

PROCESAMIENTO ADAPTABLE Y SUS APLICACIONES

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

Instituto Politécnico Nacional, ESIME Culhuacan (México)

jvelazquezl@ipn.mx, javaloso@ipn.mx, jcsanche@ipn.mx

<https://doi.org/10.3926/oms.411.1>

Velázquez López, J., Ávalos Ochoa, J. G., & Sánchez García, J. C. (2022). Procesamiento adaptable y sus aplicaciones. En M. A. Ramírez Salinas, L. N. Oliva Moreno, L. I. Garay Jimenez y P. Gomez Miranda (Ed.), *Avances 2022: Red de Investigación Computación del Instituto Politécnico Nacional, México* (pp. 3-19). Barcelona, España: OmniaScience.

Resumen

En el contexto del procesamiento digital de señales, se han desarrollado algoritmos para el filtrado adaptable, con la finalidad de resolver la interferencia de señales no deseadas en la banda útil de información. Históricamente los algoritmos de mínimos cuadrados promediados (LMS – Least Mean Square), con sus diferentes variantes, han sido los de mayor uso debido a su simplicidad computacional además de su gran estabilidad, aunque con una baja velocidad de convergencia.

Con la finalidad de tener un balance entre la velocidad de convergencia, estabilidad y reducción en la complejidad computacional, se han presentado variantes del algoritmo LMS basadas en proyecciones afines (AP – Affine Projection Algorithm), con la finalidad de mejorar su eficiencia, así como la simplificación de operaciones a realizar, teniendo como consecuencia reducción en el costo computacional. Una de estas variantes es el caso de proyecciones afines con orden evolutivo, así como su adaptación para aplicaciones en la cancelación activa de ruido (CAR). En el presente capítulo se explican estas variantes, así como las pruebas realizadas en la cancelación activa de ruido en recintos y en ductos, comparando su comportamiento bajo diferentes condiciones.

Palabras clave

Sistemas adaptativos, control activo de ruido, algoritmo de mínimos cuadrados promediados, algoritmo de proyecciones afines, filtrado digital.

1. Introducción

En diversas aplicaciones del procesamiento de señales, el filtrado adaptable, como proceso, y los filtros adaptativos como implementación, tienen alto impacto en casos como: identificación de sistemas, cancelación de eco, igualación de canales de comunicación, cancelación activa de ruido. En la Figura 1 se puede observar la estructura general del un filtro adaptativo, con un filtro digital que puede variar en el valor de sus coeficientes de acuerdo con la información que le envía el algoritmo adaptativo. En esta figura $x(n)$ es la señal de entrada al filtro, $y(n)$ es la señal de salida del filtro adaptativo, $d(n)$ es la señal deseada y $e(n)$, que es la diferencia entre $y(n)$ y $d(n)$, es la señal de error. Con esta estructura, el objetivo que se persigue es la minimización de la señal de error.

De entre la gran cantidad de algoritmos adaptativos que se pueden encontrar en la literatura, el algoritmo de mínimos cuadrados promediados (LMS-Least Mean Square), es el más utilizado dada su baja complejidad computacional. Sin embargo, su estabilidad y velocidad de convergencia, dependen en mucho de la adecuada elección de su factor de convergencia. Por esta razón existen varias modificaciones que tienen como objetivo mejorar el funcionamiento del algoritmo, mediante diferentes opciones de cálculo o uso de ese factor de convergencia [1-4].

Otra opción para una mejor actualización de coeficientes, con alto costo computacional, es el algoritmo de proyecciones afines (AP – affine projection algorithm), que presenta una alta velocidad de convergencia, por el re-uso de múlti-

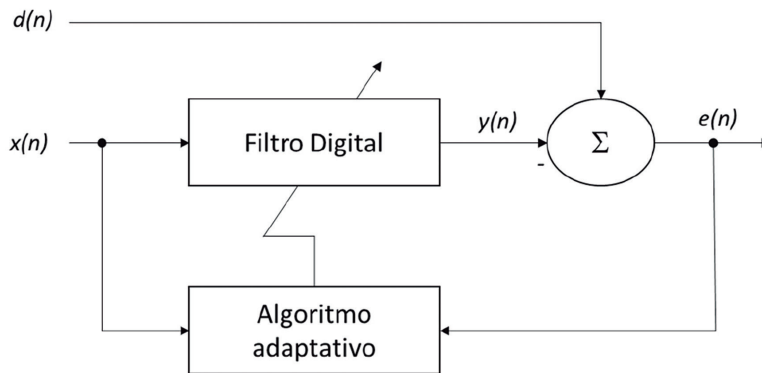


Figura 1. Estructura general de un filtro adaptativo

ples vectores de entrada, también conocido como orden de proyección (M) [5]. Básicamente si el orden de proyección es alto se tendrá una alta velocidad de convergencia; aunque en estado estacionario ésta disminuye. Para solucionar los problemas que presenta el algoritmo AP diversos autores han desarrollado modificaciones [6-9]. En algunos casos, se usan funciones de membresía o estructuras por sub-bandas, para determinar el orden de proyección de manera dinámica, para reducir la carga computacional

La combinación convexa de filtros [10-12], es una técnica mediante la cual se concierta el funcionamiento de dos filtros adaptativos trabajando con dos algoritmos de diferente rendimiento, ofreciendo características complementarias de error mínimo estacionario y velocidad de convergencia, buscando obtener la mejor aproximación a través de un algoritmo convexo.

2. Algoritmo de proyecciones afines

El fundamento del algoritmo de AP, es en las proyecciones afines de tipo sub-espacial [4] donde la actualización de los coeficientes implica el uso de más de un vector de entrada, como se muestra en:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{X}(n)(\mathbf{X}^t(n)\mathbf{X}(n) + \delta \mathbf{I}_M)^{-1} \mathbf{e}(n) \quad (1)$$

donde $\mathbf{w}(n)$ representa el vector de coeficientes, μ el factor de convergencia, δ es un parámetro de regularización, \mathbf{I}_M una matriz identidad de tamaño $M \times M$, $\mathbf{e}(n)$ es la señal de error y $\mathbf{X}(n)$ la matriz de la señal de entrada formada por los vectores $\mathbf{x}(n)$ dados por:

$$\mathbf{X}(n) = [\mathbf{x}(n), \mathbf{x}(n-1), \dots, \mathbf{x}(n-M+1)] \quad (2)$$

La señal de error $\mathbf{e}(n)$ se calcula como se muestra a continuación:

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{w}(n)\mathbf{X}(n) \quad (3)$$

donde $\mathbf{d}(n)$ es el vector de la señal deseada, el cual está compuesto de la siguiente forma:

$$\mathbf{d}(n) = [\mathbf{d}(n), \mathbf{d}(n-1), \dots, \mathbf{d}(n-M+1)] \quad (4)$$

2.1. Algoritmo de proyecciones afines con orden evolutivo

Dentro de las limitantes que se tiene para la aplicación de los algoritmos adaptativos, es la complejidad computacional que presentan para su posible implementación en sistemas de cómputo de hardware limitado. Una alternativa para la reducción de la carga computacional del algoritmo AP, es aplicar un método de ajuste dinámico del orden de proyección a través de la variación del número de vectores de entrada, conforme al valor estacionario del error cuadrático promedio [13], tomando como referencia el umbral estimado a través de (5).

$$M(n) = \begin{cases} \min\{M(n-1) + 1, M_{max}\}, & \text{si } \eta(n) < \mathbf{e}^2(n) \\ M(n-1), & \text{si } \theta(n) < \mathbf{e}^2(n) \leq \eta(n) \\ \max\{M(n-1) - 1, 1\}, & \text{si } \mathbf{e}^2(n) < \theta(n) \end{cases} \quad (5)$$

donde $M(n)$ es la cantidad de vectores de re-uso en cada n -ésima iteración, y el orden de proyección mayor a utilizar está definido por M_{max} . El umbral máximo y mínimo de (5) $\eta(n)$ y $\theta(n)$, determinan el estado estacionario del error cuadrático medio y están definidos por (6) y (7):

$$\eta(n) = \sigma_V^2 \frac{\mu M(n-1) + 2}{2 - \mu} \quad (6)$$

$$\theta(n) = \sigma_V^2 \frac{\mu k(n-1) + 2}{2 - \mu} \quad (7)$$

Esta modificación al algoritmo AP mejora las características de convergencia además de reducir la carga computacional, debido a que, en cada iteración sólo se utiliza el número de vectores de entrada que se considera necesario. Adicionalmente, es posible aplicar la teoría de combinación convexa, empleando dos filtros adaptables con diferentes condiciones en el algoritmo de adaptación cuya estructura básica se muestra en la Figura 2, donde se deduce que la señal de salida del filtro está dada por:

$$\mathbf{y}(n) = \lambda(n)\mathbf{y}_1(n) + (1 - \lambda(n))\mathbf{y}_2(n) \quad (8)$$

donde $\mathbf{y}_i(n)$ es la señal de salida del filtro i , $\lambda(n)$ se evalúa en cada iteración y corresponde a la medida de mezcla, controlando de esta manera la combinación de los filtros, el cual se obtiene como sigue:

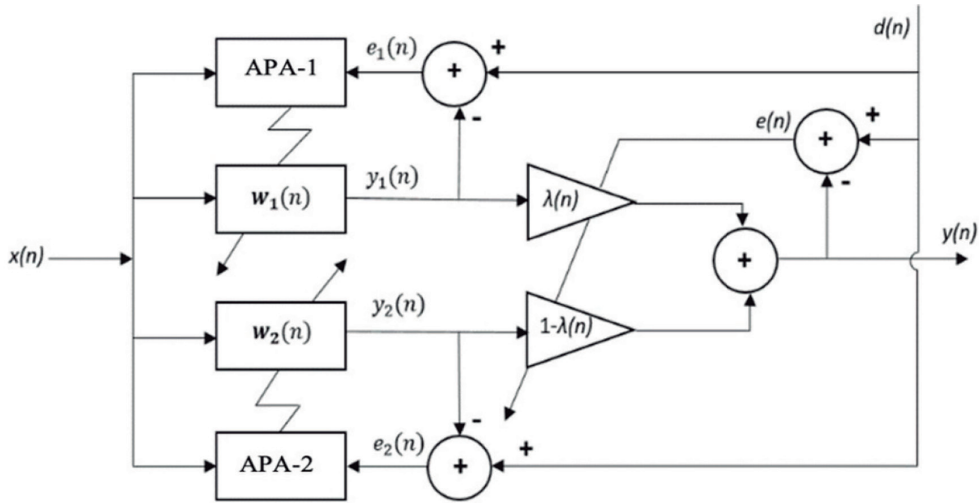


Figura 2. Estructura general de un filtro convexo

$$\lambda(n) = \frac{1}{1 + e^{-a(n)}} \quad (9)$$

donde $\mathbf{a}(n)$ se actualiza en cada iteración para reducir el error cuadrático instantáneo del filtro combinado definido por $\mathbf{e}(n)^2 = (\mathbf{d}(n) - \mathbf{y}(n))^2$ [13], como se expresa en (10).

$$\mathbf{a}(n + 1) = \mathbf{a}(n) + \mu_a \mathbf{e}(n) (\mathbf{e}_2(n) - \mathbf{e}_1(n)) \lambda(n) (1 - \lambda(n)) \quad (10)$$

De la ecuación (10) μ_a controla el ajuste de $\mathbf{a}(n)$, el cual tiene un de valor grande para aumentar la velocidad de convergencia del filtro convexo, adicionalmente $\mathbf{a}(n)$ pretende obtener las mejores características de cada filtro. De acuerdo con la ecuación (9), $\lambda(n)$ se mantiene entre 0 y 1, permitiendo la discriminación entre un filtro u otro, pero con la finalidad de evitar alteración en el funcionamiento del filtro cuando $\lambda(n)$ y $(1-\lambda(n))$ presenten un valor simultáneo igual a 0, $\mathbf{a}(n)$ se constriñe al intervalo $[-4,4]$. En consecuencia, sí $\lambda(n) \in (0,1)$, se garantiza la estabilidad del filtro convexo, combinado al aprovechamiento con las características de velocidad de la mezcla de los algoritmos usados.

Otro aspecto a prestar atención en la Figura 2, es el hecho de que los filtros no están acoplados entre sí, entonces tienen un funcionamiento independiente, teniendo como consecuencia que la mezcla de un filtro AP con orden de pro-

yección alto con el algoritmo LMS o alguna de sus variantes, la combinación considerará las mejores particularidades de ambos filtros, ocasionando una alta velocidad de convergencia, estabilidad y una reducción del costo computacional, lo cual no es posible a través de las propuestas realizadas en la combinación de dos filtros AP presentadas en [10-12], en donde se considera de manera estática, el orden de proyección del algoritmo AP.

$$\mathbf{w}(n + 1) = \mathbf{w}(n) + \mu x(n)\mathbf{e}(n) \tag{11}$$

El algoritmo de mínimos cuadrados promediados LMS, emplea (11) como ecuación de actualización de los coeficientes \mathbf{w} del filtro:

donde $x(n)$ es la señal de entrada, $e(n)$ el valor del error y el μ factor de convergencia. Se puede apreciar en (11), que la simplicidad de este algoritmo está en que sólo depende de los valores actuales de los coeficientes del filtro, el valor de la muestra actual, así como del error. Con la finalidad de aumentar la velocidad de convergencia, así como reducir el costo computacional, una de las variantes probadas de este algoritmo, es realizando una codificación del error [14] como se muestra en (12).

$$\mathbf{w}(n + 1) = \mathbf{w}(n) + \mu x(n)\mathbf{C}[e(n)] \tag{12}$$

Para el cálculo del error codificado se emplea (13)

$$\mathbf{C}[e(n)] = \mathbf{round}\left(\frac{e(n)}{2^b - 1}\right) \tag{13}$$

siendo b el número de bits con el que se realiza la codificación del error. Como se puede apreciar en esta ecuación, el valor del error codificado es un valor de tipo entero, lo que computacionalmente genera una reducción en la actualización de los coeficientes, adicionalmente, al llegar a su valor estacionario, su valor puede estar alternando entre 1 y -1, simplificando aún más la operación en la actualización de los coeficientes del filtro.

2.2. Estructura de Filtrado-x en sistemas de cancelación activa de ruido multicanal

La estructura que se emplea para el Control Activo de Ruido (CAR), tiene un arreglo similar con los identificadores de sistemas, pero la diferencia radica en

que la señal de salida del controlador $y(n)$, debe ser opuesta a la del sistema desconocido $d(n)$, además, la señal $d(n)$ atraviesa un camino acústico también conocido como trayectoria secundaria $S(z)$, que altera sus parámetros de amplitud y fase, por lo que se hace necesario contrarrestar sus efectos.

Una de las técnicas usadas para reducir el efecto de $S(z)$, es la de Filtrado x (Fx) debido a su baja complejidad computacional, la cual se ejemplifica en la Figura 3. El procedimiento que realiza consiste en filtrar la señal de referencia $x(n)$, con un modelo de la respuesta al impulso de la trayectoria secundaria, $S'(z)$, para obtener la señal de referencia filtrada, $x'(n)$, posteriormente $x'(n)$ y la señal de error $e(n)$, se emplean para actualizar los coeficientes del filtro adaptable.

En la Figura 3, K es el número de sensores de error, J el número de altavoces, $x_i(n)$ es la señal i de referencia obtenida de I sensores de referencia y $y_j(n)$ es la salida del filtro adaptable como se puede apreciar en (14)

$$y_j(n) = \mathbf{w}^T_{ji}(n) \mathbf{x}_i(n) \quad (14)$$

donde $\mathbf{w}(n)$ es el vector de coeficientes compuesto por:

$$\mathbf{w}_{ji}(n) = [\mathbf{w}_{j,i,0}(n), \mathbf{w}_{j,i,1}(n), \dots, \mathbf{w}_{j,i,M-1}(n)]^T \quad (15)$$

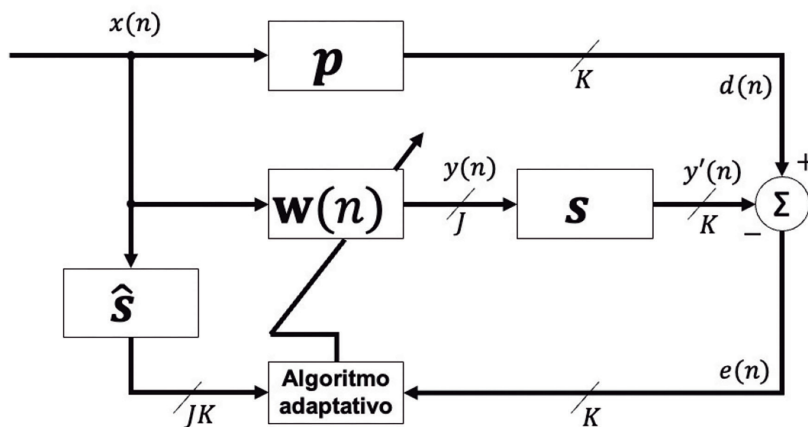


Figura 3. Sistema CAR multicanal empleando Filtrado- x

y la señal de referencia es constituida por:

$$\mathbf{x}_i(n) = [\mathbf{x}_i(n), \mathbf{x}_i(n-1), \dots, \mathbf{x}_i(n-M+1)]^T \quad (16)$$

Donde M denota el orden del filtro. Por otro lado, P es la trayectoria primaria, \tilde{S} es el sistema encontrado con una respuesta idéntica a S y $e(n)$ es la señal de error que se define como:

$$\begin{aligned} e(n) &= d(n) - y'(n) \\ &= d(n) - s(n)y(n) \end{aligned} \quad (17)$$

Donde $y'(n)$ es la señal modificada debido a los efectos de la trayectoria secundaria S y $d(n)$ representa la señal deseada.

3. Resultados

Hasta el momento se ha presentado el algoritmo de proyecciones como una alternativa al incremento en la velocidad de convergencia, pero con la limitante en la complejidad computacional y en el nivel del error cuadrático promediado, así como una propuesta a la mejora de su desempeño mediante el ajuste dinámico al orden de proyección y la estructura convexa para combinarse con un algoritmo de menor complejidad. En los siguientes apartados se presentan resultados obtenidos de pruebas realizadas a las propuestas presentadas con anterioridad en sistemas de Control Activo de Ruido (CAR). En la primera parte se revisa el comportamiento a nivel simulación, y posteriormente se analiza su comportamiento en una aplicación de cancelación activa de ruido en ductos.

3.1. Simulación de combinación convexa de los algoritmos AP-LMS para sistemas CAR

El algoritmo propuesto es una combinación de un algoritmo de alta velocidad de convergencia (algoritmo AP) y uno de baja complejidad computacional (algoritmo LMS). Para reducir el número de actualizaciones requeridas por el algoritmo y su carga computacional, se usó la teoría de la codificación del error [13]. Además, los algoritmos se probaron en aplicaciones de control activo de ruido,

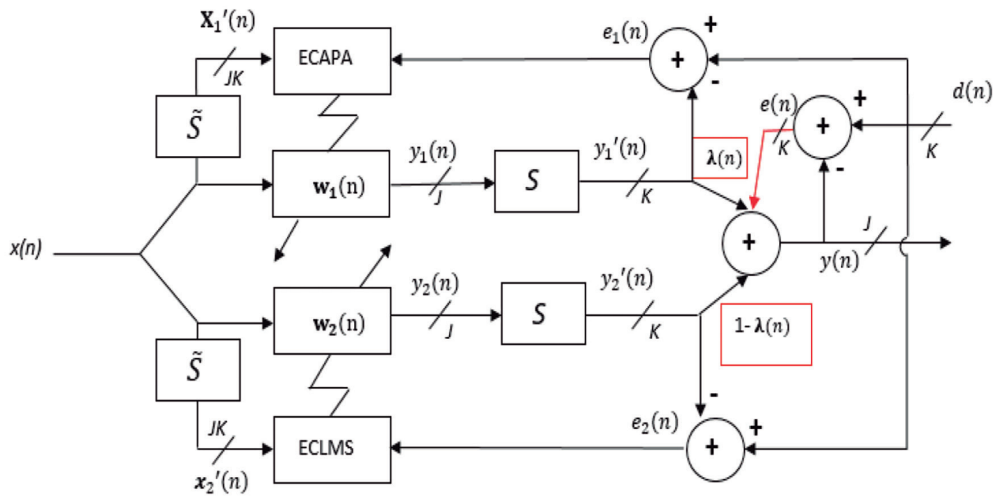


Figura 4. Estructura general del algoritmo propuesto

por lo que se utilizó la estructura de filtrado-x mostrada anteriormente. Por este motivo los algoritmos desarrollados se nombraron como FX-ECAP (filtered-x error coded AP) y FX-ECLMS (filtered-x error coded LMS). La Figura 4 muestra la estructura general de un filtro convexo utilizando la propuesta descrita.

Resultados de simulación empleando Matlab™, evaluándose en una configuración multicanal 1:2:2. Los resultados se obtuvieron al promediar veinticinco experimentos individuales con 40,000 iteraciones. Para evaluar el funcionamiento del algoritmo se llevaron a cabo dos experimentos diferentes, en el primero se usó como señal de referencia una señal multitono y en el segundo ruido Gaussiano blanco. Las trayectorias primarias P y secundarias S se modelaron utilizando filtros de respuesta al impulso finita (FIR) de longitudes 256 y 128 respectivamente, obtenidos de [3]. Para probar la capacidad de seguimiento del algoritmo los coeficientes de la trayectoria P se invirtieron a la mitad de las iteraciones. Además, en los micrófonos de error se agregó ruido gaussiano hasta obtener una relación señal a ruido (SNR) de 30 dB. En todos los experimentos la longitud de los filtros adaptativos y el orden de proyección (L) se establecieron como $N = 100$ y $L = 5$. El funcionamiento de los algoritmos se evaluó mediante el error cuadrático medio (MSE – mean square error) obtenido en los micrófonos de error. Los resultados obtenidos se compararon con la versión convencional de los algoritmos FXECAP y FXECLMS.

• **Caso 1: señal multitono**

Para esta prueba se utilizó como señal de referencia una señal multitono con frecuencias de 200, 300 y 400 Hz, muestreadas a 2 kHz. Tanto para los algoritmos ECAP como ECLMS, el error se codificó utilizando 4 bits, además el tamaño del paso usado en μ_{ECAP} para la trayectoria 11 es 0.03, para la trayectoria 12 se usó 0.01, 0.009 para la trayectoria 21 y 0.01 para la trayectoria 22, para el algoritmo ECLMS se usó 0.0000095 para todas las trayectorias, estos valores se eligieron para obtener la velocidad de convergencia más rápida. En la Figura 5 se presenta el MSE medido en el micrófono de error 1 (los resultados obtenidos en el micrófono de error 2 son bastante similares por lo que solo se reportan los obtenidos en el primer micrófono). Las curvas de aprendizaje muestran que el algoritmo propuesto retiene la velocidad del algoritmo FXECAP pero reduce el desajuste al nivel del algoritmo FXECLMS.

En la Tabla 1, se muestra el número de ocasiones que se actualizan los coeficientes de cada algoritmo. Como se puede observar, la principal ventaja del algoritmo convexo propuesto es que solo necesita ejecutar en un 4.8% de las iteraciones el algoritmo FXECAP, mientras que el algoritmo FXECLMS actualiza sus coeficientes un 72.2 % del proceso.

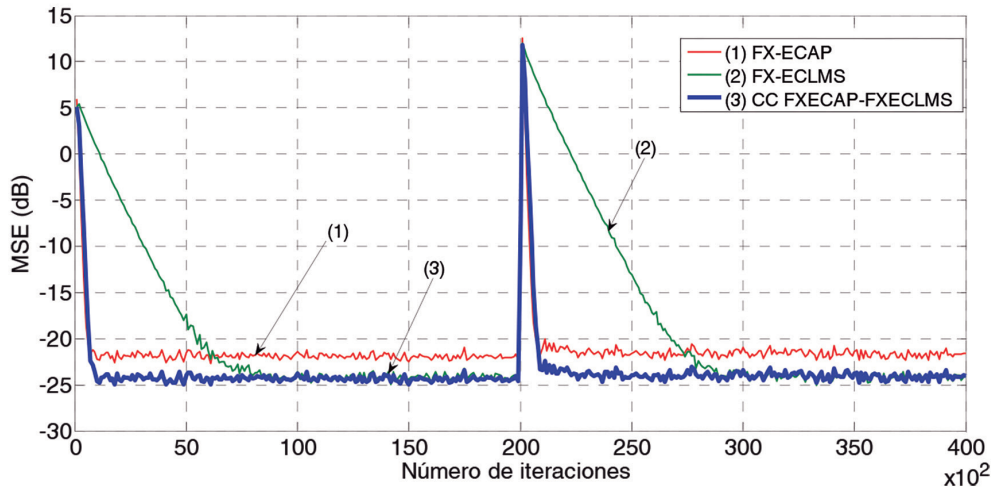


Figura 5. MSE del algoritmo FXECAP, el algoritmo FXECLMS y el algoritmo CC FXECAP-FXECLMS utilizando como señal de referencia una señal multitono

Algoritmo	Actualizaciones
FXECLMS	577,643 (72.20%)
FXECAP	39,144 (4.8%)

Tabla 1. Número de actualizaciones para el caso I

• **Caso 2: señal de ruido Gaussiano blanco**

Para la segunda prueba se usó como señal de referencia una señal de ruido Gaussiano blanco. De nuevo se codificó el error a 4 bits y se eligió el factor de convergencia μ_{ECAP} para la trayectoria 11 de 0.01, para la trayectoria 12 de 0.02, 0.03 para la trayectoria 21 y 0.02 para la trayectoria 22, para el algoritmo ECLMS se usó 0.0001 para todas las trayectorias. La Figura 6 presenta el MSE obtenido. Como se observa, el algoritmo propuesto retiene la velocidad del algoritmo FXECAP y reduce el desajuste al nivel del algoritmo FXECLMS, al igual que en el experimento anterior.

La Tabla 2 presenta el número de actualizaciones de coeficientes realizadas por cada uno de los algoritmos que componen la propuesta convexa. Los resultados muestran de nuevo que el algoritmo FXECAP se ejecuta en muy pocas ocasio-

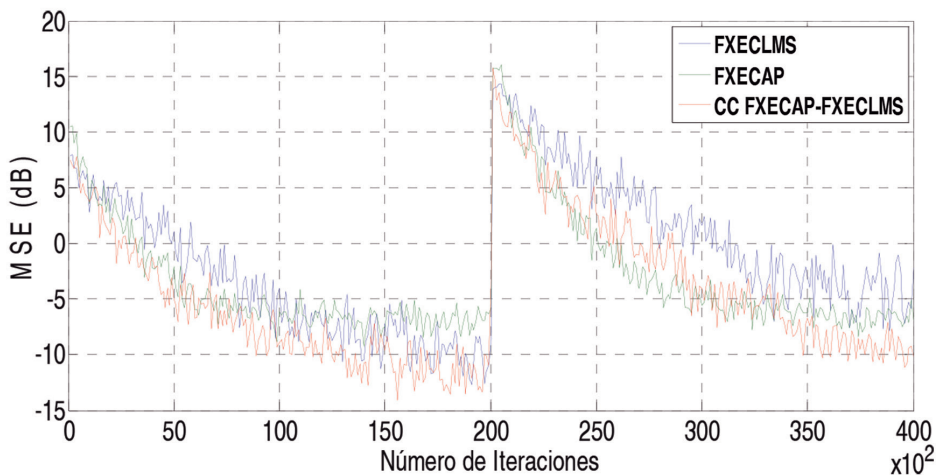


Figura 6. MSE del algoritmo FXECAP, algoritmo FXECLMS y el algoritmo CC FXECAP-FXECLMS, utilizando como señal de referencia ruido Gaussiano blanco

nes durante el proceso, lo que significa una considerable reducción en la carga computacional.

Algoritmo	Actualizaciones
FXECLMS	581,026 (72.63%)
FXECAP	87,640 (10.96%)

Tabla 1. Número de actualizaciones para el caso II

3.2. Comportamiento de los algoritmos FXECAP y FXECLMS aplicados en sistemas CAR en ductos

Para comprobar el funcionamiento y rendimiento del algoritmo se compararon los resultados del algoritmo FXECAP-L con orden evolutivo y factor de escala dinámico con respecto al algoritmo FXAP-L y el algoritmo FXECLMS-L. Se realizaron dos pruebas en un modelo a escala de un ducto de ventilación, en la primera se utilizó como señal de ruido un tono senoidal con una frecuencia de 500 Hz y en la segunda prueba una suma de tonos con frecuencias fundamentales de 600, 750 y 900 Hz. La evaluación se realizó mediante una computadora externa utilizada para grabar la señal del micrófono de error, posteriormente se obtuvo su espectro de potencia usando el software Signal Analyzer de Matlab.

Las dimensiones del ducto son de 121 centímetros de largo y 9 centímetros de ancho. Está construido con madera triplay y esta internamente forrado con vinilo adhesivo para disminuir las reverberaciones ocasionadas por la rugosidad de la madera. En uno de los extremos del ducto se colocó una bocina como fuente primaria de ruido, posteriormente se ubicó el micrófono de referencia a 11 centímetros de distancia, un micrófono de error que se localiza a 117 centímetros de la fuente de ruido y una bocina antiruido a 80 centímetros. Esta configuración se puede apreciar en la Figura 7.

Posteriormente se instalaron filtros analógicos pasabajas con frecuencia de corte de 1 kHz entre el micrófono de referencia y el de error y los convertidores CAD/CDA, esto con el fin de evitar el efecto “antialiasing” y de reducir el ancho de banda de las señales entrantes. Cabe mencionar que para todos los experimentos

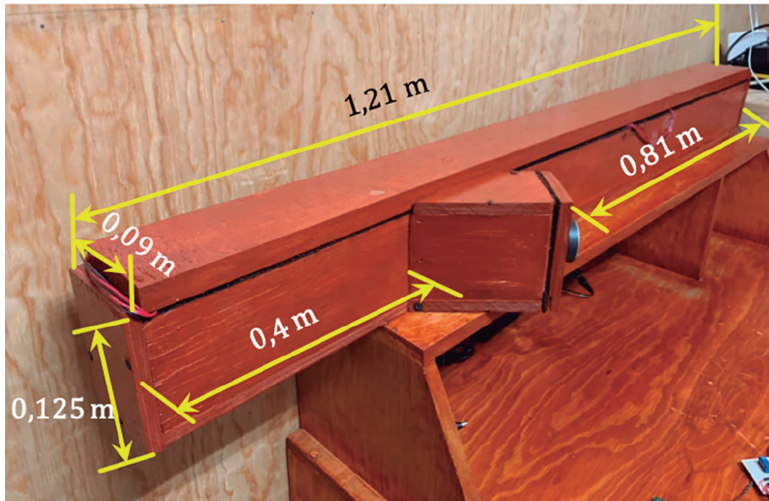


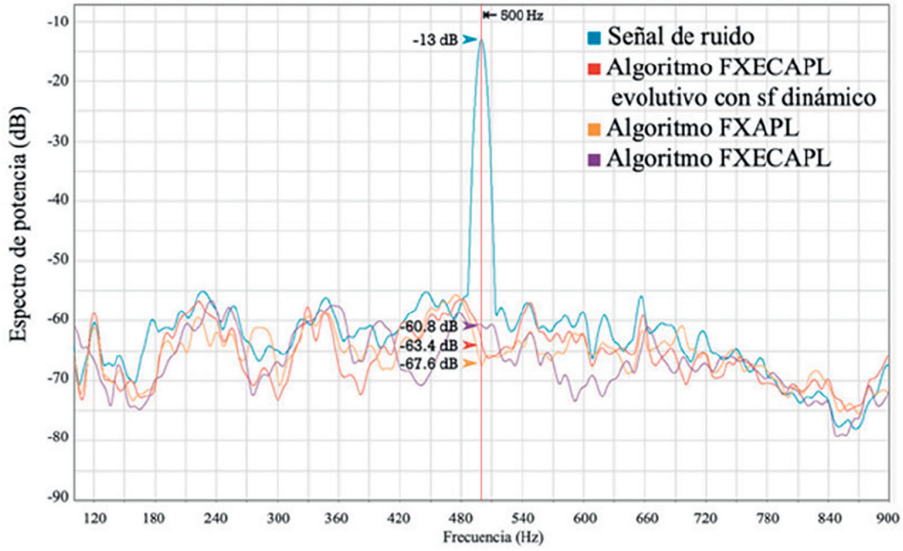
Figura 7. Ducto acústico usado para sistemas de CAR

la longitud del filtro adaptativo fue de 128 coeficientes y el orden de proyección para todos los algoritmos fue de $L = 5$. En la Figura 8a, se muestra los espectros de potencia de las señales obtenidas por el micrófono de error, cuando el sistema CAR está apagado y cuando es encendido, usando los diferentes algoritmos adaptativos, para la cancelación del tono de 500 Hz. En la Figura 8b se muestra los espectros de potencia de la segunda prueba, donde se usó como señal de ruido una suma de tonos senoidales con frecuencias de 600, 750 y 900 Hz. Como se puede apreciar, la señal de ruido presenta una potencia de -17, -26.3 y -21.4 dB para cada una de sus frecuencias, para la cancelación de una señal multitono.

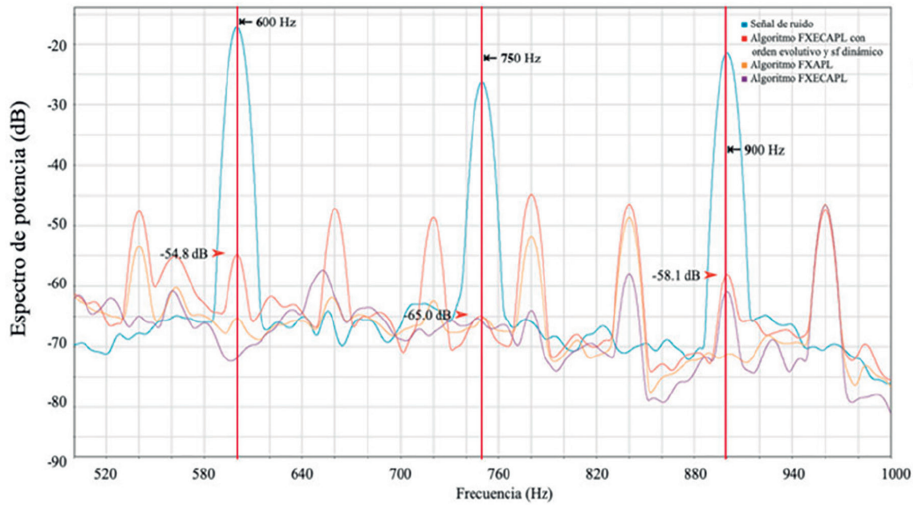
4. Conclusiones

Las pruebas realizadas a los algoritmos y estructuras adaptativas desarrolladas son altamente efectivas para el diseño de sistemas de ANC, donde se requiera alta velocidad de convergencia y baja carga computacional, adicionalmente, los resultados obtenidos consideraron el diseño de algoritmos y su implementación en sistemas de un solo canal. Lo anterior demuestra que es posible continuar en el desarrollo de sistemas específicos que puedan ser implementados en recintos acústicos con problemas de contaminación de ruido de múltiples fuentes, por lo que de esta forma se podrían realizar sistemas que puedan ser utilizados en

hospitales, escuelas, oficinas u hogares; beneficiando la salud de los usuarios, ya que la exposición al ruido puede provocar estrés, presión arterial alta, problemas cardiacos, problemas respiratorios, trastornos del sueño e incluso generar una pérdida auditiva con el tiempo.



a) Cancelación de tono de 500 Hz



b) Cancelación de sumas de tonos

Figura 8. Espectro de potencia del sistema CAR

Agradecimientos

El contenido del presente capítulo, fue obtenido gracias al apoyo de la Secretaría de Investigación y Posgrado del Instituto Politécnico Nacional, para la realización de los proyectos de investigación con registro: 20171627, 20180004 y 20200013

Referencias

- [1] Cheng, S., Wei, Y., Chen, Y., Liang, S., & Wang, Y. (2017). A universal modified LMS algorithm with iteration order hybrid switching. *ISA transactions*, 67, 67-75. <https://doi.org/10.1016/j.isatra.2016.11.019>
- [2] Chaudhary, N. I., Latif, R., Raja, M. A. Z., & Machado, J. T. (2020). An innovative fractional order LMS algorithm for power signal parameter estimation. *Applied Mathematical Modelling*, 83, 703-718. <https://doi.org/10.1016/j.apm.2020.03.014>
- [3] Chergui, L., & Bouguezel, S. (2019). A New Post-whitening Transform Domain LMS Algorithm. *Traitement du Signal*, 36(3). <https://doi.org/10.18280/ts.360307>
- [4] Ahn, D. C., Lee, J. W., Shin, S. J., & Song, W. J. (2017). A new robust variable weighting coefficients diffusion LMS algorithm. *Signal Processing*, 131, 300-306. <https://doi.org/10.1016/j.sigpro.2016.08.023>
- [5] Ozeki, K., & Umeda, T. (1984). An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties. *Electronics and Communications in Japan (Part I: Communications)*, 67(5), 19-27. <https://doi.org/10.1002/ecja.4400670503>
- [6] Song, P., Zhao, H., & Zeng, X. (2019). Robust diffusion affine projection algorithm with variable step-size over distributed networks. *IEEE Access*, 7, 150484-150491. <https://doi.org/10.1109/ACCESS.2019.2947636>
- [7] Zheng, Z., & Zhao, H. (2017). Robust set-membership affine projection algorithm with coefficient vector reuse. *Circuits, Systems, and Signal Processing*, 36(9), 3843-3853. <https://doi.org/10.1007/s00034-016-0471-8>

- [8] Zhao, H., Zheng, Z., Wang, Z., & Chen, B. (2017). Improved affine projection subband adaptive filter for high background noise environments. *Signal Processing*, 137, 356-362. <https://doi.org/10.1016/j.sigpro.2017.02.017>
- [9] Qu, Z., & Zheng, Z. (2017). An Efficient L_0 Norm Constraint Memory Improved Proportionate Affine Projection Algorithm. *Circuits, Systems, and Signal Processing*, 36(8), 3448-3456. <https://doi.org/10.1007/s00034-016-0467-4>
- [10] Vázquez, Á. A., Avalos, J. G., Sánchez, G., Sánchez, J. C., & Pérez, H. (2020). A Comparative Survey of Convex Combination of Adaptive Filters. *IETE Journal of Research*, 1-11. <https://doi.org/10.1080/03772063.2020.1844075>
- [11] Ren, C., Wang, Z., & Zhao, Z. (2016). Adaptive combination of affine projection and NLMS algorithms based on variable step-sizes. *Digital Signal Processing*, 59, 86-99. <https://doi.org/10.1016/j.dsp.2016.07.022>
- [12] Rodriguez, J., Ibarra, I., Pichardo, E., Avalos, J. G., & Sanchez, J. C. (2018, November). Convex Combination of FXECAP–FXECLMS Algorithms for Active Noise Control. *2018 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)* (pp. 1-6). IEEE. <https://doi.org/10.1109/ROPEC.2018.8661379>
- [13] E. Pichardo, X. Maya, J. G. Avalos, J. C. Sánchez (2018). Combinación convexa del algoritmo de proyecciones afines con orden evolutivo y el algoritmo NLMS. *Número Especial de la Revista Aristas: Investigación Básica y Aplicada. ISSN 2007-9478, Vol.6, Núm. 12, 2018*, (pp. 141-145).
- [14] Avalos, J. G., Rodriguez, A., Martinez, H. M., Sanchez, J. C., & Perez, H. M. (2017). Multichannel Filtered-X Error Coded Affine Projection-Like Algorithm with Evolving Order. *Shock and Vibration*, 2017. <https://doi.org/10.1155/2017/3864951>