientras la señal de error del método estático es similar al obtenido a la obtenida sin cambios (fig.21) y es constante ya que no reconoce que ha habido cambios sino que ha vuelto a calcular los filtros otra vez. Sin embargo, vemos que en el método adaptativo, la señal de error en un determinado instante se readapta de nuevo al cambio de los caminos acústicos.

Como ya hemos explicado, aunque el nivel de cancelación sea peor que el método estático (-3.70dB frente a -23.54dB), los métodos adaptativos permiten que los coeficientes de los filtros se adapten y por eso es el método apropiado para nuestro simulador.

Además, comprobamos que aunque el error aumenta al cambiar los caminos, consigue reducir el nivel de cancelación al mismo nivel que antes del cambio, con lo que probamos la robustez del sistema:

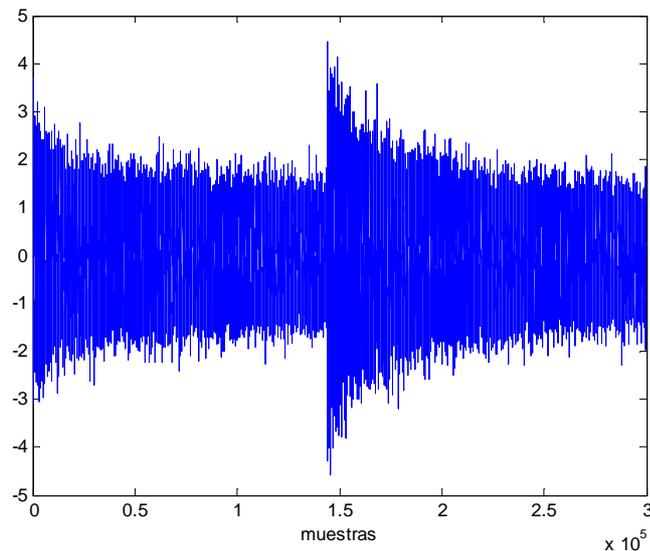


Fig.25. Evolución de la señal de error en el método adaptativo con cambios

Si ahora quisieramos cambiar la información que llega al sensor, debemos primero añadir otra señal de entrada ya que sino no tendría sentido.

Caso B.1) Sistemas estático y adaptativo con configuración 2:2:2 sin cambios

La comparación de esta configuración ya la realizamos en el apartado IV.2. para ver si los resultados obtenidos eran aceptables (pág.30). Llegamos a la conclusión de que para entornos acústicos no variables, el método estático funciona mejor ya que consigue reducir el error unos 22 dB frente a los 3dB que consigue el método adaptativo. Además, comparando la relación S/I,

vemos que la interferencia en el método estático es mucho mayor en los canales del sistema adaptativo.

Como ya hemos repetido varias veces, aunque el error apenas cae 3dB, es mejor que no hacer nada y aunque los resultados obtenidos comparados son peores que en el caso estático, la ventaja de este tipo de algoritmos es la adaptación a entornos acústicas cambiantes.

Caso B.2) Sistemas estático y adaptativo con configuración 2:2:2 con cambios

Simulamos primero el sistema estático con la siguiente configuración:

```
>>XTC_cambios(1024,randn(300000,1), randn(300000,1));
```

En las figuras 16 y 17, podemos ver las señales de error de los canales 1 y 2 como la diferencia entre las señales originales retardadas y las señales captadas respectivamente:

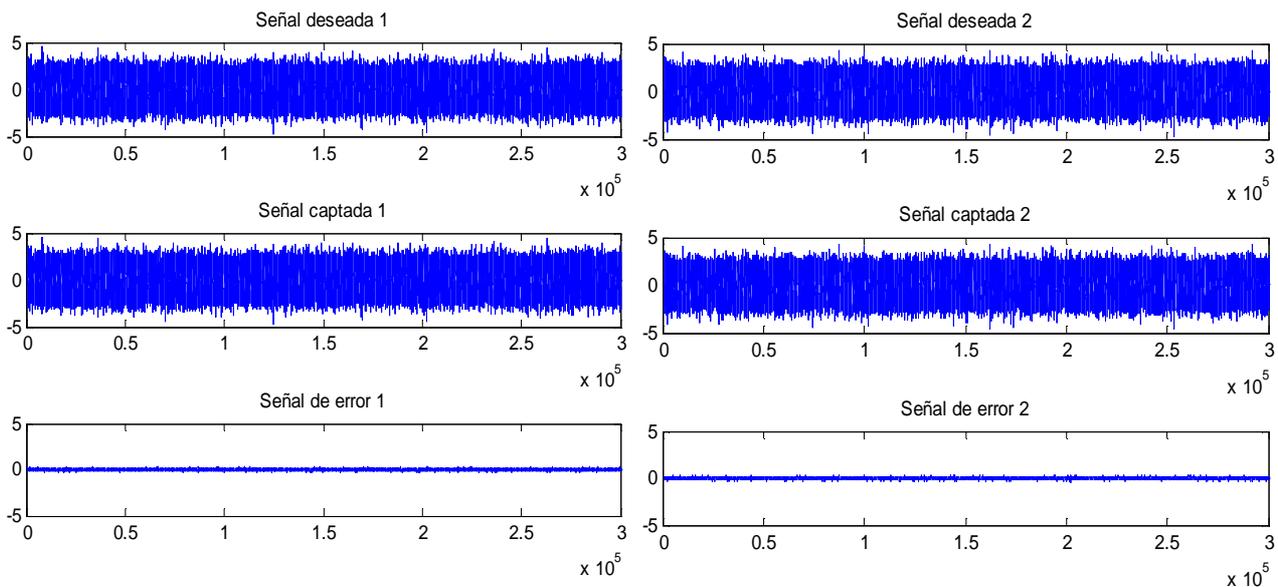


Fig.26 y 27. Resultados del canal1 (derecha) y del canal 2 (izquierda) del sistema estático en configuración 2:2:2 con cambios.

El nivel de cancelación obtenido para el canal 1 es de - a -22,61dB y para el canal 2 es de -- 22,02dB. Por lo tanto, se ha conseguido reducir el error unos 22dB. La relación señal útil frente a interferente (S/I), en el canal 1 es de 23,63dB y para el canal 2 es de 21.22dB.

Como vemos, los resultados son idénticos a los obtenidos sin cambios por lo que hemos comentado con anterioridad.

Para el simulador multicanal, usaremos la siguiente configuración mostrada en el interfaz:

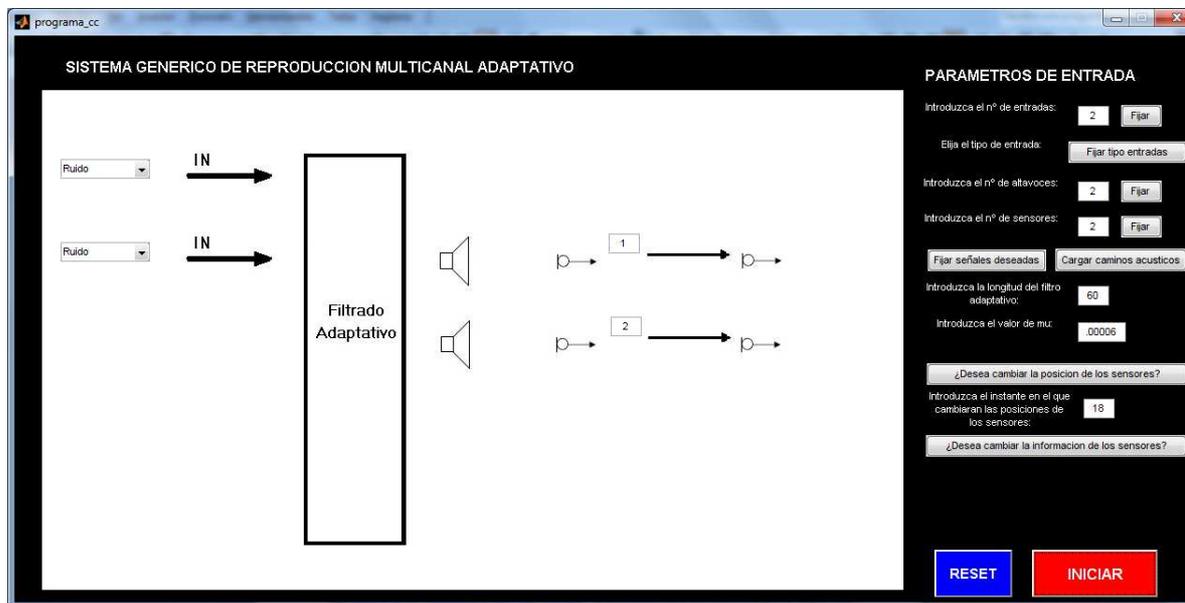


Fig.28. Configuración 2:2:2 del interfaz del simulador multicanal adaptativo con cambios en los caminos

Y los resultados obtenidos son:

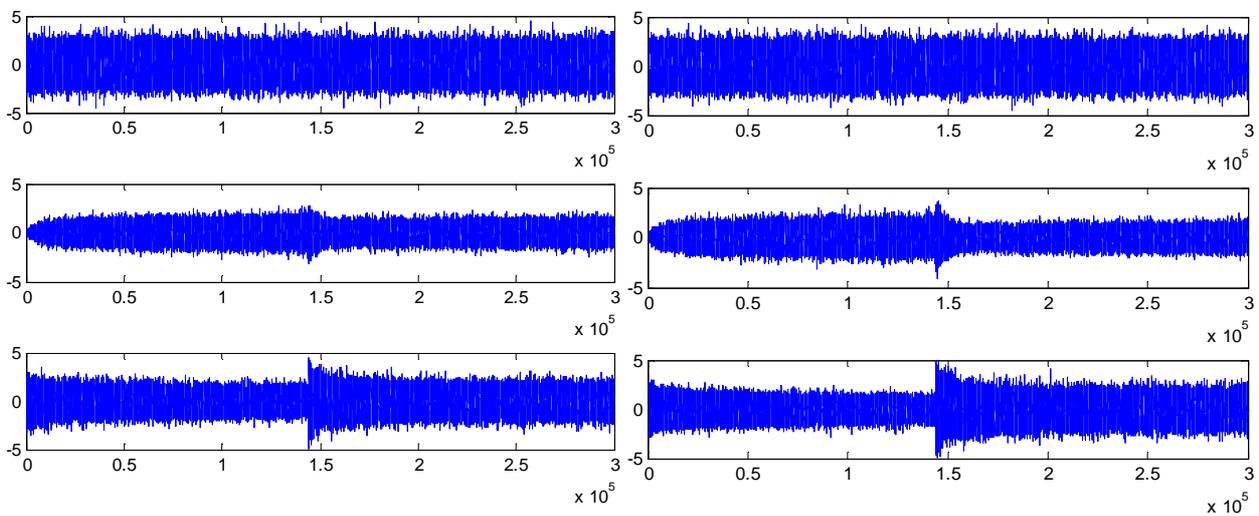


Fig.29 y 30. Resultados del canal1 (derecha) y del canal 2 (izquierda) del sistema adaptativo en configuración 2:2:2 con cambios.

Si comparamos estos resultados con los obtenidos en el sistema anterior, vemos que el nivel de cancelación obtenido para el canal 1 es de -22,23dB frente a -22,61dB del método estático y para el canal 2 es de -2,10dB frente a -22,02dB. Como ya hemos visto en otras configuraciones, el método estático consigue un nivel de cancelación mayor. Respecto a la relación (S/I), en el canal 1 es de

1,70dB frente a 23,63dB del método estático y para el canal 2 es de 2,07dB frente a 21.22dB. Con lo que vemos que la interferencia es mucho mayor en los canales del sistema adaptativo.

Comparando con los resultados obtenidos en el método adaptativo sin cambios con configuración 2:2:2, vemos que en esta ocasión no hemos vuelto a la cancelación que teníamos antes del cambio puesto que hemos pasado de reducir 4,56dB a 2,23dB en el canal 1 y para el canal 2, hemos pasado de reducir 5,58dB a 2,10dB. Por lo tanto, al aumentar el número de entradas, altavoces y micrófonos, hemos obtenido peor nivel de cancelación, alrededor de 2dB menos. Respecto a la relación (S/I), en el canal 1 es de 3,0405dB frente a 1,70dB y para el canal 2 es de 4,34dB frente a 2,07dB. Por lo tanto, obtenemos peor relación S/I, es decir, tenemos más presente la señal interferente al aumentar el número de entradas, altavoces y micrófonos, aumenta casi la mitad.

Si ahora, además cambiamos la información que llega a los sensores de la siguiente manera:

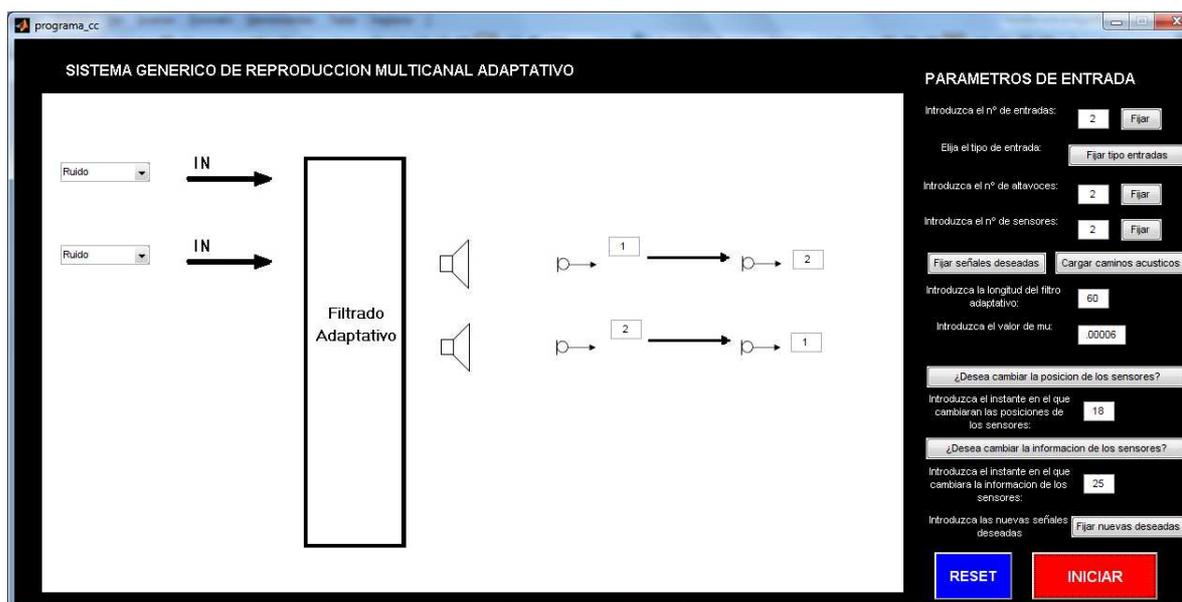


Fig.31. Configuración 2:2:2 del interfaz del simulador multicanal adaptativo con cambios en los caminos y en las señales deseadas

Los resultados obtenidos han sido los siguientes:

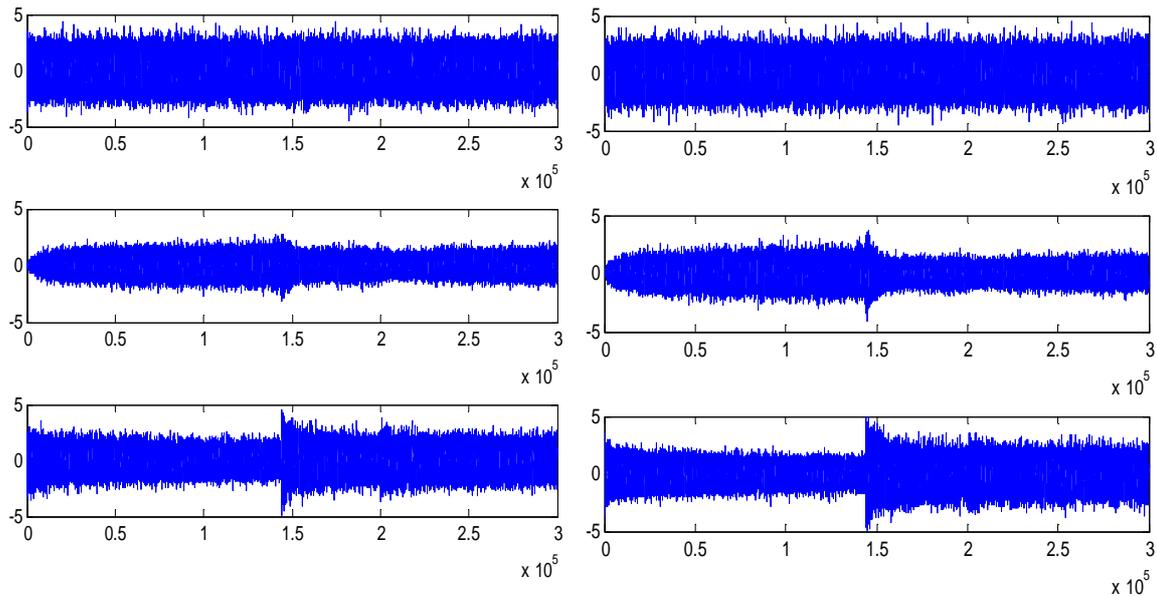


Fig.32 y 33. Resultados del canal1 (derecha) y del canal 2 (izquierda) del sistema adaptativo en configuración 2:2:2 con cambios en los caminos y en las señales deseadas.

Si comparamos con los resultados obtenidos con cambios en los caminos acústicos, hemos pasado de reducir 2,23dB a 2,04dB en el canal 1 y para el canal 2, hemos pasado de reducir 2,10dB a 1,95dB. Respecto a la relación (S/I), en el canal 1 es de 1,70dB frente a 1,60dB del método estático y para el canal 2 es de 2,07dB frente a 1,97dB.

Como vemos, las diferencias son mínimas por lo que podemos decir que si cambiamos la posición de los micrófonos obtenemos un nivel de cancelación y una relación S/I muy similar a sí además cambiamos la información de las respuestas deseadas en los sensores.

Por lo tanto, una vez estudiados los casos A y B, a modo de conclusión, podemos afirmar que el algoritmo es estable a cambios en los caminos acústicos y que las ventajas que obtiene el sistema adaptativo frente a los no adaptativos es que obtienen mejor comportamiento en caso de variación de los caminos acústicos y sobre todo se readapta automáticamente si se decide cambiar las señales deseadas.

Otras configuraciones interesantes que hemos estudiado han sido sistemas adaptativos con configuración 2:2:4 sin y con cambios para simular que hay dos oyentes, sistemas con configuración de 4:4:4 con y sin cambios para ver cerciorar si al aumentar el número de entradas, fuentes y micrófonos empeora el nivel de cancelación, comprobar si aumentando el número de altavoces, se obtienen mejores resultados, cambiar el tipo de señales de entrada, etc...pero se ha preferido mostrar los casos restantes en la defensa de la Tesina con la ayuda de escuchas de señales obtenidas en cada caso para notar si existe cancelación o no.

V. CONCLUSIONES

En la siguiente sección se describen las contribuciones realizadas en esta tesina:

- Se ha desarrollado un sistema multicanal adaptativo gobernado por el algoritmo de filtrado-XLMS robusto y flexible frente a desplazamientos relativos de las posiciones de los micrófonos y frente a cambios en las señales deseadas.
 - Para ello, se ha diseñado una interfaz gráfica que nos permite introducir todo tipo de configuraciones multicanal (hasta un límite decidido por criterio propio, no por limitaciones del sistema), visualizarlas correctamente y realizar simulaciones, con y sin entornos acústicos variables, con facilidad.
 - Se ha comprobado las ventajas que tienen los filtros adaptativos ante los estáticos. Los métodos adaptativos son útiles cuando no se conoce de antemano que filtro debemos usar. Por lo tanto, nos han permitido diseñar un sistema flexible que se readapta ante cambios en el entorno acústico. Sin embargo, el cancelador de crosstalk estático no es un sistema modificable a posteriori, sino que suele ser un sistema poco flexible. Los coeficientes de los filtros adaptativos cambian con el tiempo, son calculados cuando el filtro se implementa y se reajustan automáticamente en cada iteración mientras dura su fase de aprendizaje. En contraposición, los coeficientes de los filtros fijos son, teóricamente, invariantes con el tiempo. Se ha comprobado que el algoritmo de filtrado-XLMS es un método simple de implementar y es estable cuando el paso de convergencia seleccionado es el óptimo.
 - Se ha comprobado que al aumentar el número de entradas, altavoces y micrófonos, hemos obtenido peor nivel de cancelación. Hemos visto que para obtener más o menos buenos resultados, el número de altavoces debe ser similar al número de puntos de control y el número de fuentes no debe ser mayor que el de altavoces. A su vez, aunque existan sistemas que no cumplan estas condiciones, el sistema responde aunque los resultados no sean del todo óptimos. También se ha constatado, que cuanto mayor sea el número de altavoces que empleemos, mayor será la exactitud de la solución buscada.
 - Se ha comprobado que los sistemas adaptativos funcionan mejor que los estáticos ante cambios en las condiciones del entorno y aunque el error apenas cae 3dB, es mejor que no utilizarlos. Dado que el sistema desarrollado es crítico en cuanto a la elección de parámetros (como el paso de convergencia y la longitud de los filtros), y que es posible depurar mejor el código implementado, el ajuste del simulador y por tanto, sus resultados, tienen margen de mejora.
-

VI. LÍNEAS FUTURAS

A continuación se describen las líneas de investigación que quedan abiertas después del trabajo realizado en esta tesina:

- Se podría programar el sistema multicanal obtenido para una aplicación en tiempo real en C o en cualquier lenguaje que permita interactuar de una forma cómoda con las tarjetas de audio.
 - Para comparar mejor los resultados, también podría implementarse un simulador genérico multicanal para el método estático.
 - Aunque se ha explicado que una de las ventajas que obtiene el sistema adaptativo frente a los no adaptativos es que obtienen mejor comportamiento en caso de variación de los caminos acústicos, habría que estudiar el margen de variación para el cual esto ocurre.
 - Aunque el algoritmo de filtrado-XLMS es uno de los más utilizados dada su estabilidad y su velocidad de convergencia, para configuraciones multicanal con un número alto de altavoces y micrófonos y con cambios en el entorno acústico, el coste computacional es demasiado elevado, más aún para la implementación en tiempo real. Por lo que sería conveniente estudiar otras técnicas para intentar reducirlo [1]. También, sería conveniente comprobar como se comporta el sistema cuando los caminos estimados no son iguales a los acústicos. Por estas dos razones, se propone el método descrito en [8] que utiliza una técnica para realizar el crosstalk de forma adaptativa sin usar estima de los caminos acústicos y utilizando dos filtros FIR llamados filtros correctores en cada altavoz para compensar la distorsión de crosstalk que producen los caminos acústicos. Estos filtros contienen la información de los filtros adaptativos, es decir, se actualizan según los valores de los coeficientes de h multiplicados por un peso w para decidir qué coeficientes del filtro inverso son más importantes.
 - Hoy día, uno de los sistemas de reproducción de audio espacial más prometedores es Wave Field Synthesis (WFS) [9], ya que permite la generación de campos sonoros con características temporales y espaciales dentro de un área amplia de escucha por medio de arrays de altavoces. Este método ofrece un área de escucha amplia con alta calidad y permite que un oyente pueda situarse en cualquier posición de la sala e incluso moverse por ella sin que eso altere la imagen sonora reconstruida. Por lo tanto, se podría implementar el sistema desarrollado usando técnicas de WFS, donde las fuentes se colocarían virtualmente en el espacio. Inicialmente se podría trabajar en espacio libre, para posteriormente buscar la solución en recintos reverberantes y también, contrastar los resultados de la simulación con resultados de medidas reales.
-

BIBLIOGRAFÍA

- [1] M.Bouchard, S.Quednau, *Multichannel Recursive-Least-Squares Algorithms and Fast-Transversal-Filter Algorithms for Active Noise Control and Sound Reproduction Systems*. IEEE transactions on speech and audio processing, Vol.8, N°.5, September 2000.
 - [2] O.E. Moreno, A. González, J. J. López, *Implementación de un Cancelador de crosstalk en una plataforma de bajo coste*. Grupo de Tratamiento de Audio y Comunicaciones (GTAC). Instituto de Telecomunicaciones y Aplicaciones Multimedia (iTEAM). Universidad Politécnica de Valencia.
 - [3] P.A. Nelson, H. Hamada, S.J. Elliott , *Adaptive Inverse Filters for Stereophonic Sound Reproduction*.IEEE Transactions on Signal Processing, Vol.40, N°7, 1992.
 - [4] S.J. Elliott, I.M. Stothers, P.A. Nelson, *A Multiple Error LMS Algorithm and Its Application to The Active Control of Sound and Vibration*. IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-35, N°.10, 1997.
 - [5] T. Nufire, *Crosstalk Cancellation*. 7700 Geary Boulevard, #210 San Francisco, CA 94121 415-752-9313.
 - [6] S. Saiki, M. Fukumoto, L. Liu, *Sound Field Reproduction System Exploiting Cross-talk Cancellation Adaptive Filter*. Kochi University of Technology, Kami 782-8502 Kochi Japan.
 - [7] M. Ferrer, *Filtrado multicanal para control local sonoro basado en algoritmos de proyección afín*. 2008. Grupo de Tratamiento de Audio y Comunicaciones (GTAC). Instituto de Telecomunicaciones y Aplicaciones Multimedia (iTEAM). Universidad Politécnica de Valencia.
 - [8] JunSeong Kim, SangGyun Kim, Chang D.Yoo, *A novel adaptive crosstalk cancellation using psychoacoustic model for 3D audio*. Dept. of EECS, Div. of EE, KAIST 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea.
 - [9] G.Theile, H.Witteck, *Wave field synthesis – a promising spatial audio rendering concept* .Institut für Rundfunktechnik (IRT), Germany. Published in Journal of the Institute of Image Information and Television Engineers, 2007.
-

ANEXO

Ficheros Adjuntos a la Memoria

Se adjunta dentro del CD, un archivo .RAR con todos los códigos de los sistemas implementados, todos los archivos necesarios para el funcionamiento de la interfaz gráfica, caminos acústicos y archivos de audio utilizados.
