

Implementación de un cancelador activo de ruido en un ducto usando el algoritmo FX-ECLMS

E. Pichardo, X. Maya, A. A. Vázquez, J. G. Avalos, J. C. Sánchez

Instituto Politécnico Nacional,
Sección de Estudios de Posgrado e Investigación,
ESIME CULHUACAN,
México

edua_pim95@hotmail.es, xmayar1300@alumno.ipn.mx,
avazquezp1000@alumno.ipn.mx, javaloso@ipn.mx, jcsanchezgarcia@gmail.com

Resumen. En este artículo se presenta la implementación de un sistema de cancelación activa de ruido en un ducto usando el algoritmo de mínimos cuadrados promediados con error codificado (Error Coded Least Mean Square-ECLMS). Para reducir los efectos producidos por la trayectoria secundaria se utilizó la estructura de filtrado-X. El sistema fue implementado en el procesador digital de señales TMS320C6713 de Texas Instruments. Los resultados experimentales demuestran que al usar el algoritmo ECLMS se logra cancelar de manera adecuada señales de baja frecuencia, además, la carga computacional se reduce considerablemente.

Palabras clave: control activo de ruido, filtrado adaptativo, algoritmo ECLMS.

Implementation of Active Noise Control in a Duct Using the FX-ECLMS Algorithm

Abstract. This paper deals with the implementation of an active noise control system in a duct using the error coded least mean square (ECLMS) algorithm. The effects caused by the secondary path were reduced using a filter-x structure. The system was implemented in the digital signal processor TMS320C6713 from Texas Instruments. The experimental results show that using the ECLMS algorithm the low frequency signals are effectively cancelled, moreover, the computational cost is considerably reduced.

Keywords. Active noise control, adaptive filter, ECLMS algorithm.

1. Introducción

La Cancelación Activa de Ruido (CAR) se basa en el principio de superposición, el cual consiste en utilizar una fuente canceladora de ruido que se encarga de atenuar la

fuente primaria de ruido, esto se logra produciendo una señal anti-ruido que es igual en amplitud y frecuencia a la señal de ruido, pero con fase contraria. Para generar la señal anti-ruido es necesario utilizar técnicas de filtrado adaptativo debido a que las características de la señal como el contenido espectral, la amplitud y la fase, así como el medio y la velocidad de propagación no son estacionarios [1].

La Fig. 1 muestra la estructura general de los filtros adaptativos, en donde $x(n)$ es una señal de entrada, $d(n)$ es la señal deseada, $y(n)$ es la señal de salida del filtro adaptativo y $e(n)$ es la señal de error necesaria para adaptar los parámetros del filtro, la cual se calcula mediante la diferencia entre $y(n)$ y $d(n)$ [2].

La principal característica de los filtros adaptativos es que son variantes en el tiempo debido a que es necesario compensar los cambios de la señal de entrada para obtener la señal de salida deseada. Para poder compensar estos cambios se utilizan algoritmos adaptativos, los cuales se encargan de actualizar los coeficientes del filtro adaptativo.

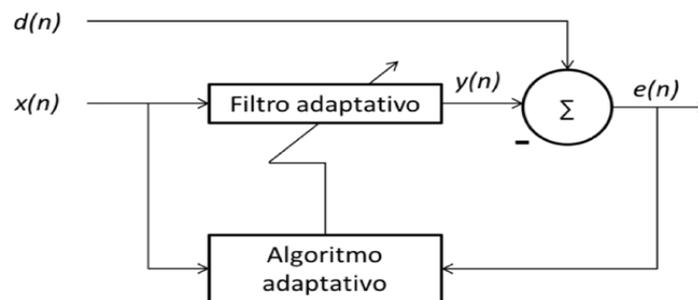


Fig. 1. Estructura general de filtro adaptativo.

Existe una gran variedad de algoritmos que se pueden clasificar en algoritmos de baja complejidad computacional con baja velocidad de convergencia y algoritmos de alta velocidad de convergencia con un alto coste computacional. Algunos de los algoritmos más utilizados por su baja complejidad computacional son el algoritmo de mínimos cuadrados promediados (Least Mean Square-LMS) y su versión normalizada (Normalized Least Mean Square-NLMS). Dichos algoritmos han sido implementados con éxito en diferentes trabajos de sistemas CAR [3-6], sin embargo, como se mencionó anteriormente la velocidad de convergencia de los algoritmos utilizados suele ser lenta. En [7] se presenta un sistema CAR en el que utiliza el algoritmo de proyecciones afines (Affine Projection-AP), el cual presenta una velocidad de convergencia más rápida que la de los algoritmos LMS, sin embargo, su complejidad computacional es bastante elevada ya que para el cálculo de coeficientes se requiere realizar diversas operaciones de inversión de matrices, esto provoca que su implementación en hardware sea muy complicada. Por lo antes mencionado, es evidente que el reto en el desarrollo de los sistemas CAR se encuentra en el uso de un algoritmo que pueda ser de alta velocidad de convergencia y que su complejidad computacional sea baja.

En [8] se presentó el algoritmo de mínimos cuadrados promediados con error codificado (ECLMS), el cual es una variante que presenta una alta velocidad de convergencia y una baja complejidad computacional con respecto al algoritmo LMS, sin embargo,

aún no se han reportado trabajos en los cuales se aplique a sistemas CAR. En este trabajo presentamos una versión del algoritmo ECLMS con filtrado-X (FX-ECLMS) para sistemas CAR, con la finalidad de realizar un sistema que mantenga un equilibrio entre la velocidad de convergencia y su complejidad computacional. Los resultados experimentales demuestran que el algoritmo cancela de manera adecuada el ruido de bajas frecuencias dentro de un ducto, y lo más importante es que la carga computacional se reduce considerablemente.

2. Algoritmo ECLMS

El algoritmo ECLMS reduce las operaciones de punto flotante mediante el proceso de codificación del error, de esta forma se reduce la complejidad computacional y se modifica el factor de convergencia con la finalidad de aumentar la velocidad de convergencia.

La ecuación de actualización de coeficientes $w(n)$ del algoritmo se muestra en (1), donde se observa que el producto entre el error codificado, $C[e(n)]$, y la muestra de la señal de entrada, $x(n)$, es una operación entre valores enteros, por lo tanto, el número de operaciones de punto flotante se reduce:

$$w(n + 1) = w(n) + \mu C[e(n)]x(n). \quad (1)$$

La codificación del error se realiza efectuando un proceso de cuantización en la señal de error y posteriormente asignando un código a la muestra cuantizada. Este proceso se realiza con la ecuación (2):

$$C[e(n)] = \text{round} \left(\frac{Q[e(n)]}{\text{Res}} \right). \quad (2)$$

Siendo $Q[e(n)]$ la cuantización del error, Res la resolución del codificador y round una función que redondea el resultado obtenido de la división entre ambos, la resolución es calculada con (3):

$$\text{Res} = \frac{e_{\max}}{2^b - 1}, \quad (3)$$

donde e_{\max} es el error máximo probable y b el número de bits utilizados para codificar la señal. Debido a que se desconoce el error máximo, se hace una estimación a partir de la amplitud máxima de la señal deseada o de la señal de entrada del filtro adaptativo [8]. En el algoritmo ECLMS, conforme el proceso de adaptación avanza la señal de error es minimizada hasta que la función round ajusta los valores a cero o uno comportándose como los algoritmos de signo, por lo tanto, si la señal de error se mantiene en esos valores no es necesario actualizar los coeficientes del filtro, dicho proceso se muestra en (4) [9].

$$w(n + 1) = \begin{cases} w(n) & \text{si } C[e(n)] = 0 \text{ o } 1 \\ w(n) + \mu C[e(n)]x(n) & \text{si } C[e(n)] \neq 0 \text{ o } 1 \end{cases} \quad (4)$$

3. Estructura FX-ECLMS

El concepto básico de la cancelación activa de ruido consiste en procesar dos señales provenientes una de un sensor de referencia ubicado cerca de la fuente de ruido y otra de un sensor primario ubicado cerca de la fuente secundaria, es decir la fuente que produce la señal anti-ruido. Posteriormente estas señales son procesadas mediante un algoritmo adaptativo el cual usa la señal del micrófono de referencia para estimar la señal anti ruido.

En un escenario ideal, la Cancelación Activa de Ruido se puede considerar un problema de identificación de sistemas. El sistema a identificar es llamado trayectoria primaria $P(z)$, el cual está conformado por la respuesta acústica entre el micrófono de referencia y el micrófono de error. Sin embargo, un sistema CAR está conformado por varios sistemas cuya función de transferencia debe ser tomada en cuenta.

Estos sistemas incluyen micrófonos y sus preamplificadores, bocinas, amplificadores, filtros pasa bajas, convertidores A/D y D/A y la trayectoria entre la bocina que produce el anti-ruido y el micrófono que se encarga de monitorear la señal de error. El efecto acumulado de todos estos sistemas es denominado trayectoria secundaria $S(z)$.

La trayectoria secundaria provoca que la señal de error no está alineada en tiempo con la señal de referencia, lo que produce inestabilidad en el sistema. Para evitar este efecto se usa la estructura de filtrado-X. El método consiste en ubicar un sistema con una respuesta idéntica a la de $S(z)$, el cual es denominado $S'(z)$, de manera que la señal de referencia se filtrará en este sistema y su señal de salida servirá para realizar el cálculo de los coeficientes del algoritmo como se muestra en la Fig. 2 [1].

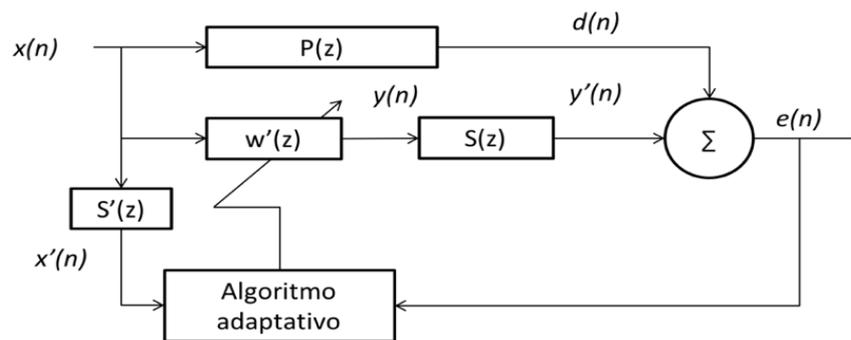


Fig. 2. Estructura de Filtrado-x.

De esta manera la señal de entrada al algoritmo adaptativo $x'(n)$ difiere de la señal de ruido de referencia $x(n)$, de la misma manera que la señal $y'(n)$ difiere de la señal de salida del filtro adaptativo $y(n)$ debido a los efectos de $S(z)$.

El rendimiento del sistema CAR depende de la identificación de $S(z)$, mientras más precisa sea la estimación de $S(z)$ mayor será la velocidad de convergencia.

3.1 Modelado del camino secundario

El modelado del camino secundario $S(z)$ se muestra en la Fig. 3. Para obtener la estimación $S'(z)$, es necesario excitar ambos modelos con la misma señal de entrada, posteriormente se ajustan los coeficientes del filtro $S'(z)$ mediante un algoritmo adaptativo para disminuir la diferencia entre ambas señales, en este caso es posible usar el algoritmo LMS si el proceso se hace fuera de línea, ya que no sería de gran relevancia la velocidad de convergencia del algoritmo.

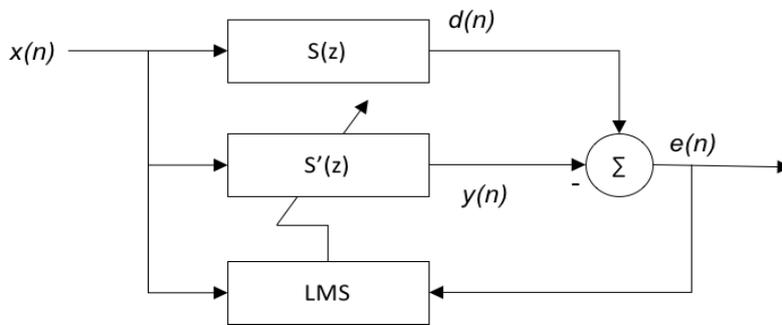


Fig. 3. Modelado de la trayectoria secundaria $S(z)$.

Para obtener los coeficientes del filtro $S'(z)$ se hace uso de la ecuación (5)

$$s'(n + 1) = s'(n) + \mu x(n)e(n), \quad (5)$$

donde $s'(n)$ es el vector de coeficientes de $S'(z)$, μ es el factor que determina la velocidad de convergencia, y $x(n)$ es la señal de entrada que excita a los sistemas.

La señal de salida del filtro se obtiene mediante (6):

$$y(n) = x(n) * s'(n). \quad (6)$$

3.2 Modelado de la trayectoria primaria y atenuación de la señal de ruido

Una vez obtenidos los coeficientes que modelan la trayectoria secundaria se procede a la etapa de cancelación la cual consiste en filtrar la señal de referencia $x(n)$ mediante $S'(z)$ y usar la señal resultante $x'(n)$ para calcular los coeficientes del algoritmo ECLMS como se muestra en la Fig. 4.

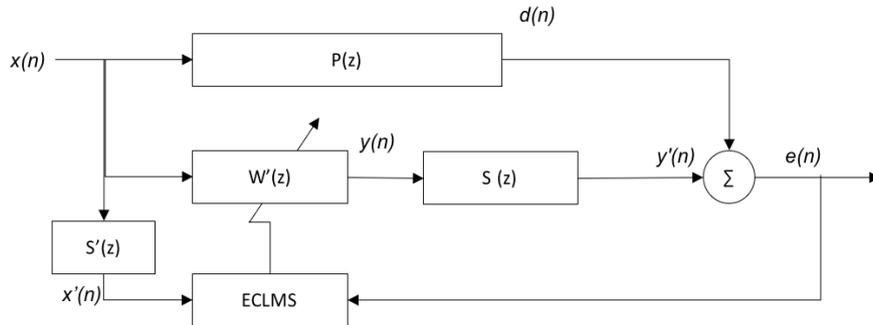


Fig. 4. Modelado de la trayectoria primaria y atenuación de la señal de ruido.

En el esquema de la Fig. 4. se observa que la señal deseada $d(n)$ es la señal de referencia $x(n)$ modificada debido a la propagación de esta por el camino físico primario.

Filtrando la señal de referencia mediante $W'(z)$, se obtiene $y(n)$ la cual es propagada por la bocina canceladora, después la trayectoria $S(z)$ la altera produciendo así la señal $y'(n)$. La señal de error se obtiene mediante su respectivo micrófono y esta representa la suma acústica entre las señales $d(n)$ y $y'(n)$.

De esta forma el algoritmo FX-ECLMS se establece mediante (7):

$$w'(n + 1) = w'(n) + \mu x'(n)C[e(n)]. \quad (7)$$

$w'(n)$ es el vector de coeficientes que modelan la trayectoria primaria, μ es el factor de convergencia y $C[e(n)]$ es la función que codifica el error.

La señal anti ruido se describe mediante (8):

$$y(n) = x(n) * p'(n). \quad (8)$$

4. Escenario de pruebas

El escenario utilizado para las pruebas del algoritmo fue un ducto de madera, Fig. 5 el cual tiene una sección transversal cuadrada con una dimensión aproximada de 9cm y una longitud de 121cm.

Se construyeron filtros analógicos pasa bajas de 500 Hz para evitar que señales con frecuencias altas modificaran la señal de referencia. El controlador fue implementado en el DSP TMS320C6713 el cual fue programado con la herramienta Code Composer Studio. Los micrófonos utilizados son de tipo MEMS, modelo ADMP401-1, el cual tiene una respuesta omnidireccional y una sensibilidad de -37 dBV, y como amplificador de potencia para la señal anti-ruido se utilizó la tarjeta TPA3110 Clase-D de Texas Instruments. Todos los componentes necesarios para implementar el sistema de control de ruido activo fueron posicionados como se muestra en el diagrama de la Fig. 6.



Fig. 5. Escenario de pruebas.

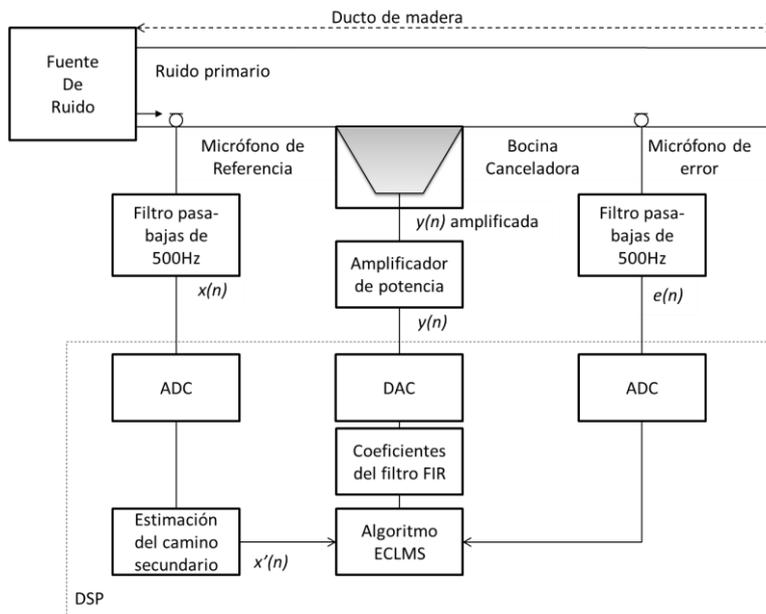


Fig. 6. Diagrama de bloques del sistema CAR.

5. Resultados

Con el propósito de evaluar el algoritmo FX-ECLMS, se realizaron diversas pruebas y los resultados fueron comparados con los del algoritmo FX-LMS. Los experimentos se realizaron utilizando como señal de referencia un tono sinusoidal con frecuencias desde los 280 Hz hasta los 580 Hz y la suma de diferentes tonos sinusoidales con frecuencias fundamentales de 300, 360 y 420 Hz. No se realizó con frecuencias más altas, debido a que es bien conocido que este tipo de sistemas tienen un buen funcionamiento para señales de baja frecuencia.

5.1 Resultados de la implementación

Ejecutando el algoritmo propuesto se analizó la señal que recibe el micrófono de error cuando el sistema CAR se encuentra activo e inactivo, dichas señales fueron almacenadas para ser comparadas en amplitud (V) y en potencia (dB). Los resultados para un tono de 320 Hz utilizando el algoritmo FX-LMS y el algoritmo FX-ECLMS, respectivamente, se muestran en la Fig. 7 y en la Fig. 8.

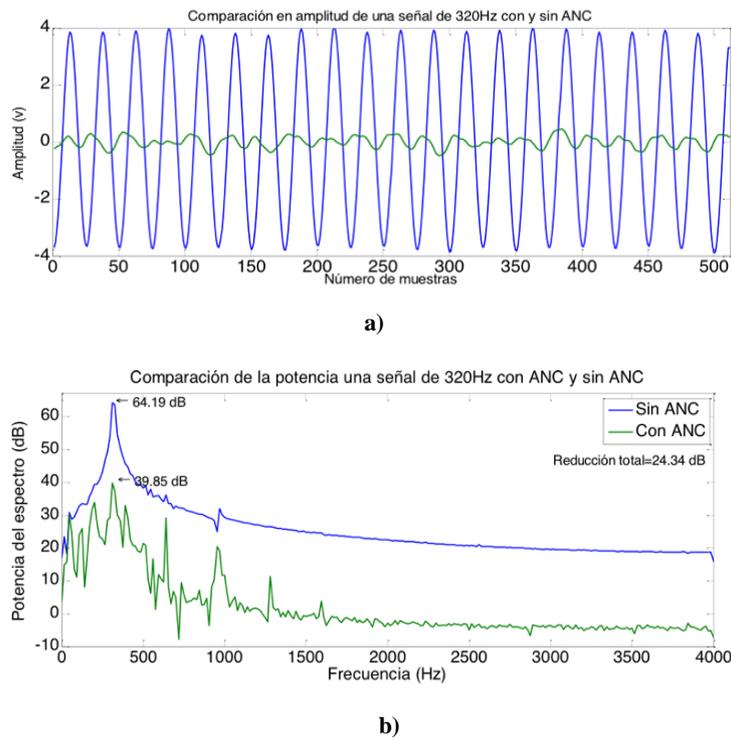
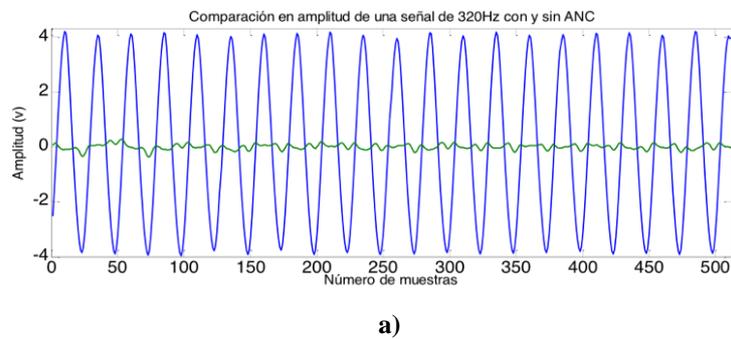
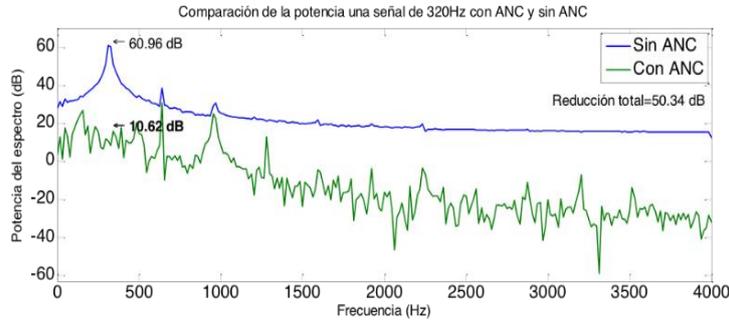


Fig. 7. Señal obtenida en el micrófono de error utilizando el algoritmo FX-LMS y una señal de referencia de 320 Hz. a) Comparación en amplitud. b) Comparación del espectro de frecuencia.





b)

Fig. 8. Señal obtenida en el micrófono de error utilizando el algoritmo FX-ECLMS y una señal de referencia de 320 Hz. a) Comparación en amplitud. b) Comparación del espectro de frecuencia.

La Tabla 1 muestra los resultados obtenidos de las diversas pruebas, las cuales demuestran que se reduce la potencia de la señal entre 24 dB y 41 dB, usando el algoritmo FX-LMS, y de 38 dB hasta 50 dB cuando se utilizó el algoritmo FX-ECLMS. También se observa la diferencia del número de actualizaciones hechas por cada uno de los algoritmos. Como se mencionó anteriormente el algoritmo FX-ECLMS no necesita actualizar sus coeficientes en cada iteración, como lo haría la versión original. De esta manera, para una señal de ruido con frecuencia fundamental de 320 Hz el algoritmo FX-ECLMS logró ahorrar un total de 28.31% de actualizaciones, en el caso de una señal de ruido con frecuencia fundamental de 350 Hz llevo a cabo un total de 58,024 actualizaciones, lo cual representó una reducción del 36.31% del total de actualizaciones.

Tabla 1. Comparativa del número de actualizaciones realizadas por los algoritmos FX-LMS y FX-ECLMS, y nivel de potencia reducida.

Frecuencia fundamental de la señal (Hz)	Actualizaciones hechas por el algoritmo FX-LMS	Actualizaciones hechas por el algoritmo FX-ECLMS	Potencia reducida FX-LMS (dB)	Potencia reducida FX-ECLMS (dB)
300	101,418 (100%)	79,497 (78.38%)	41.39	38.08
320	100,674 (100%)	72,177 (71.69%)	24.34	50.34
350	91,095 (100%)	58,024 (63.69%)	31.41	44.54

Para comprobar la eficiencia del algoritmo se hizo una prueba más robusta, la cual consistió en usar como fuente de ruido una señal multitono con frecuencias de 300, 360 y 420 Hz.

Para este experimento la potencia de los tonos se redujo en diferente nivel para cada una de las frecuencias. La Tabla 2 muestra las reducciones en potencia y las actualizaciones realizadas por cada uno de los algoritmos utilizados. En este caso el algoritmo

propuesto no reduce en forma considerable los niveles de potencia con respecto a los obtenidos con el algoritmo FX-LMS, sin embargo, el número de actualizaciones se reduce.

Tabla 2. Comparativa del número de actualizaciones realizadas y potencia reducida por los algoritmos Fx-LMS y Fx-ECLMS para una señal multitono de 300, 360 y 420 Hz.

Frecuencias fundamentales del multitono (Hz)	Actualizaciones FX-LMS	Actualizaciones FX-ECLMS	Potencia reducida Fx-LMS (dB)	Potencia reducida Fx-ECLMS (dB)
300	408,071 (100%)	388,558 (95.21%)	31.23	21.66
360			13.77	9.66
420			5.85	2.82

6. Conclusiones

Los resultados obtenidos demuestran que el algoritmo FX-ECLMS implementado en un sistema CAR reduce de manera adecuada las señales de ruido de baja frecuencia. Cabe mencionar, que en algunos casos la reducción del ruido es mayor que la obtenida con el algoritmo FX-LMS. Además, se comprobó que el algoritmo FX-ECLMS no requiere actualizar sus coeficientes en cada iteración, reduciendo así la carga computacional, lo cual es muy benéfico ya que se puede liberar recursos del procesador y así utilizarlos para realizar otras tareas. También es importante considerar que la codificación del error disminuye el número de operaciones de punto flotante, de esta forma al realizar un menor número de operaciones y en menos ciclos de reloj, el procesador reducirá el consumo de potencia. Por lo tanto, queda demostrado que este algoritmo puede ser una gran alternativa para la implementación de sistemas CAR.

Referencias

1. Kuo, S. M., Morgan, D. R.: Active Noise Control System; Algorithms and DSP implementations. 1st edn. John Wiley & Sons, Inc., New York (1996)
2. Diniz, P. S. R.: Adaptive Filtering: Algorithms and practical implementation. 4th edn. Springer, Boston (2013)
3. Tokatli, A.: Design and implementation of a DSP based active noise controller for headsets. Dissertation, Middle East Technical University, Turkey (2004)
4. Moazzam, M., Rabbani, M. S.: Performance Evaluation of Different Active Noise Control (ANC) Algorithms for Attenuating Noise in a Duct. Dissertation, Blekinge Institute of Technology, Sweden (2013)
5. Garcia, M., Diego, P., Quintana, R.: DSP implementation of the FxLMS algorithm for active noise control: Texas instruments TSM320C6713DSK. In: Automatic Control (CCAC), IEEE 2nd Colombian Conference, pp. 1–6 (2015)
6. Sidhu, S.: Implementation of Active Noise Cancellation in a Duct. Dissertation, Simon Fraser University, Canada (2013)

7. Ferrer, M., De Diego, M., González, A., Pinero, G.: Efficient implementation of the affine projection algorithm for active noise control applications. In: Signal Processing Conference, 12th European, IEEE (2004)
8. Velázquez, J., Ávalos, J. G., Sánchez, J. C.: Comparación e Implementación de los Algoritmos ECLMS y OAECLMS en un DSP. *Computación y Sistemas*, vol. 14, no. 2, p. 175–185 (2010)
9. Martínez, H. M., Avalos, J. G., Sánchez, J. C., Velázquez, J.: Mejora de la velocidad de procesamiento del algoritmo ECLMS e implementación en el DSP C6713. In: Congreso Internacional en Ingeniería Electrónica, pp. 306–311 (2014)