

Modelado y evaluación del algoritmo LSL para la cancelación de ruido en señales de voz.

Francisco DIAZ PEREZ

Arquitectura y Tecnología de Computadores-Universidad Politécnica de Madrid
Madrid, 28031, España

Victoria RODELLAR BIARGE

Facultad de Informática, Universidad Politécnica de Madrid
Madrid - Boadilla del Monte 28660, España

Elvira MARTÍNEZ DE ICAYA

Arquitectura y Tecnología de Computadores-Universidad Politécnica de Madrid
Madrid, 28031, España

y

Pedro GOMEZ VILDA

Facultad de Informática, Universidad Politécnica de Madrid
Madrid - Boadilla del Monte 28660, España

RESUMEN

Es conocido el uso de algoritmos adaptativos para la cancelación de ruido en diversos campos, en especial nos interesa abordar el procesamiento de señales de voz en entornos de reuniones o debates “multi-party”. Este trabajo está organizado en tres partes; primero se realiza una implementación totalmente parametrizable y flexible del algoritmo LSL “*Least Squares Lattice*”. Esta implementación es útil para la configuración óptima de los parámetros (épsilon, tasa de aprendizaje, número de etapas), y la selección de los formatos numéricos adecuados para lograr un desarrollo eficiente en hardware reconfigurable. En la segunda parte se presenta el modelo LMMSD “*Lineal Mixture Model for Signal Data*”, este modelo se utiliza para generar conjuntos de señales con el objetivo de evaluar algoritmos de cancelación de ruido. A diferencia de otros modelos, este admite la contaminación de la señal de referencia de ruido con la señal primaria de voz, y viceversa. Finalmente, en la tercera parte, se utiliza el modelo de datos desarrollado en experimentos con diferentes patrones de señales. Los resultados obtenidos revelan la importancia de la señal de ruido de referencia en la cancelación y posibilita la mejora del algoritmo presentado.

Palabras Claves:

Cancelación adaptativa de ruido, Procesamiento de voz. Simulación.

1. INTRODUCCIÓN

Desde 1975 en que Windrow y otros [5] formularon el concepto de “Cancelación adaptativa de ruido” muchos investigadores han implementado y elaborado algoritmos orientados a la eliminación o reducción de señales no deseadas. El proceso de cancelación de ruido tiene gran interés en diversos campos; por ejemplo en aplicaciones de reconocimiento de voz en entornos ruidosos, en las mejoras de las comunicaciones en video-conferencias, también en llamadas telefónicas que se producen en zonas industriales o

centros comerciales donde hay diferentes fuentes de ruido. Otras aplicaciones, no relacionadas directamente con la señal de voz, las podemos encontrar en telemedicina; en los exámenes de ECG para fetos es común que ocurran interferencias con el latido del corazón de la madre.

En especial nos interesa la cancelación de ruido en señales de voz, en entornos de reuniones o debates “multi-party”, donde se requiere un tratamiento diferente al filtrado de otras señales. En los entornos “multi-party” tanto la señal primaria como la fuente de ruido no son estacionarias puesto que corresponden a discursos orales, en [6] se describe un sistema general de reducción de ruido para la voz, con dos fuentes de señales.

La cancelación de ruido basada en el filtrado estático puede ser útil en situaciones donde el ruido es invariante en el tiempo, pero resulta poco efectiva cuando el ruido o la señal primaria tienen un comportamiento dinámico, en estos casos es necesario realizar el filtrado adaptativo. El filtrado adaptativo tiene dos fases definidas en el tiempo; en una primera fase el filtro se “adapta” o varía sus pesos recursivamente para optimizar cierto criterio establecido inicialmente, y en una segunda fase se produce el filtrado de la señal de entrada. Después de cierto tiempo de adaptación, ambas fases actúan de forma solapada, es decir realizando el filtrado y adaptándose a los nuevos patrones de entrada.

Entre los requisitos generales que debe poseer un filtro adaptativo se encuentran; alta velocidad de convergencia, robustez, adecuada precisión numérica y baja complejidad computacional. En una implementación determinada estos requisitos pueden entrar en conflicto y es necesario llegar a un estado de compromiso entre estos factores.

Son conocidos los algoritmos RLS “*Recursive Least Squares*”, LMS “*Least Mean Squares*”, y LSL “*Least Squares Lattice*”, es imposible afirmar que uno de ellos es mejor que otro, y todos tienen ciertas ventajas e inconvenientes [7], [3], [4], [2]. En este trabajo se analiza únicamente el Algoritmo LSL “*Least Squares Lattice*”, que es conocido por su robustez [8], [1].

El trabajo se organiza como sigue; en la sección 2 y 3 se exponen los fundamentos básicos del algoritmo LSL y se describe, según una metodología de desarrollo Top-Down, la

implementación realizada en el entorno de simulación Matlab/Simulink. Posteriormente, en la sección 4 se presenta el modelo LMMSD “*Lineal Mixture Model for Signal Data*” para generar datos de pruebas y evaluación de la cancelación de ruido. En la sección 5 se realizan algunos experimentos con diferentes patrones de señales contaminadas. Al final exponemos las conclusiones.

2. FUNDAMENTOS DEL ALGORITMO LSL

EL algoritmo LSL realiza un filtrado adaptativo de M etapas configurables con estructura de celosía en el dominio del tiempo [6]. Las salidas de la etapa m -ésima asociada a la muestra n se expresan por las siguientes ecuaciones:

$$f_m(n) = f_{m-1}(n) - K_m^f b_{m-1}(n-1) \quad (1)$$

$$b_m(n) = b_{m-1}(n-1) - K_m^b f_{m-1}(n) \quad (2)$$

donde los valores K_m^f y K_m^b reciben el nombre de coeficientes de reflexión, y las variables $f_m(n)$ y $b_m(n)$ representan los errores m -ésimos hacia delante “*forward*” y hacia atrás “*backward*” respectivamente.

En este sentido, K_m^b representa el factor en el cual $f_{m-1}(n)$ está en la dirección de $b_{m-1}(n-1)$, y del mismo modo K_m^f representa el factor en el cual $b_{m-1}(n-1)$ está en la dirección de $f_{m-1}(n)$. Para una muestra $x(n)$ los correspondientes valores de $f_m(n)$ y $b_m(n)$ se formulan como predictores lineales:

$$f_m(n) = x(n) - \sum_{k=1}^m a_{mk} x(n-k) \quad (3)$$

$$b_m(n) = \quad (4)$$

$$x(n-m) - \sum_{k=1}^m a_{mk} x(n-m+k) \quad (5)$$

Los coeficientes a_{mk} deben ser actualizados en función de minimizar el error cuadrático medio ponderado; hacia delante ε_m^f y hacia atrás ε_m^b respectivamente:

$$r_m^f = \min\{\varepsilon_m^f(n)\} = \min\{\sum_{k=0}^n W^{n-k} |f_m(k)|^2\} \quad (6)$$

$$r_m^b = \min\{\varepsilon_m^b(n)\} = \min\{\sum_{k=0}^n W^{n-k} |b_m(k)|^2\} \quad (7)$$

donde W es el factor de ponderación.

3. IMPLEMENTACIÓN EN MATLAB-SIMULINK

La inestabilidad numérica inherente al algoritmo LSL es objeto de frecuentes estudios tanto en desarrollo con formatos en coma fija como en coma flotante, pues la robustez implica elevar el número de bits en el formato de los datos, la cuestión general radica en encontrar la mejor distribución de los formatos numéricos en punto fijo, para reducir la propagación de errores, o los desbordamientos. Una implementación basada en bucles, no posibilita un estudio localizado de los formatos numéricos, mientras que si desglosamos secuencialmente las etapas del filtrado podemos realizar un análisis más real.

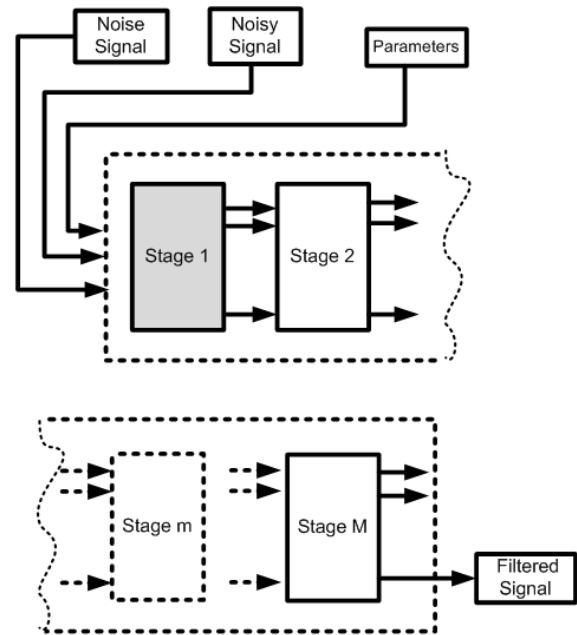


Figura 1. Diagrama en Bloques del modelo de simulación. (Nivel Superior).

La implementación del algoritmo LSL en Matlab/Simulink que se presenta tiene por objetivo desarrollar una plataforma base “base line” para estudiar las posibilidades de paralelismo en las operaciones de los diferentes bloques, así como el estudio de diferentes formatos en punto fijo de las variables y datos para lograr una síntesis eficiente en hardware reconfigurable. Mediante la metodología “*Top-Down*”, se elaboraron diferentes bloques de procesos en tres niveles jerárquicos.

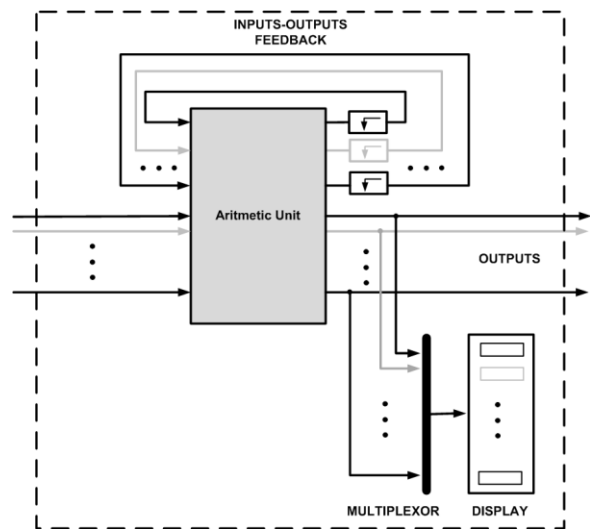


Figura 2. Diagrama en Bloques para la m -ésima etapa. (Nivel 2).

La Figura 1 muestra el nivel superior donde se representa el modelo completo. En el lado izquierdo se identifican tres bloques de entrada; (1) la señal ruidosa, o señal primaria corrompida con ruido, (2) la señal de ruido de referencia, y (3) los parámetros del algoritmo. En el centro, enmarcado con líneas discontinuas se encuentra la secuencia de etapas de

filtrado, la primera etapa es diferente al resto, a partir de la segunda etapa, las salidas de la etapa anterior son las entradas de la etapa siguiente, finalmente el resultado del algoritmo se obtiene en la salida de la última etapa. El número de etapas es parametrizable y se establece antes de realizar el procesado.

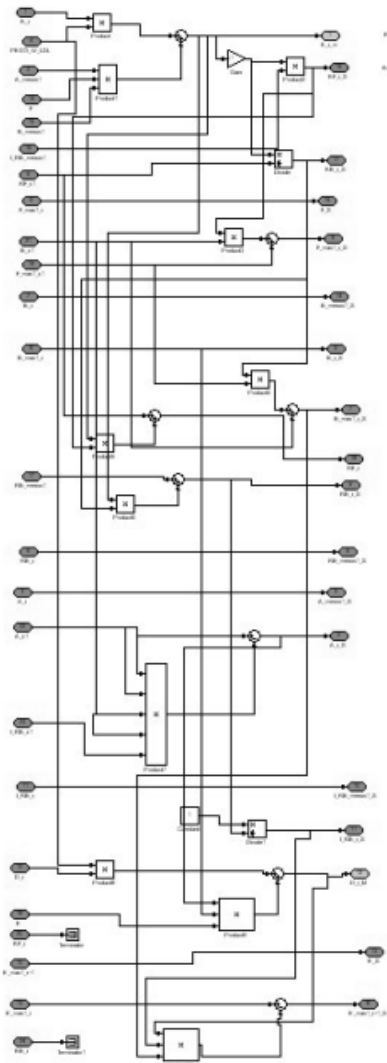


Figura 3. “Vista de Pájaro” de la Unidad Aritmética. (Nivel 3).

El contenido de las etapas de filtrado son especificadas en el nivel 2. Hay dos modelos de etapas como es de esperar; un modelo que describe la primera etapa, y otro modelo que describe a las demás etapas. En este nivel solo se especifican los puertos de entrada y salida a la unidad aritmética propia de la etapa. En la Figura 2 se detalla el nivel 2 para la “*m-esima*” etapa, en la parte superior se representan aquellos datos de salida que son retroalimentados, lo que requiere el almacenando del dato obtenido en la ejecución anterior de la propia etapa, esta persistencia de los datos se logra con inserción de bloques “*Memory*” entre la salida y la entrada correspondiente. Adicionalmente, en este nivel se ha incluido un bloque “*Display*” que permite visualizar el estado actual de la etapa, aunque este bloque no tiene interés para la síntesis hardware del algoritmo, el mismo se ha incorporado para facilitar el estudio del comportamiento del bloque con diferentes parámetros y formatos numéricos.

En el tercer nivel, o nivel inferior, al igual que en el nivel 2, hay dos modelos, uno para la primera etapa y otro para el resto de las etapas, en este nivel se describe la unidad aritmética propia de cada etapa. El contenido de las unidades aritméticas se detalla en un trabajo previo [6]. En la Figura 3 se muestra la unidad aritmética a “vista de pájaro” para el bloque de la fFigura 2. Como se observa, algunas operaciones se ejecutan en paralelo mientras que otras tienen dependencia secuencial entre ellas.

4. MODELO DE DATOS LMMSD.

En un sistema real la señal de voz ruidosa (señal mezclada con ruido), y la señal de referencia de ruido se pueden registrar con sendos micrófonos orientados convenientemente y dispuestos cada uno cerca de la fuente adecuada; ya sea voz o ruido, sin embargo, la señal de voz primaria (señal de voz libre de ruido) solo se puede estimar. Para evaluar la funcionalidad de cualquier algoritmo de cancelación de ruido es necesario disponer además de las dos señales mencionadas anteriormente, de la señal de voz primaria. Estimar la señal de voz primaria es el objetivo final del proceso de cancelación de ruido, sin embargo en sistemas reales, es muy difícil su registro, y en algunos casos imposible, ya que el registro que se produce en un micrófono siempre contiene elementos contaminantes.

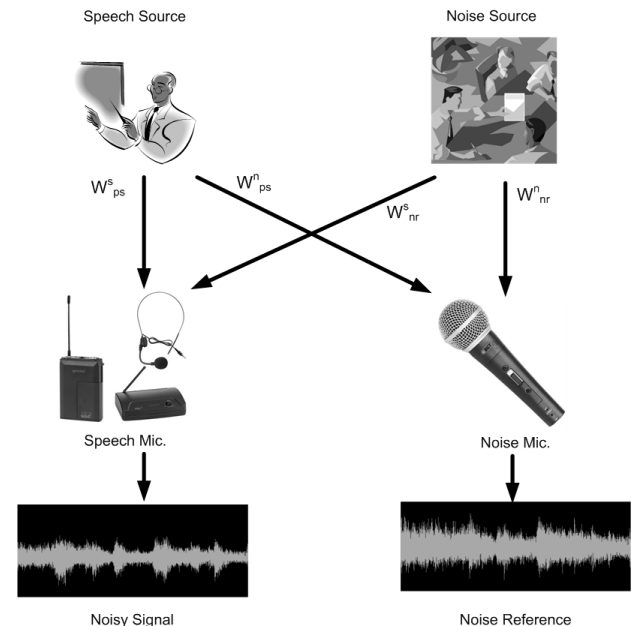


Figura 4. Modelo de datos LMMSD.

El presente modelo de datos, además de conformar patrones de datos, es útil para realizar una evaluación teórica de cualquier algoritmo de cancelación. La efectividad de un algoritmo de cancelación de ruido se pueden cuantificar por la medida de la similitud entre la señal filtrada y la señal de voz primaria. En este estudio se propone el modelo LMMSD “*Lineal Mixture Model for Signal Data*”. El modelo LMMSD opera en un escenario “*multy party*” acorde a la Figura 4.

El modelo LMMSD permite generar patrones de señales tomando dos fuentes ideales; la primera; *Speech Source*, para la señal primaria, y la segunda, *Noise Source* para el ruido. En cualquier sistema de registro de señales, no ideales. La señal producida en las fuentes están expuestas a dos tipos básicos de transformaciones; degradación y contaminación.

La degradación se explica por la pérdida de energía que sufre la señal desde el sitio en que se origina al punto en que se registra, de la misma forma, la señal que se registra está expuesta a ciertas transformaciones motivadas por la interferencia de otras señales.

En la parte superior de la gráfica del modelo LMMSD se detallan las dos fuentes principales: Señal Primaria (PS , *primary signal*); que representa la voz limpia, que queremos extraer con la cancelación de ruido, y la señal de Ruido Puro (PN , *pure noise*); que representa la fuente de ruido más importante del entorno en que se realiza el experimento, más abajo aparecen sendos micrófonos, asociados, que registran las respectivas señales para realizar la cancelación adaptativa. Como es lógico, las fuentes no están aisladas, y en las señales que registran los micrófonos hay cierto contenido de la otra fuente. La funcionalidad de los micrófonos NS y NR se puede formular según las funciones de transferencias:

$$NS = \varphi^s(\text{signals}(PS, PN), \text{weights}(w_{ps}^s, w_{nr}^s)) \quad (8)$$

$$NR = \varphi^n(\text{signals}(PS, PN), \text{weights}(w_{ps}^n, w_{nr}^n)) \quad (9)$$

sujeto a las siguientes restricciones:

$$\varphi^s = \varphi^n, \quad (10)$$

para todo $k = s, n$; $e \ i = ps, nr$ se cumple:

$$0 < w_i^k < 1, \quad (11)$$

$$y \quad w_{ps}^s > w_{nr}^s, \quad w_{nr}^n > w_{ps}^n$$

La ecuación 10 implica que ambos micrófonos operan de la misma forma realizando una función de transferencia (φ^s, φ^n) de las señales PS y PN en diferentes proporciones acorde a los pesos w_i^k asociados. Supuestamente en las fuentes se genera la señal con una potencia, o energía, normalizada igual a la unidad. En el paso de la señal, desde la fuente hasta los micrófonos se produce alguna degradación y/o contaminación. Los factores de degradación y contaminación se consideran invariables en el tiempo y es modelado con los pesos w_i^k , de esta forma el micrófono S (*Speech*) registra la señal ruidosa NS (*Noisy Signal*) con los pesos asociados w_{ps}^s, w_{nr}^s , por su parte el micrófono N (*Noise*) registra el ruido de referencia NR (*Noise Reference*) mediante las ponderaciones asociadas w_{ps}^n, w_{nr}^n . Cada micrófono registra con mayor potencia la señal asociada a la fuente, por ejemplo, si elaboramos un modelo de datos con los siguientes valores $w_{ps}^s = 1, w_{nr}^s = 0.6, w_{ps}^n = 0.8$, y $w_{nr}^n = 0.2$, nos indica que la señal ruidosa contiene una combinación de máxima potencia normalizada (valor unidad) para la señal fuente de voz, y una potencia normalizada de 0.6 para la fuente de ruido, similarmente la señal de ruido de referencia esta compuesta por el 0.8 de la potencia de la fuente de ruido, y 0.2 de la potencia de la fuente de voz. La transformación que se produce en las señales registradas no se limita simplemente a variar la amplitud de las señales, sino también, se producen cambios en el espectro de frecuencias. En un experimento real los valores de los pesos asociados al modelo dependen de muchos factores, por ejemplo de las características constructivas de los micrófonos, de la orientación, y de la distancia a la fuente, estos entre otros factores. Según este modelo los sistemas muy ruidosos se caracterizan por valores de w_{nr}^s elevados, cercanos a la unidad.

Asimismo los valores de w_{ps}^n tienden a ser muy pequeños cercanos a cero, de hecho, en muchos esquemas de cancelación de ruido se obvia la contaminación de la señal de referencia de ruido con la señal primaria, sin embargo resulta conveniente la influencia de este factor ($w_{ps}^n > 0$) en la efectividad de la cancelación de ruido.

Una medida de la efectividad de un algoritmo de cancelación de ruido se puede definir por la raíz cuadrada del error cuadrático medio RMSE (Root Mean Square Error) de la señal filtrada con respecto a la señal primaria, o sea:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (PS_i - FS_i)^2}{N}} \quad (12)$$

donde:

- N : Número de observaciones de las señales.
- PS_i : i -ésima observación de la señal primaria.
- FS_i : i -ésima observación de la señal filtrada.

5. EXPERIMENTOS Y RESULTADOS

Para representar las señales de entradas al algoritmo de cancelación acorde al modelo LMMSD descrito en el epígrafe anterior se realizó una grabación libre de ruido de la secuencia de dígitos "nine six eight nine six eight", que corresponde a la señal "Speech Source", y para la señal "Noise Source" se tomó un segmento, de igual longitud que la anterior, de la conocida canción de Frank Sinatra "My Way", estas señales se observan en la Figura 5.

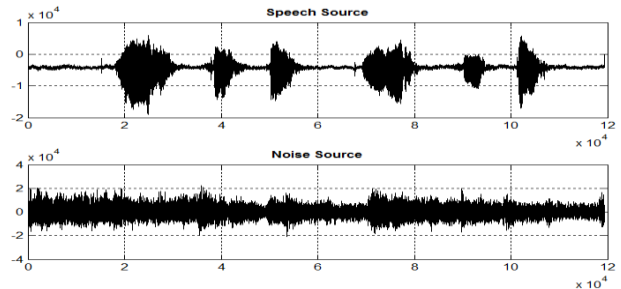


Figura 5. Señales Puras.

Entonces el registro de "Speech mic.", según la Figura 4, se evaluó para los pesos $w_{ps}^s = 0.98, w_{nr}^s = 0.87$, esta combinación corresponde a un sistema altamente ruidoso. En especial nos interesa conocer la respuesta del sistema para diferentes patrones de la señal de ruido de referencia. Los patrones de la señal de ruido de referencia se obtuvieron con los valores de $w_{nr}^n = 0.96$, combinados con valores de w_{nr}^s iguales a (0.00, 0.01, 0.05, 0.10), es decir, se crearon 4 patrones para el ruido de referencia, variando en un amplio rango que va desde el primer patrón que no contiene influencia de la señal de voz, hasta el cuarto que contiene una contaminación del 10% de la señal de voz pura. Finalmente los patrones de señales quedaron almacenadas en archivos WAV, con formato de audio monofónico, 16 bits, y 22.0050 KHz. En la parte superior de la Figura 6 se muestran las dos trazas preparadas, según el modelo LMMSD, para el valor de $w_{nr}^s = 0.05$, junto a los demás fijados, y en la parte inferior se muestra la traza obtenida después del filtrado. Observe la similitud del resultado con la señal primaria de voz "Speech Source".

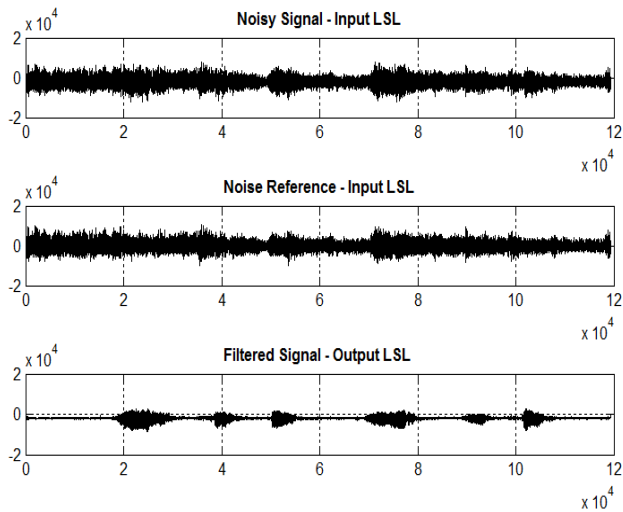


Figura 6. Señales E/S. Algoritmo LSL.

En la tabla 1 se muestran los resultados de RMSE obtenidos con las señales procesadas, utilizando 14 etapas, con ϵ igual a 1×10^{-3} y coeficiente de aprendizaje igual a 0.9999, como se observa los mejores índices son alcanzados cuando el ruido de referencia no contiene información de la señal de voz, y en la misma medida que contiene datos de la señal de voz se degrada la cancelación de ruido. En sentido práctico esto sugiere que los micrófonos destinados al registro del ruido deben captar la menor información posible de la señal de voz primaria.

Tabla 1. Efectividad del Algoritmo de Cancelación.

w_{nr}^s	RMSE
0.00	2319.65
0.01	2341.86
0.05	2445.91
0.10	2609.69

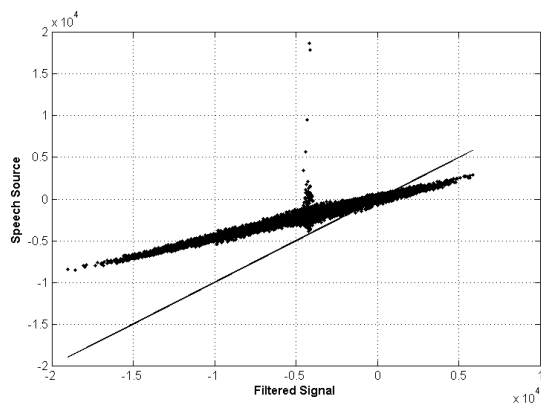


Figura 7. Diagrama de Correlación.

Un análisis más detallado se muestra en la Figura 7, donde se correlacionan las señales. La línea fina corresponde a la correlación de la señal limpia "Speech source" consigo misma, mientras que la línea superior corresponde a la correlación obtenida con la señal filtrada, observe que la mayor dispersión se produce en los valores entorno a la mediana de la señal, además se observa que ambos gráficos no tienen la misma

pendiente, esto induce a pensar que el algoritmo de cancelación introduce cierta distorsión y desplazamiento de los valores temporales.

6. CONCLUSIONES

En este trabajo se ha presentado una implementación totalmente parametrizable del algoritmo LSL, que permite el estudio de diferentes formatos numéricos y evaluar su desempeño acorde a la configuración de los parámetros. Por otra parte, a fin de evaluar la cancelación de ruido se ha presentado el modelo LMMSD. Este modelo permite evaluar cualquier algoritmo de cancelación de ruido con diferentes patrones de señales teóricas, lo que es útil para simular el comportamiento de situaciones reales. Según los resultados obtenidos en los patrones analizados, se debe evitar la contaminación de la señal de ruido de referencia con la señal de voz primaria. Por otra parte, una forma para reducir el error de la cancelación ofrecido por el algoritmo LSL se puede realizar con el post-procesado de la señal filtrada a fin de reducir la dispersión que se observa en la figura 7.

7. AGRADECIMIENTOS

Este trabajo ha sido desarrollado dentro del proyecto CCG06-UPM/INF28, TEC 2006-12887-C02-00, del Plan Nacional de I+D del Ministerio de Educación y Ciencia, y el Proyecto HESPERIA (<http://www.proyecto-hesperia.org>) del Programa CENIT del Ministerio de Industria, España.

8. REFERENCIAS

- [1] F. Albu, J. Kadlec, N. Coleman, A. Fagan, "Pipelined implementations of the a priori error-feedback LSL algorithm using logarithmic arithmetic", IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02), vol. 3, pp. 2681-2684, 13-17 May 2002.
- [2] F. Ling, D. Manolakis, J. G. Proakis, "Numerically robust least-squares lattice-ladder algorithms with direct updating of the reflection coefficients", IEEE Trans. on ASSP, August 1986, vol. 34, no. 4, pp. 837-845.
- [3] John G. Proakis and Dimitris K Manolakis, Digital Signal Processing, Prentice Hall 4th Ed., april 2006.
- [4] S. Haykin. Adaptive filter theory, Prentice Hall, Int., 4th Ed., 2002.
- [5] B. Windrow, et al., "Adaptive noise cancelling: Principles and applications", Proc. IEEE, vol. 63, no. 12, pp. 1692-1975, Dec. 1975.
- [6] Rodellar, V.; Alvarez, A.; Gonzalez, C.; Gomez, P.; Martinez, E., "FPGA Implementation of an Adaptive Noise Canceller for Robust Speech Enhancement Interfaces", 4th Southern Conference on Programmable Logic, 2008, pp. 13 - 18, 26-28 March 2008.
- [7] Y.V. Zakharov, G.P. White, L. Jie, "Low-Complexity RLS Algorithms Using Dichotomous Coordinate Descent Iterations", IEEE Transactions on Signal Processing, vol. 56, no. 7, part 2, pp. 3150 - 3161, July 2008.
- [8] A.H.C. Carezia, P.M.S. Burt, M. Gerken, M.D. Miranda, M.T.M. Da Silva, "A stable and efficient DSP implementation of a LSL algorithm for acoustic echo cancelling", Proceedings. (ICASSP'01). 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001, vol. 2, pp. 921 - 924, 7-11 May 2001.