

# Comparación e Implementación de los Algoritmos ECLMS y OAELMS en un DSP

## Comparison and Implementation on a DSP of ECLMS and OAELMS Algorithm

José Velázquez López, Juan Gerardo Ávalos Ochoa y Juan Carlos Sánchez García

Sección de Estudios de Posgrado e Investigación ESIME Culhuacan, Instituto Politécnico Nacional  
México D.F.

[jvelazquezl@ipn.mx](mailto:jvelazquezl@ipn.mx), [gerardo.avalos@ieee.org](mailto:gerardo.avalos@ieee.org), [jcsanchezgarcia@gmail.com](mailto:jcsanchezgarcia@gmail.com)

*Artículo recibido en Agosto, 2008; aceptado en Enero, 2010*

**Resumen.** En este artículo se presenta dos variantes del algoritmo de adaptación de la media de los mínimos cuadrados, los cuales tienen como objetivo reducir la complejidad computacional y aumentar la velocidad de procesamiento, ya que están basados en la codificación del error para reducir las operaciones de punto flotante. Se presentan los resultados tanto de simulación como de implementación en el DSP, demostrando la factibilidad de aplicarse ambos algoritmos en filtros adaptables digitales.

**Palabras clave:** Error Codificado, Filtros adaptables, Media de los mínimos cuadrados, Predicción lineal, Sobreadaptado..

**Abstract.** In this paper, two variants of the least mean square algorithm are presented, which their objective is to reduce computational complexity and increase the processing speed, due to they are based on error codifying in order to reduce the number of floating point operations. The results are presented both in simulation and implementation on a DSP, they show the factibility to be used in digital adaptive filters.

**Keywords:** Codified error, Adaptive Filters, Least mean square, Linear prediction, Overadapted..

## 1 Introducción

Los sistemas de comunicaciones digitales, presentan a menudo problemas de interferencia intersímbolo, ruido aditivo, eco eléctrico y eco acústico; los cuales se resuelven utilizando filtros adaptables. Los filtros adaptables son sistemas variantes en el tiempo, los cuales tienen la capacidad de ajustar sus parámetros de acuerdo a la variación de los fenómenos que se presentan en su entorno y de manera general presentan una estructura con cuatro terminales como la que se muestra en la Fig. 1, donde la entrada  $x$  corresponde a la señal de entrada a procesar, y es la señal de salida de filtro la cual se tiene que

aproximar a la señal deseada  $d$  y  $e$  es la señal de error igual a la diferencia entre la señal deseada y la señal de salida del filtro, la cual es empleada por el algoritmo para ajustar los parámetros del filtro (Carusone and Johns 2000).

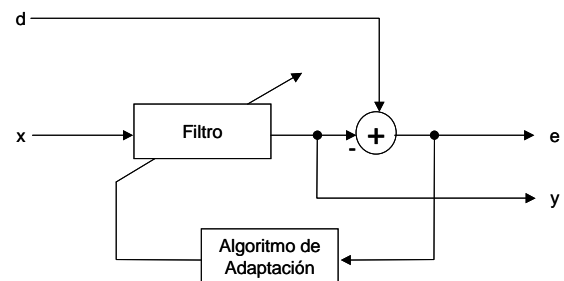


Fig. 1. Estructura General de un filtro adaptable

Los filtros adaptables se pueden dividir según la técnica usada para su diseño, en donde se pueden tener filtros analógicos, los cuales presentan una rápida velocidad de procesamiento, pero su funcionamiento se ve afectado por el desbalanceo (offset) que presentan sus elementos (Pérez et al 2001), el cual afecta de manera considerable al algoritmo de adaptación. Los filtros digitales se caracterizan por tener buena precisión y de rápido desarrollo debido a que se pueden implementar en sistemas programables; pero están limitados en velocidad de procesamiento, debido a que dependen de la arquitectura del sistema digital y además se requiere de uso de operaciones de punto flotante, haciendo más complejo su diseño. Los filtros adaptables mixtos, buscan aprovechar las ventajas que ofrecen los sistemas analógicos y

digitales (Figuroa et al 2004, Tai-Cheng 2001, Kucic et al 2001, Carusone et al 2003).

La complejidad de implementación de filtros adaptables digitales en circuitos integrados, depende del algoritmo de adaptación, el cual es el encargado de realizar la actualización de los parámetros del filtro, en función del error obtenido, y también influye en el diseño del filtro adaptable, ya que depende de la complejidad computacional del algoritmo. Principalmente se manejan los algoritmos de adaptación de la media de los mínimos cuadrados también conocidos como LMS (Least Mean Square) o el algoritmo de mínimos cuadrados recursivos RLS (Recursive Least Square).

## 2 Algoritmos ECLMS y OAECLMS

Un bloque fundamental de los filtros adaptables, es el algoritmo de adaptación, ya que depende de él en gran medida la velocidad de respuesta y la obtención del error mínimo local. El algoritmo LMS (Widrow 2005), es uno de los algoritmos más usados por su baja complejidad computacional y por lo tanto se puede implementar fácilmente en sistemas digitales; como se muestra en la ecuación (1), donde se aprecia que solo depende del valor instantáneo del error  $e(T)$ , el vector  $k$  de pesos  $w_k(T)$  y el vector de datos almacenados en la línea de retardos  $x(T)$  (Farhang 1998).

$$w_k(T+1) = w_k(T) + \mu e(T)x_k(T) \quad (1)$$

En la ecuación (1), se observa la presencia del factor de convergencia  $\mu$ , el cuál es directamente proporcional a la velocidad de convergencia e inversamente proporcional al error mínimo local. La ecuación (2), indica la condición que debe de satisfacer  $\mu$  para evitar la divergencia (Pérez H. et al 2001), en donde la potencia promedio se estima a partir de unas cuantas muestras de la señal de entrada.

$$0 < \mu < \frac{1}{Nx^2(T)} \quad (2)$$

Para reducir la complejidad computacional del algoritmo LMS se han propuesto variantes en donde se emplea solamente el signo de los datos (Farhang 1998), haciendo que la velocidad de convergencia

sea menor debido a la pérdida de información de la magnitud de los datos. Otras variantes emplean factor de convergencia variable aumentando la velocidad de convergencia y la complejidad computacional del algoritmo haciéndolo más difícil de implementar (Nakano et al. 1995). Otra opción ha sido en la actualización parcial de los pesos del filtro, para reducir el número de operaciones por iteración (Godavarti and Cowan 2005). Debido a que se requiere de un algoritmo que permita facilidad de diseño e implementación en sistemas digitales, se propone el algoritmo ECLMS mostrado en la ecuación (3) (Velázquez J. et al 2004); el cual tiene como característica principal la reducción de operaciones de punto flotante al realizar la codificación del error usando números enteros.

$$w_k(T+1) = w_k(T) + \mu C[e(T)]x_k(T) \quad (3)$$

Como se puede ver en (3), con la codificación del error  $C[e(T)]$ , no se altera el algoritmo de adaptación, debido a que la codificación del error se puede realizar por separado. Entonces la adecuación de sistemas existentes no se afecta, ya que solo basta agregar el codificador.

Analizando la ecuación (3), el número de operaciones requeridas para el algoritmo ECLMS se reduce, ya que se puede detectar que el producto entre el error codificado y la muestra de la señal de entrada, es una operación entre valores enteros. En cuanto al factor de convergencia, este tiene una representación de punto flotante, pero se puede simplificar su representación al utilizar una representación en potencia de base dos, lo cual permite emplear un solo bit para representar el valor del factor de convergencia. Entonces, por cada elemento de retardo del filtro se realiza una multiplicación entre valores enteros, complementado de una operación de desplazamiento de punto decimal indicado por la posición que ocupa el bit del factor de convergencia y una suma; con lo cual se puede observar la reducción de operaciones de punto flotante para implementarlo en un sistema digital.

El proceso de codificación del error se realiza dividiendo el error cuantificado entre el valor de la resolución del error  $Res$ . Todo esto se realiza usando la ecuación (4).

$$C[e(T)] = \frac{Q[e(T)]}{\text{Res}} \quad (4)$$

donde  $Q[e(T)]$  es la cuantificación del error y se obtiene tomando la muestra del error y se aproxima al nivel más cercano de un conjunto de niveles ya definidos.

El valor de la resolución está definido por el número de bits  $n$  con el cual se realiza la codificación del error como lo indica la ecuación (5), donde  $Err_{max}$  es el error máximo a codificar con  $n$  bits. Como este valor no se conoce con certeza, se toma a partir de la amplitud máxima de la señal de entrada.

$$\text{Res} = \frac{Err_{max}}{2^n - 1} \quad (5)$$

Sustituyendo el proceso de codificación del error (4) en la ecuación del algoritmo ECLMS se obtiene (6), donde se puede apreciar la modificación indirecta del factor de convergencia, debido a que al aumentar el número de bits, la resolución disminuye y por lo tanto aumentará el factor de convergencia.

$$w_k(T+1) = w_k(T) + \mu \frac{Q[e(T)]}{\text{Res}} x_k(T) \quad (6)$$

Entonces para encontrar la condición que evita la divergencia del algoritmo, se sustituye (4) y (5) en (2) para obtener el valor mostrado en (7).

$$0 < \mu < \frac{Err_{max}}{(2^n - 1)N x^2(T)} \quad (7)$$

Una variante aplicada al algoritmo LMS para aumentar la velocidad de convergencia y reducir el error cuadrático medio (MSE - Mean Square Error), es aplicando el rehúso de datos (DR-LMS - Date Reusing Least Mean Square) (Roy and Shynk 1989), el cual consiste en actualizar el valor de los pesos en  $j$  iteraciones adicionales utilizando los mismos datos almacenados en la línea de retardos del filtro; tal como lo indica (8).

$$w_k(T+j+1) = w_k(T+j) + \mu e(T+j) x_k(T) \quad (8)$$

El algoritmo DR-LMS se puede emplear cuando la velocidad de procesamiento es grande, además está opción del rehúso de datos presenta las mismas características que el algoritmo LMS normalizado (NLMS - Normalized Least Mean Square), con la ventaja de que DR-LMS no requiere de divisiones y además de que es capaz de responder a valores pequeños de señal de entrada.

Entonces combinando el algoritmo DR-LMS y el algoritmo ECLMS para aprovechar sus ventajas de velocidad de convergencia, minimización del error mínimo local y reducción de operaciones de punto flotante, se obtiene el algoritmo OAECLMS mostrado en la ecuación (9).

$$w_k(T+j+1) = w_k(T+j) + \mu C[e(T+j)] x_k(T) \quad (9)$$

## 2.1 Resultados de Simulación de los algoritmos ECLMS y OAECLMS.

En este apartado, se presentan los resultados de simulación obtenidos con Matlab™ de los algoritmos ECLMS y OAECLMS, tomando como referencia los algoritmos LMS y NLMS, empleando las mismas condiciones de factor de convergencia  $\mu = 1 \times 10^{-5}$ , señal de entrada compuesta por la suma de tonos simples a diferente frecuencia y fase, y una relación señal a ruido de 30 dB para observar su comportamiento a perturbaciones de la señal de entrada. Además todos los resultados mostrados se obtuvieron utilizando un filtro adaptable como predictor lineal. Es importante hacer notar que se tomó el error cuadrático promedio entre la señal de salida y la señal deseada, con el objetivo de apreciar con mayor facilidad la aproximación del filtro adaptable a la respuesta de la señal deseada.

En la Fig. 2 se presenta la curva del error cuadrático medio (MSE), resultado de la comparación entre el algoritmo LMS y ECLMS cuando se codifica con 4 y 8 bits, comprobando que el factor de convergencia se altera de manera indirecta con la codificación, al obtenerse una velocidad de convergencia mayor con el algoritmo ECLMS. Además se puede apreciar que el número de bits usados en la codificación del error es directamente proporcional a la velocidad de convergencia.

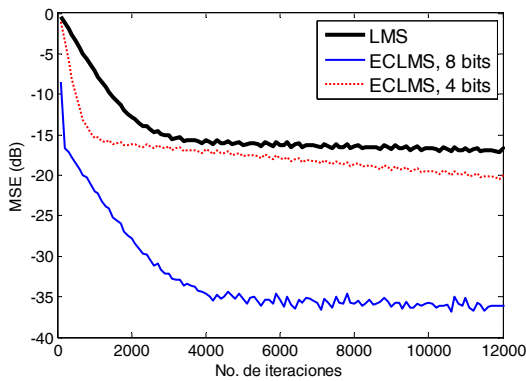


Fig. 2. Comparación de los algoritmos LMS y ECLMS

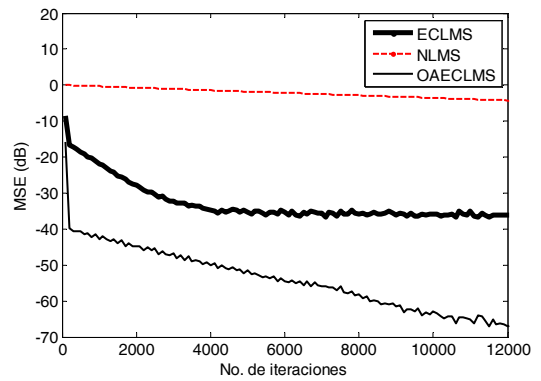


Fig. 3. MSE comparativo entre OAECLMS, NLMS y ECLMS

Otro aspecto importante a tomar en cuenta en el diseño de filtros adaptables digitales usando el algoritmo ECLMS u OAECLMS, es el hecho de que al ir avanzando el proceso, la magnitud del error irá disminuyendo de tal manera que cuando se encuentre en un valor ubicado entre 1.5 y 0.5 veces el valor de la resolución, entonces el algoritmo ECLMS tendrá un comportamiento similar al algoritmo LMS de signo, debido a que el error codificado tomará el signo del error. Además, cuando el error es menor a la mitad del valor de la resolución, el error codificado será igual a cero, haciendo que el valor de los pesos del filtro se mantenga constante evitando así realizar operaciones de actualización de pesos.

En la Fig. 3 se muestran los resultados obtenidos al emplear el algoritmo OAECLMS, comparando su curva de convergencia con el algoritmo LMS normalizado (NLMS) y el algoritmo ECLMS, empleando el mismo factor de convergencia y ocho bits en la codificación del error.

Se puede notar que el algoritmo OAECLMS presenta una mejor velocidad de convergencia, además de un error cuadrático mínimo menor a los algoritmos ECLMS y OAECLMS.

Para verificar la cercanía del comportamiento entre el algoritmo NLMS y DRLMS, se realizó la simulación para observar su curva del MSE empleando diferentes factores de convergencia, obteniéndose los resultados mostrados en la Fig. 4, donde se aprecia claramente que al aumentar el factor de convergencia, la magnitud del error cuadrático mínimo disminuye, además de que se alcanza en un menor número de iteraciones, lo cual es una característica del algoritmo NLMS.

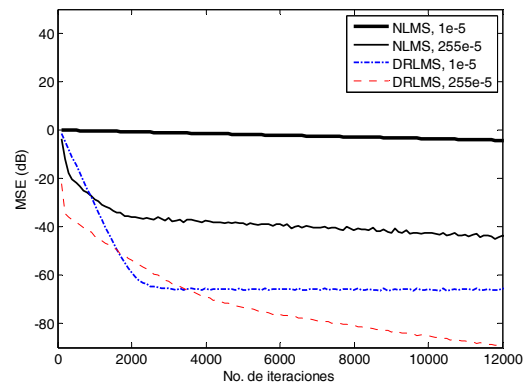


Fig. 4. Comportamiento entre algoritmo NLMS y DRLMS

La similitud entre el algoritmo DR-LMS y NLMS, se puede aprovechar para emplearlo en el algoritmo OAECLMS, en donde al combinar la ventaja de poder emplear un factor de convergencia mayor y la reducción de operaciones de punto flotante, entonces la complejidad computacional se puede reducir si se piensa en la implementación de filtros adaptables digitales en circuitos integrados. En la Fig. 5 se observa el comportamiento del algoritmo OAECLMS cuando se emplean diferentes valores en el factor de convergencia y utilizando ciclos de sobre-adaptación diferentes.

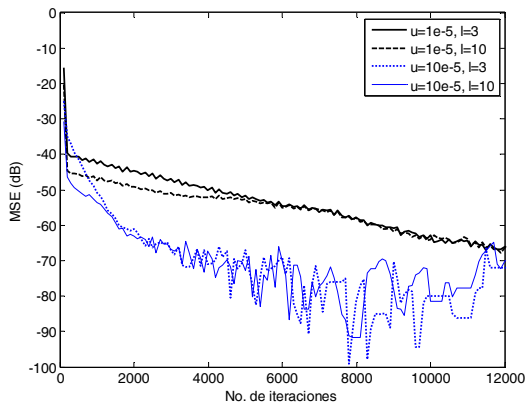


Fig. 5. Curva de convergencia para algoritmo OAECLMS

Los resultados de la Fig. 5, demuestran que al aumentar el factor de convergencia, se logra mejorar el error cuadrático medio mínimo, además de que el aumento del número de ciclos de sobre-adaptación, no logra mejorar de manera significativa la velocidad de convergencia o el error cuadrático medio mínimo, por lo cual se puede deducir que con un número pequeño de ciclos de sobre-adaptación, es suficiente para poder obtener una respuesta óptima.

Para observar el comportamiento del algoritmo ECLMS y OAECLMS, se realizaron las pruebas para medir el MSE empleando un archivo de audio con la frase: "ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA DEL INSTITUTO POLITÉCNICO NACIONAL", con una frecuencia de muestreo de 8 Khz, cuya gráfica en el dominio del tiempo se muestra en la Fig. 6, en donde se pueden ver los silencios entre cada palabra, lo cual permite observar el comportamiento a cambios de la señal.

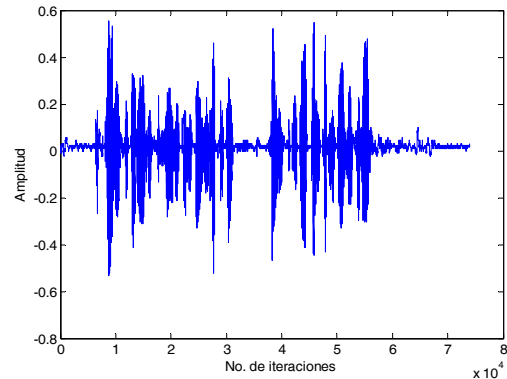


Fig. 6. Gráfica de la señal de voz en el dominio del tiempo

Estas simulaciones se realizaron con la finalidad de tener una aproximación de la respuesta de los algoritmos propuestos, cuando trabajan en el DSP (Digital Signal Processor). En la Fig. 7, se presenta la comparación del MSE entre el algoritmo ECLMS y OAECLMS.

En las gráficas de la Fig. 7, se observa la capacidad de respuesta de los algoritmos de error codificado, a adaptarse a los cambios de la señal cuando se presentan silencios en la frase, comprobando así su efectividad y posibilidad de implementación en aplicaciones donde se requiere respuesta en tiempo real. Además al aumentar el factor de convergencia en el algoritmo OAECLMS, se logra reducir más el error cuadrático medio.

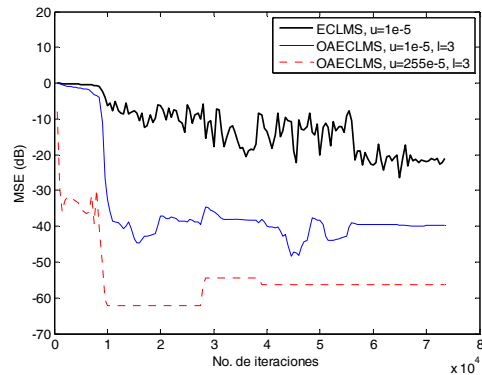


Fig. 7. MSE para ECLMS y OAECLMS con señal de voz

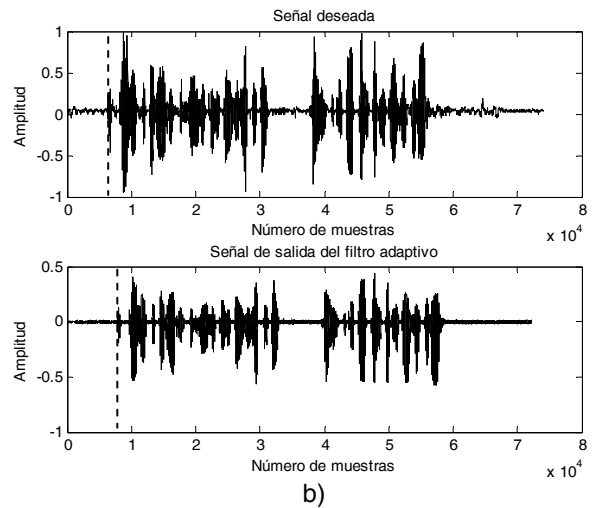
### 3 Implementación de los algoritmos en el DSP TMS320C6713

Para la implementación de los algoritmos se utilizó el kit de desarrollo DSK TMS320C6713 (Developer Starter Kit), el cual es una tarjeta que permite desarrollar diversas aplicaciones con los procesadores digitales de señales de Texas Instruments™. El kit de desarrollo empleado, contiene un DSP TMS320C6713 que opera a 225 MHz, una memoria externa de 16 Mbytes, memoria flash de 512 Kbytes, un codificador-decodificador (Codec) estéreo con dos entradas y dos salidas analógicas, un dispositivo lógico programable (CPLD – Complex Programmable Logic Device), cuatro leds y cuatro dips switch programables, interfaz para memoria externa de 32 bits a 100 MHz, interfaz para expansión de periféricos y soporte de emulación JTAG por USB. El procesador C6713 cuenta con ocho unidades funcionales o de ejecución compuestas por seis unidades lógico aritméticas (ALUs) y dos unidades multiplicadoras, con las que puede realizar hasta 1350 millones de operaciones de punto flotante por segundo (MFLOPS) y 450 millones de multiacumulaciones por segundo (MMACS), y está basado en la arquitectura VLIW (Very Long Instruction Word – Palabra de Instrucción Muy Larga), este tipo de arquitecturas se caracterizan por tener muchas unidades funcionales que operan en paralelo, permitiendo de esa forma ejecutar varias instrucciones en un mismo ciclo de reloj.

Para las pruebas de funcionamiento de los algoritmos se utilizó la estructura adaptiva de predictor lineal, empleando la señal de audio mostrada en la Fig. 6, con un factor de convergencia  $\mu=1 \times 10^{-9}$  y 5 coeficientes. Haciendo uso de MATLAB™, se obtuvo la forma de onda en el tiempo mostrado en la Fig. 8 y los espectrogramas de la señal deseada y la señal de salida del predictor lineal usando el algoritmo LMS en la Fig. 9.

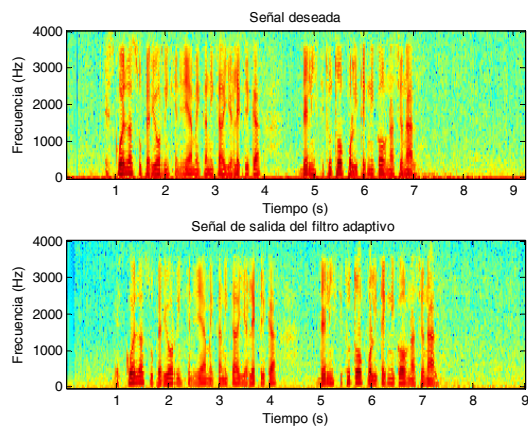
El espectrograma permite comparar en el dominio del tiempo y de la frecuencia de manera simultánea, por lo que es posible tener una mejor perspectiva del comportamiento en ambos dominios.

En la Fig. 8 está marcado con una línea el punto de referencia que se tomó para obtener el tiempo de retraso que existe en la señal de salida del predictor lineal con respecto a la señal de referencia. Como se puede observar el retraso producido por el procesador es mínimo.



**Fig. 8.** a) Señales de voz de entrada y salida del predictor lineal usando el algoritmo LMS. b) Espectrogramas de la señal de entrada y la señal de salida del predictor lineal usando el algoritmo LMS.

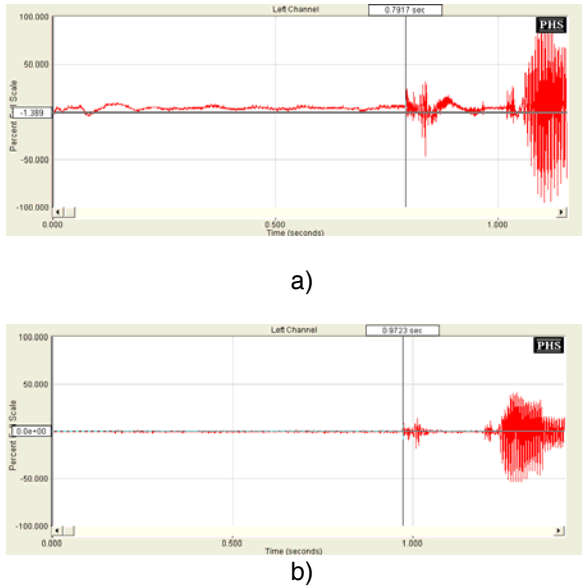
En la Fig. 9, se muestran los espectrogramas de la señal entrada y salida del filtro, en donde se observa que la señal de salida del filtro adaptable presenta algunas diferencias en las componentes de alta frecuencia, sin embargo las componentes de baja frecuencia no presentan diferencias notables.



**Fig. 9.** Espectrogramas de la señal de entrada y la señal de salida del predictor lineal usando el algoritmo LMS

La Fig. 10 muestra el acercamiento que se hizo al punto de referencia de la señal deseada, en el cual se registró un tiempo de 791.7 ms. De igual

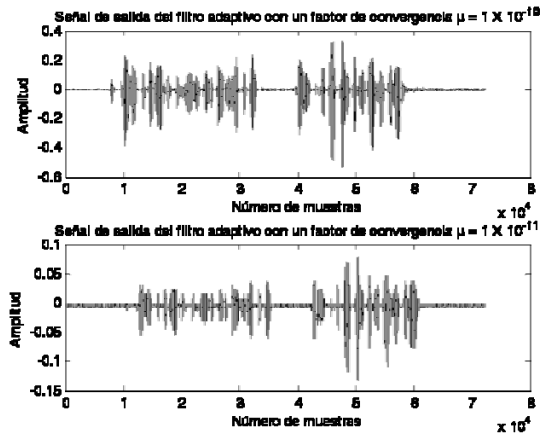
forma se realizó un acercamiento a la señal de salida del filtro adaptable, en la que se obtuvo un tiempo de 972.3 ms, por lo tanto el retraso generado con respecto a la señal original es de 180.6 ms.



**Fig. 10.** a) Acercamiento de la señal deseada.  
b) Acercamiento a la señal de salida del predictor lineal usando el algoritmo LMS.

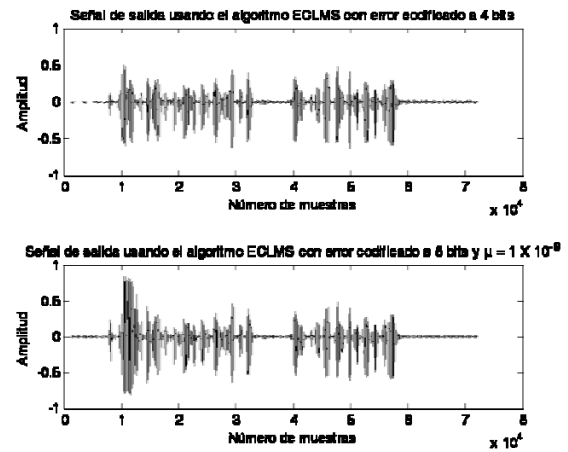
El algoritmo LMS también se probó con diferentes valores del factor de convergencia. En la Fig. 11 se muestran los resultados que se obtuvieron al usar un factor de convergencia  $\mu=1 \times 10^{-10}$  y  $\mu=1 \times 10^{-11}$ .

Como se puede observar al utilizar un valor de  $\mu=1 \times 10^{-11}$  el sistema ya no funciona de manera adecuada debido a que tarda en adaptarse a los cambios en la estadística de la señal, esto nos indica que al disminuir más el valor del factor de convergencia el sistema tarda en adaptarse y por lo tanto ya no se obtiene respuesta a la salida del filtro, además, se hicieron pruebas aumentando el valor de  $\mu$  en donde se observó que al hacerlo mayor a  $1 \times 10^{-7}$  el sistema diverge [J. G. Avalos et al; Kertamavaz Nasser], por lo tanto se encontró que los valores óptimos con los cuales trabaja el algoritmo LMS son con los factores de convergencia que se encuentran entre  $1 \times 10^{-7}$  y  $1 \times 10^{-9}$ .



**Fig. 11.** Señales de salida del predictor lineal usando el algoritmo LMS con diferentes factores de convergencia.

Las pruebas realizadas al algoritmo ECLMS se realizaron con error codificado a 4 y 8 bits. En la Fig. 12 se observa las formas de onda en el tiempo de la señal de salida del predictor lineal con cinco coeficientes, usando el ECLMS con error codificado a 4 y 8 bits y un factor de convergencia de  $1 \times 10^{-9}$ .



**Fig. 12.** Señales de salida del predictor lineal usando el algoritmo ECLMS con error codificado a 4 y 8 bits

En los espectrogramas obtenidos de la Fig. 13, se puede observar algunas diferencias en las altas frecuencias, sin embargo, las componentes en frecuencia de la señal obtenida con el algoritmo ECLMS codificado a 4 bits presentan una mayor similitud a los de la señal deseada que las

componentes de la señal obtenida con el algoritmo LMS.

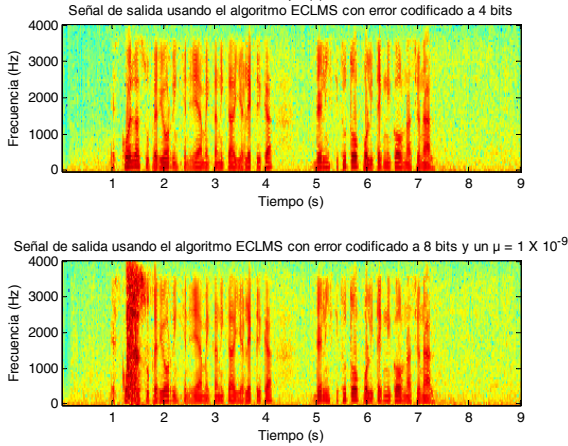


Fig. 13. Espectrogramas de la señal de salida del predictor lineal usando el algoritmo ECLMS con error codificado a 4 y 8 bits

Se realizó un acercamiento a las señales de salida del filtro adaptable, el tiempo registrado con el algoritmo ECLMS codificado a 4 bits fue de 970.3 ms, mientras que cuando se codificó el error a 8 bits se obtuvo un tiempo de 961.5 ms, sin embargo, aunque la respuesta es más rápida se puede observar que en las primeras muestras la señal está distorsionada, debido a que el incremento de bits provoca que el factor de convergencia aumente, lo que ocasiona que el error sea mayor. En las pruebas realizadas se observó que para evitar la divergencia se debe utilizar valores más pequeños que  $\mu=1 \times 10^{-9}$  Fig. 14.

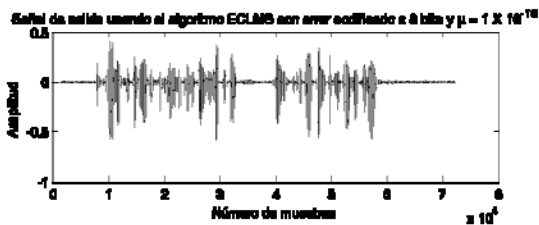


Fig. 14. Señal de salida del predictor lineal usando algoritmo ECLMS con error codificado a 8 bits y  $\mu = 1 \times 10^{-10}$

En la Fig. 15 se muestra la señal de salida del filtro adaptable usando los algoritmos ECLMS y

OAELMS con tres iteraciones adicionales, usando un factor de convergencia  $\mu=1 \times 10^{-7}$  y el error está codificado a 4 bits para ambos algoritmos. Como se puede observar el algoritmo LMS diverge, mientras que el algoritmo OAELMS alcanza a converger, debido a que el re-cálculo de los pesos ocasiona la reducción de la potencia del error [J. G. Avalos et al 2007]. En el espectrograma de la señal obtenida con el algoritmo OAELMS se puede observar que las componentes de baja frecuencia permanecen intactas, solamente se observan algunas componentes adicionales en las frecuencias altas.

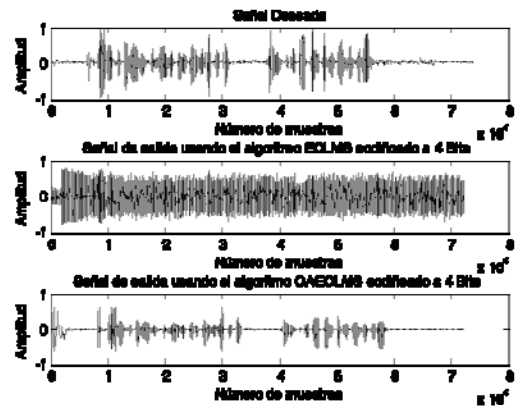


Fig. 15. Formas de onda en el tiempo de señales de entrada y salida del predictor lineal, usando ECLMS y OAELMS, con error codificado a 4 bits.

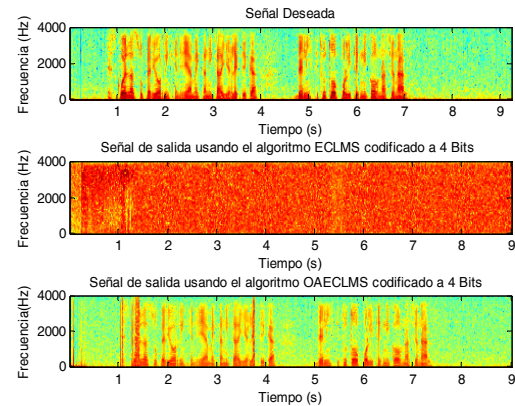


Fig. 16. Espectrogramas de las señales de entrada y salida del predictor lineal, usando los algoritmos ECLMS y OAELMS con error codificado a 4bits



Para observar la diferencia en las velocidades de convergencia se realizó un acercamiento a la señal generada con el algoritmo OAELMS, en la cual se registró un tiempo de 1.0339 s.

En la Fig. 17 se presentan las formas de onda en el tiempo obtenidas con los algoritmos ECLMS y OAELMS sobre-adaptado con tres iteraciones adicionales, con un factor de convergencia  $\mu=1 \times 10^{-7}$  y el error está codificado a 8 bits para ambos algoritmos. En este caso el algoritmo ECLMS de nuevo no alcanza a converger, mientras que el algoritmo OAELMS aumenta su velocidad de convergencia ya que se utilizó un número mayor de bits para la codificación del error.

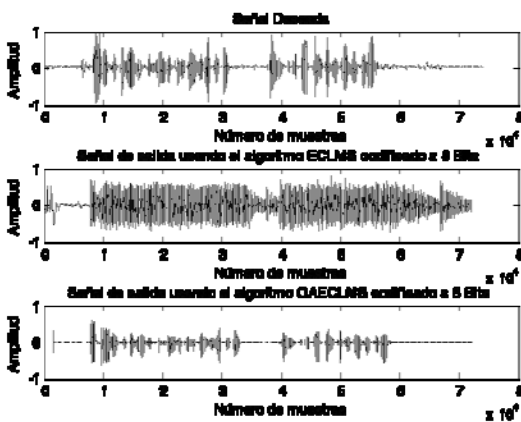


Fig. 17. Formas de onda en el tiempo de señales de entrada y salida del predictor lineal, usando ECLMS y OAELMS, con error codificado a 8 bits

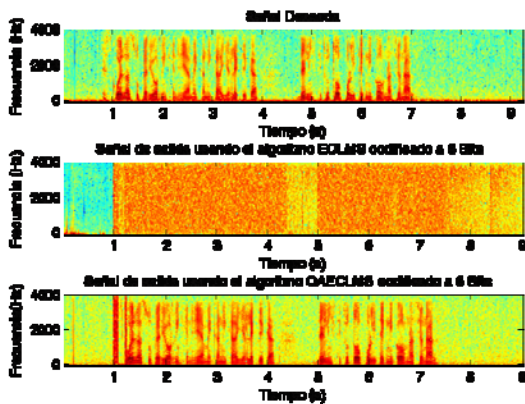


Fig. 18. Espectrogramas de las señales de entrada y salida del predictor lineal, usando los algoritmos ECLMS y OAELMS con error codificado a 4bits

Al realizar un acercamiento a la señal generada con el algoritmo OAELMS se registró un tiempo de 984.7 ms, es decir es 49.2 ms más rápida que la señal obtenida con el algoritmo OAELMS con error codificado a 4 bits.

Tabla 1. Tabla comparativa de tiempo de respuesta y ciclos de reloj

Algoritmo	$\mu$	$n$	Tiempo de respuesta	Ciclos de reloj
LMS	$1 \times 10^{-8}$	-	970.2ms	76
ECLMS	$1 \times 10^{-8}$	4	968.5ms	71
LMS	$1 \times 10^{-9}$	-	972.3 ms	76
ECLMS	$1 \times 10^{-9}$	4	970.3 ms	71
ECLMS	$1 \times 10^{-9}$	8	961.5 ms	71
ECLMS	$1 \times 10^{-10}$	-	975.7 ms	76
ECLMS	$1 \times 10^{-10}$	4	973 ms	71
ECLMS	$1 \times 10^{-10}$	8	969.2 ms	71
ECLMS	$1 \times 10^{-11}$	-	1.5726 s	76
ECLMS	$1 \times 10^{-11}$	4	971.8 ms	71
ECLMS	$1 \times 10^{-11}$	8	969.6 ms	71

En la Tabla 1, se muestran la comparación de tiempo de convergencia y el número de ciclos de reloj en los que se ejecutan los algoritmos LMS, ECLMS y OAELMS. Como se puede observar los algoritmos ECLMS y OAELMS se ejecutan en un menor número de ciclos debido a la reducción de operaciones de punto flotante, por tal motivo su tiempo de convergencia es menor que el del algoritmo LMS, además, al hacer más pequeño el factor de convergencia los algoritmos LMS y ECLMS presentan un incremento en el tiempo de convergencia, mientras que el OAELMS presenta un tiempo menor. También se puede observar que con un mayor número de bits en la codificación del error el tiempo de convergencia es menor, sin embargo, se debe tener cuidado al elegir el número de bits ya que un valor muy grande provocará que el algoritmo diverja.

### 4 Conclusiones

Los resultados obtenidos tanto en simulación como en implementación en el DSP, demuestran las ventajas del algoritmo ECLMS en reducción de operaciones de punto flotante y mejora en la velocidad de convergencia, en donde se puede observar que la modificación propuesta, no altera la estructura del filtro adaptable teniendo la

versatilidad de ser implementado con algunas otras variantes del algoritmo LMS como lo es el DR-LMS.

A través de las formas de onda en el tiempo y los espectrogramas se comprobó que el algoritmo ECLMS reduce el número de operaciones de punto flotante, por lo tanto el número de ciclos de instrucción que se ejecutan es menor provocando que la velocidad de procesamiento sea mayor, con lo cual se puede ver la factibilidad de implementación del algoritmo en un arreglo lógico programable, con una mejor complejidad de diseño.

Se ha comprobado que el algoritmo ECLMS reduce el tiempo de procesamiento, esto nos da la posibilidad de realizar la sobre adaptación o re-uso de datos, la cual es una técnica que permite recalcular el valor de los pesos en cada iteración produciendo una reducción en el valor mínimo del error cuadrático medio.

Con los resultados obtenidos se pudo comprobar que al aplicar el algoritmo OAECLMS en la configuración de predictor lineal con señales de audio, se puede usar valores más grandes del factor de convergencia sin que el algoritmo diverja lo cual produce un aumento en la velocidad de convergencia.

Además, la simplificación de la aritmética se logra con la codificación del error, debido a que solo se trabaja con números enteros, reduciendo así la complejidad de diseño con sistemas lógicos programables e incluso, realizar la implementación de un filtro adaptivo digital en un circuito integrado.

## Referencias

- Avalos J. G. González J. M., Velázquez J., Sánchez J. C. (2007).** Implementación del algoritmo LMS en el DSP TMS320C6713. *XXIX Congreso Internacional de Ingeniería Electrónica*, Chihuahua, México. 169-173.
- Avalos J. G. González J. M., Velázquez J., Sánchez J. C. (2007).** Implementación del Algoritmo LMS con Error Codificado en el DSP TMS320C6713, *Memoria digital del 10º Congreso Nacional de Ingeniería Electromecánica y de Sistemas*, D.F., México.
- Carusone A. C., Johns D. A. (2003).** Digital LMS Adaptation of Analog Filters Without Gradient Information. *IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing*, 50(9), 539-552.
- Carusone A., Johns D. A. (2000).** Analogue adaptive filters: past and present. *IEE Proceedings. Circuits, Devices and Systems*, 147(1), 82-90.
- Figuroa Miguel, Bridges Seth, Hsu David, and Diorio Chris (2004).** A 19.2 GOPS Mixed-Signal Filter With Floating-Gate Adaptation. *IEEE Journal of Solid State Circuits*, 39(7), 1196-1201.
- Kehtarnavaz Nasser (2005).** *Real – Time Digital Signal Processing Based on the TMS320C6000*. Amsterdam: Elsevier.
- Kucic M, Hasler P., Dugger J., Anderson D (2001).** Programmable and Adaptive Analog Filters using Arrays of Floating-Gate Circuits. *Conference on Advanced Research in VLSI*, Salt Lake City, Utah USA. 148-162.
- M. Godavarti and C. F. N. Cowan (2005).** Partial Update LMS Algorithms. *IEEE Transactions on Signal Processing*, 53(7), 2382-2389.
- Nakano Mariko, Perez Héctor, Niño de Rivera Luis, Casco Fausto y Sánchez Juan Carlos (1995).** A Time Varying Step Size NLMS Algorithm For Adaptive Echo Canceller Structures. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science*, E78(2), 254-258.
- Pérez Héctor, Nakano Mariko, Niño de Rivera Luis (2001).** Desarrollos recientes en sistemas adaptivos analógicos. *Revista Científica*, 5(3), 113-129.
- Roy S. y Shynk J. J. (1989).** **Analysis of the Data-Reusing LMS Algorithm.** *Proceedings of the 32nd Midwest Symposium on Circuits and Systems*, Champaign, USA, 1127-1130.
- Sayed Ali H (2003).** *Fundamentals of Adaptive Filtering*, New York: John Wiley & Sons 233-238.
- Shivaling S. Mahant-Shetti, Hosur S. y Gatherer A. (1997).** The Log-Log LMS Algorithm. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Munich, Germany, -2357-2360.
- Tai-Cheng Lee and Bezhad Razavi (2001).** A 125 Mixed-Signal Echo Canceller for Gigabit Ethernet on Copper Wire. *IEEE Journal of Solid-State circuits*, 36(3), 366-373.
- Velázquez J., Sánchez J. C., Ávalos J. G. (2008).** Algoritmo LMS con Error Codificado para Filtrado Adaptivo. *Información Tecnológica*, 19(5). 85-96
- Widrow B. (2005).** Thinking About Thinking: The Discovery of the LMS Algorithm. *IEEE Signal Processing Magazine*, 22(1), 100-106.



**José Velázquez López**

*Obtuvo el grado de Ingeniero en Comunicaciones y Electrónica en la Escuela Superior de Ingeniería en Microelectrónica en el 2000, el grado de Maestro en Ciencias en Ingeniería en Microelectrónica y Doctor en Ciencias en Comunicaciones y Electrónica en la Sección de Graduados e Investigación de la ESIME Culhuacan del IPN México en, 2002 y 2007 respectivamente. Actualmente es profesor Titular del departamento de Ingeniería en Computación en la ESIME Culhuacan.*



**Juan Gerardo Ávalos Ochoa**

*Recibió el grado de Ingeniero en Comunicaciones y Electrónica en el 2006 y el grado de Maestro en Ciencias en el 2008, en la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacan del Instituto Politécnico Nacional. Actualmente está realizando sus estudios de Doctorado en Comunicaciones y Electrónica, en la ESIME Culhuacan. Su área de interés es el procesamiento digital de señales.*



**Juan Carlos Sánchez García**

*Recibió el grado de Ingeniero en Electrónica y el Grado de Doctor en Ciencias en la Universidad Autónoma Metropolitana, de la Ciudad de México en 1987 y 2000, respectivamente. De 1987 a 1997 fue profesor en la Universidad Autónoma Metropolitana. Desde 1997 es profesor investigador en la Sección de Estudios de Posgrado en Investigación, en la ESIME Culhuacan, del Instituto Politécnico Nacional (IPN) de México.*