

Diseño e implementación de filtros adaptativos en un sistema de microcontroladores programables (PSoC)

NATHALY MURCIA SEPÚLVEDA¹

JENNY ANDREA SUÁREZ BARBOSA²

SALVADOR PACHECO³

JAIME G. BARRERO PÉREZ⁴

RESUMEN

Este artículo presenta el diseño y la implementación de dos filtros adaptativos (Wiener y *least mean square* [LMS]), con el objetivo de eliminar señales no deseadas presentes en una señal de referencia o deseada. Dichos filtros adaptativos fueron programados en el sistema de desarrollo PSoCTM (*programmable system on chip*); y mediante la herramienta computacional LabView[™] se desarrolló un generador virtual. Los datos generados a partir de este se usaron como entradas al sistema de filtrado y constituyeron la base del proceso llevado a cabo. Asimismo, se describen las pruebas realizadas y el análisis de cada tipo de filtro ante diferentes condiciones de ruido para una comparación final de los resultados.

Palabras clave: filtro adaptativo, filtro Wiener, filtro LMS (*least mean square*), PSoC (*programmable system on chip*).

¹ Ingeniera electrónica por la Universidad Industrial de Santander, Bucaramanga, Colombia. Correo electrónico: nathysmurcia15@hotmail.com

² Ingeniera electrónica por la Universidad Industrial de Santander, Bucaramanga, Colombia. Correo electrónico: jennyandrea22@gmail.com

³ Ingeniero electrónico por la Universidad Industrial de Santander, Bucaramanga, Colombia. Estudiante de la maestría en Ingeniería Electrónica en esta misma universidad. Correo electrónico: salvadorpacheco@gmail.com

⁴ Magister en Potencia Eléctrica e ingeniero electricista por la Universidad Industrial de Santander, Bucaramanga, Colombia. Profesor titular EEET de esta misma universidad. Correo electrónico: jbarrero@uis.edu.co

FECHA DE RECEPCIÓN: 29 DE ENERO DEL 2014 • FECHA DE APROBACIÓN: 24 DE ABRIL DEL 2014

Cómo citar este artículo: Murcia Sepúlveda, N., Suárez Barbosa, J. A., Pacheco, S. y Barrero Pérez, J. G. (2014). Diseño e implementación de filtros adaptativos en un sistema de microcontroladores programables (PSoC). *Épsilon* (22), 107-127.

Design and Implementation of Adaptive Filters in a Programmable Microcontroller System (PSoC)

ABSTRACT

This paper presents the design and implementation of two adaptive filters (Wiener filter and LMS [least mean square] filter), in order to remove unwanted signals in a desired or reference signal. These adaptive filters were programmed in the PSoCTM (programmable system on chip) development system; and a virtual generator was developed by the LabView[™] computational tool. The data generated were used as inputs to the filter system and formed the basis of the process carried out. Similarly, the article describes the tests and the analysis of each filter type under different noise conditions for a final comparison of the results.

Keywords: Adaptive filter, Wiener filter, LMS (least mean square) filter, PSoC (programmable system on chip).

Desenho e implementação de filtros adaptativos em um sistema de micro controladores programáveis (PSoC)

RESUMO

Este artigo apresenta o desenho e a implementação de dos filtros adaptativos (filtro Wiener e filtro LMS [least mean square]), com o objetivo de eliminar sinais não desejadas presentes em uma sinal de referência ou desejada. Estes filtros adaptativos foram programados no sistema de desenvolvimento PSoCTM (*programmable system on chip*); e mediante a ferramenta computacional LabView[™] se desenvolveu um gerador virtual. os dados gerados a partir deste se usaram como entradas ao sistema de filtrado e constituíram a base do processo levado a cabo. Da mesma forma, se descrevem as provas realizadas e a análise de cada tipo de filtro ante diferentes condições de ruído para uma comparação final dos resultados.

Palavras chave: filtro adaptativo, filtro Wiener, filtro LMS (*least mean square*), PSoC (*programmable system on chip*).

Introducción

Cuando las señales por filtrar son variantes en el tiempo o se encuentran en ambientes muy contaminados, los filtros de coeficientes constantes típicamente empleados no logran ser suficientes, pues trabajan con frecuencias fijas y, como su nombre lo indica, con coeficientes fijos. Esto limita su funcionamiento y evita que se procese adecuadamente la señal (Jones, 2008).

Los filtros adaptativos permiten, de una forma dinámica, confiable y segura, la reducción de señales no deseadas. Estos filtros se configuran con el propósito de reducir a cero la señal de error; para ello, se requiere que la salida de filtro sea igual a la señal deseada y se actualicen sus coeficientes de una muestra a la siguiente (Geoffrey, 2010). Dependiendo de cómo se minimiza el error, se puede implementar el filtro adaptativo.

El filtrado adaptativo puede ser usado en señales biológicas de baja frecuencia (Barajas, 2005); es el caso de señales provenientes de un sistema de medición de la presión arterial, el cual puede estar contaminado con ruidos provenientes del paciente o del entorno.

Se han realizado diferentes trabajos de implementación de filtros adaptativos en sistemas embebidos. En Augusto y Meza (2009) se implementa el filtro *least mean square* (LMS) en un procesador ARM LPC2378 para la cancelación de ruido en señales de voz. El filtro LMS también es abordado en Blanco y Ruiz (2009), donde se realiza su implementación en un *field programmable gate array* (FPGA), también aplicado a señales de voz. En Barajas (2005) se aborda la implementación de filtros adaptativos en un procesador digital de señal (DSP) para filtrar señales electrocardiográficas.

En este artículo se presentan dos filtros adaptativos: el Wiener y el LMS y los resultados de su implementación en un sistema embebido PSoC (*programmable system on chip*). La tecnología PSoC se escoge con el fin de aprovechar las características de tipo análogo de estos sistemas en trabajos posteriores.

Los filtros se prueban inicialmente fuera de línea con señales simuladas en Matlab™. Posteriormente, se usa LabView™ para generar las señales de prueba para

los filtros implementados en el sistema PSoC™. Los datos son enviados vía serial para ser visualizados en LabView™.

Los resultados presentados en el presente artículo son parte de una tesis de pregrado (Barbosa y Murcia, 2013) que, a su vez, sirve de apoyo a un trabajo de investigación que actualmente se está realizando en maestría, centrado en señales de presión arterial, titulado *Caracterización y estimación de la presión arterial latido a latido a través de modelado de parámetros de sensibilidad del método de tonometría*.

Revisión teórica

A continuación se realiza una descripción teórica de los conceptos relacionados con los filtros adaptativos tratados y algunos conceptos matemáticos usados, con el fin de hacer más clara su posterior implementación.

La mayoría de los filtros adaptativos de la actualidad se implementan por medio de sistemas digitales, debido a que presentan algunas ventajas sobre filtros analógicos, como alta confiabilidad, exactitud y menor sensibilidad a la temperatura y al envejecimiento. Sin embargo, los filtros analógicos presentan una velocidad de convergencia mayor que la de los digitales (Durán, 2010), debido a que en estos últimos se requieren un número elevado de operaciones de punto flotante (Iglesias, 2012).

La estructura de un filtro adaptativo es un sistema al que le llegan dos señales $x(n)$ y $e(n)$. Esta última se conoce como *señal de error* y es el resultado de la resta de la señal deseada, $d(n)$, y la salida del filtro, $y(n)$. A los coeficientes o pesos (*weight*) del filtro se les llama $w(n)$ y son los que multiplican a la entrada $x(n)$ para obtener la salida, como se muestra en la figura 1. Dichos coeficientes se reprograman de una muestra a la siguiente, a través de un algoritmo de adaptación, y cambian su valor dinámicamente en el tiempo.

Donde:

$$y_n = W_n X_n \quad (1)$$

$$e_n = d_n - y_n \quad (2)$$

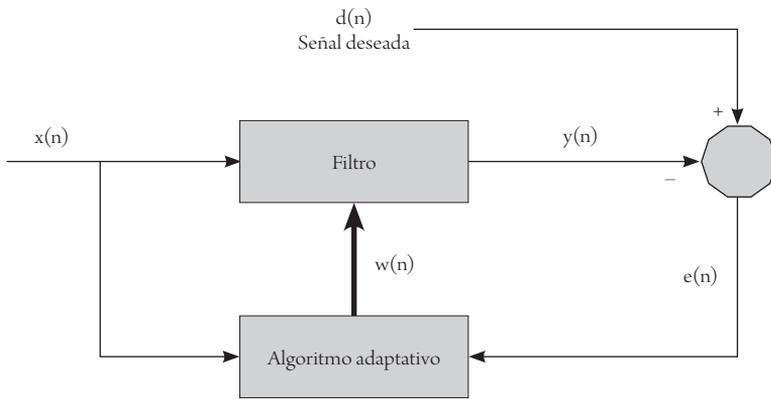


Figura 1. Estructura directa de un filtro adaptativo

Fuente: elaboración propia

En principio, el objetivo es hacer que la señal de error sea cero. Para ello, el sistema debe configurarse para que a partir de la señal de entrada $x(n)$ se genere una salida $y(n)$, de forma que sea igual a la señal deseada $d(n)$. Así como existen diferentes formas de minimizar el error, existen también diferentes tipos de filtros adaptativos (Manolakis, Ingle y Kogon, 2005). Los algoritmos de adaptación más conocidos son el Wiener, los mínimos cuadrados recursivos (RLS) y el LMS (mínimos cuadrados promediados). Debido al mayor gasto computacional requerido en la implementación de el algoritmo RLS (Borisagar, Komal y Kulkarni, 2010), el presente artículo se centrará en los filtros Wiener y LMS.

Filtro Wiener

El filtro de Wiener es uno de los filtros lineales óptimos más importantes. El objetivo del filtrado de Wiener es determinar la respuesta al impulso $h(n)$, de forma que el error $e(n)$ sea, en términos estadísticos, mínimo. El criterio elegido es la minimización del valor cuadrático medio del error:

$$\phi = E \{ | e (n) |^2 \} \quad (3)$$

Se considera un filtro causal de longitud p , orden $p-1$, de respuesta al impulso $w(n)$ y función de red (Augusto y Meza, 2009):

$$W(z) = \sum_{n=0}^{p-1} w(n)z^{-n} \quad (4)$$

Si $x(n)$ es la entrada al filtro, la salida es la estimación de $d(n)$, obtenida como la convolución de $w(n)$ con $x(n)$:

$$d(n) = \sum_{l=0}^{p-1} w(l)x(n-1) \quad (5)$$

Se deben encontrar los coeficientes que minimizan la función de coste; por tanto, derivando Φ respecto a $w^*(k)$ e igualando las derivadas a cero para los valores posibles de k ($k=0, 1, \dots, p-1$), al derivar $e^*(n)$ y sustituir, se obtiene el principio de ortogonalidad o teorema de proyección. Este teorema indica que Φ es mínimo y los coeficientes del filtro asumen sus valores óptimos cuando $e(n)$ es ortogonal a cada muestra de entrada $x(n)$, que es utilizada para el cálculo de la estimación. Por tanto, el error también es ortogonal a la salida del filtro. Este principio establece una condición suficiente y necesaria para la optimización. Se tiene un conjunto de p ecuaciones lineales con p incógnitas $w(k)$, para $k=0, 1, \dots, p-1$. En forma matricial, utilizando el hecho de que la secuencia de autocorrelación posee simetría conjugada, $r_x(k) = r_x^*(-k)$, se pueden expresar las ecuaciones de Wiener-Hopf en notación vectorial como (Physionet, 2013):

$$R_x w = r_{dx} \quad (6)$$

Donde R_x es la matriz de autocorrelación Toeplitz hermitica $p \times p$, w es el vector de coeficientes del filtro y r_{dx} es el vector de correlación cruzada entre la señal deseada $d(n)$ y la señal recibida $x(n)$.

En cada iteración, la actualización de un peso requiere dos multiplicaciones complejas y una suma. Por tanto, para un filtro de p coeficientes, se realizan $2p + 1$ multiplicaciones complejas y $2p$ sumas complejas por iteración; es decir, la carga computacional es del orden de p [$O(p)$].

Función de correlación

La correlación cruzada es una medida de la similitud entre dos señales. Es frecuentemente usada para encontrar características relevantes en una señal desconocida

por medio de la comparación con otra que sí se conoce. Dadas dos secuencias de señal reales $x(n)$ y $y(n)$, cada una con energía finita, la correlación cruzada de $x(n)$ y $y(n)$ es una secuencia $r_{xy}(l)$ definida como:

$$r_{yx}(l) = \sum_{n=-\infty}^{\infty} x(n+l)y(n) \quad l = 0, \pm 1, \pm 2, \dots \quad (7)$$

Matriz Toeplitz

En el álgebra lineal, una matriz de Toeplitz es una matriz cuadrada con todas sus diagonales, de izquierda a derecha, paralelas numéricamente. Una matriz de Toeplitz presenta la siguiente estructura (Gray, 2005):

$$\forall a_{ij} \in T \rightarrow a_{ij} = a_{i+1, j+1} \quad (8)$$

Factorización lower-upper e inversión de matrices

La descomposición *lower-upper* (LU) proporciona un método eficiente para calcular la matriz inversa, la cual tiene un número importante de aplicaciones en la práctica de la ingeniería (Chapra y Canele, 2007).

La inversa se puede calcular en forma de columna-por-columna, para generar soluciones con vectores unitarios que están en el lado derecho de la matriz como constantes. Por ejemplo, si las constantes del lado derecho de la ecuación tienen un número 1 en la primera posición y ceros en las otras, la solución resultante será la primera columna de la matriz inversa. En forma similar, si se usa un vector unitario que tiene 1 en el segundo renglón, el resultado será la segunda columna de la matriz inversa.

De esta manera, cuando se tiene la descomposición LU, es decir, expresada una matriz A en las matrices triangulares inferior L y superior U , se procede a determinar la primera columna de la matriz inversa efectuando el procedimiento de solución. Primero, se usa la sustitución hacia adelante con un vector unitario (con el número 1 en el primer renglón) como el vector de términos independientes.

Descomposición Cholesky

Una matriz A simétrica y positiva definida puede ser factorizada de manera eficiente por medio de una matriz triangular inferior y una matriz triangular superior. Para una matriz no singular, la descomposición LU lleva a considerar una descomposición de tal tipo $A = LU$. Dadas las condiciones de A , simétrica y definida positiva, no es necesario hacer pivoteo, por lo que esta factorización se hace eficientemente y en un número de operaciones la mitad de LU , tomando la forma $A = LL^T$, donde L (que se puede considerar como la raíz cuadrada de A) es una matriz triangular inferior cuyos elementos de la diagonal son positivos (Chapra y Canele, 2007).

Para resolver un sistema lineal $Ax = b$ con A simétrica definida positiva y dada su factorización de Cholesky $A = LL^T$, primero se debe resolver $Ly = b$ y, entonces, resolver $L^T x = y$ para lograr x . De manera general, para $i = 1, \dots, n$ y $j = i + 1, \dots, n$:

$$l_{ii} = \sqrt{\left(a_{ii} - \sum_{k=0}^{i-1} l_{ik}^2 \right)} \quad (9)$$

$$l_{ji} = \left(a_{ji} - \sum_{k=1}^{i-1} l_{jk} l_{ik} \right) / l_{ii} \quad (10)$$

Entonces, ya que A es simétrica y definida positiva, se puede asegurar que los elementos sobre la diagonal de L son positivos; de esta manera, los restantes elementos son reales.

Teorema de convolución

Cuando se trata de hacer un procesamiento digital de señal, no tiene sentido hablar de *convoluciones* aplicando estrictamente la definición, dado que solo se dispone de valores en instantes discretos de tiempo. Por tanto, es necesario una aproximación numérica. Para realizar la convolución entre dos señales, se evaluará el área de la función. Y se debe disponer de muestreos de ambas señales en los instantes de tiempo, que se llamará y , donde n y k son enteros (Oppenheim, Schaffer y Buck, 1997). La convolución discreta se determina por un intervalo de muestreo:

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] \quad (11)$$

Filtro LMS

Este filtro consiste en la obtención de coeficientes que permiten adquirir el valor esperado mínimo del cuadrado de la señal de error, definida como la diferencia entre la señal deseada $[d(n)]$ y la señal producida a la salida del filtro $[y(n)]$. Una característica importante del LMS es su simplicidad, pues no requiere el uso de funciones complejas (Jones, 2008).

Si fuera posible obtener medidas exactas del vector gradiente $\nabla\zeta(n)$ en cada iteración n y se dispusiera del parámetro μ adecuadamente elegido, el vector de pesos del filtro convergería a la solución óptima que se obtuvo en el filtro de Wiener. Pero en la realidad no se dispone de estas medidas exactas del vector gradiente, ya que no se conoce la matriz de autocorrelación de la señal de entrada al filtro ni el vector de correlación cruzada entre esta señal de entrada al filtro y la respuesta deseada. Por tanto, el vector gradiente ha de ser estimado a partir de los datos.

Esta estimación puede verse como el resultado de aplicar el operador gradiente ∇ al error instantáneo $|e(n)|^2$. Si se sustituye la estimación del vector gradiente en la ecuación de actualización de los pesos utilizada en el algoritmo *steepest-descent*:

$$w(n+1) = w(n) - (1/2)\mu(-\nabla\zeta(n)) \quad (12)$$

Se obtiene la relación recursiva siguiente:

$$w(n+1) = w(n) + \mu x(n) [d^*(n) - x^H(n)w(n)] \quad (13)$$

Se puede expresar esta solución con las relaciones siguientes (Vera, 2008):

$$\text{Salida del filtro:} \quad y(n) = w^H(n)x(n) \quad (14)$$

$$\text{Estimación del error:} \quad e(n) = d(n) - y(n) \quad (15)$$

$$\text{Adaptación de los pesos del filtro:} \quad w(n+1) = w(n) + \mu x(n)e^*(n) \quad (16)$$

En cada iteración, la actualización de un peso requiere dos multiplicaciones complejas y una suma. Por tanto, para un filtro de p coeficientes, se realizan $2p + 1$ multiplicaciones complejas y $2p$ sumas complejas por iteración. Es decir, la carga computacional es del orden de p [$O(p)$].

Implementación de los algoritmos

Filtro Wiener

Como se puede observar en la ecuación (6), para encontrar los coeficientes óptimos del filtrado Wiener se debe, primero, hallar la matriz de autocorrelación Toeplitz hermítica y el vector de correlación cruzada entre la señal deseada $d(n)$ y la señal recibida $x(n)$; y, posteriormente, deberá realizarse el filtrado entre dichos coeficientes y $x(n)$. Para llevar a cabo este proceso, se debe conocer además el número de muestras n y seleccionar el grado del filtro p , que en este caso se eligieron experimentalmente en 250 y 40, respectivamente. En la figura 2 se presentan, de forma general, las etapas necesarias para la implementación del filtro Wiener.

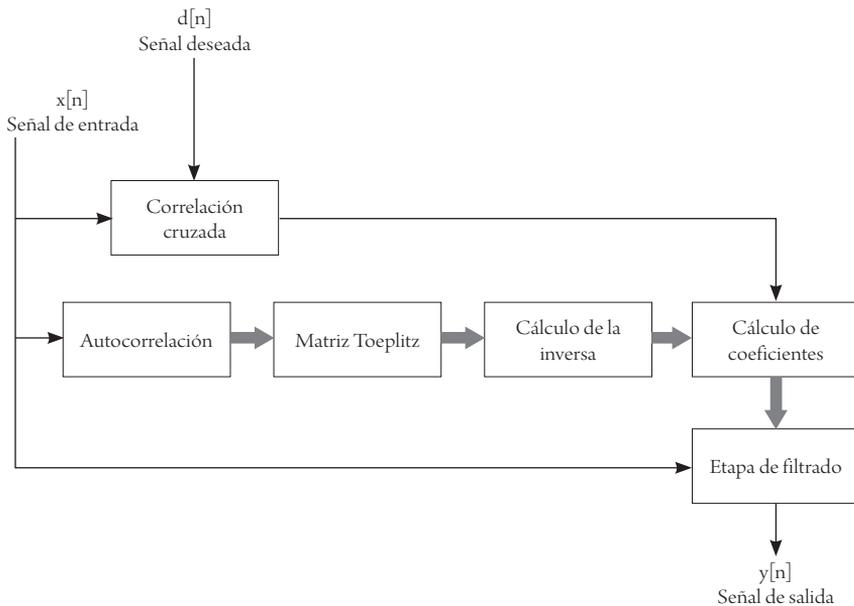


Figura 2. Diagrama de bloques del filtro Wiener

Fuente: elaboración propia.

Correlación cruzada. Para realizar la correlación cruzada entre la señal deseada y la señal de entrada, se elaboró un algoritmo basado en la ecuación (7).

Autocorrelación. También es necesario utilizar la función *correlación* para determinar la matriz de autocorrelación Toeplitz. En este caso, ya que la autocorrelación se de-

fine como la correlación cruzada de la señal consigo misma, se utiliza un algoritmo similar al de la función anterior; pero esta vez, la señal deseada es reemplazada por la señal de entrada.

Matriz Toeplitz. En esta etapa se emplea la salida de la función *autocorrelación*, pero solo son seleccionados ciertos elementos, dependiendo del grado del filtro. Para ilustrar esta etapa se utiliza la ecuación (8).

Cálculo de matriz inversa. Para calcular los coeficientes del filtro, además de calcular la matriz de autocorrelación Toeplitz, se hace necesario encontrar su matriz inversa, ya que al despejar la ecuación (6) se tiene que $w = R_x^{-1}r_{dx}$. Entonces, para llevar a cabo dicha operación se empleó, en una primera instancia, la factorización Cholesky y, luego, una serie de sustituciones para el cálculo de la matriz inversa requerida:

- *Factorización Cholesky.* Esta factorización se deriva de la factorización LU y se pretende encontrar una matriz triangular inferior (matriz L) y una matriz triangular superior (matriz U). Estas matrices serán usadas en la siguiente subetapa para el cálculo de la matriz inversa. Las ecuaciones (9) y (10) son las usadas para dicha descomposición.
- *Inversa.* En esta subetapa se realiza una serie de sustituciones llamadas *hacia adelante* y *hacia atrás*. Se expone un método para la inversión de matrices a través de una previa descomposición LU.

Cálculo de coeficientes. En esta etapa, simplemente se realiza una multiplicación de matrices o, más exactamente, entre un vector y una matriz.

Etapa de filtrado. Luego de obtener los coeficientes del filtro, solo queda aplicarlos a la entrada $x(n)$ para obtener la salida del filtro $y(n)$. Dicho procedimiento se puede observar en la figura 3 y se basa en la ecuación (11).

Filtro LMS

En la figura 4 se observa un diagrama de bloques del algoritmo del filtro LMS.

El diseño de este algoritmo se basó en las ecuaciones (14), (15) y (16), las cuales son necesarias para conocer la salida del filtro, estimar el error y calcular los coeficientes, respectivamente. Una mejor forma de ilustrar este algoritmo es por medio de un diagrama de flujo, como el que se muestra en la figura 5.

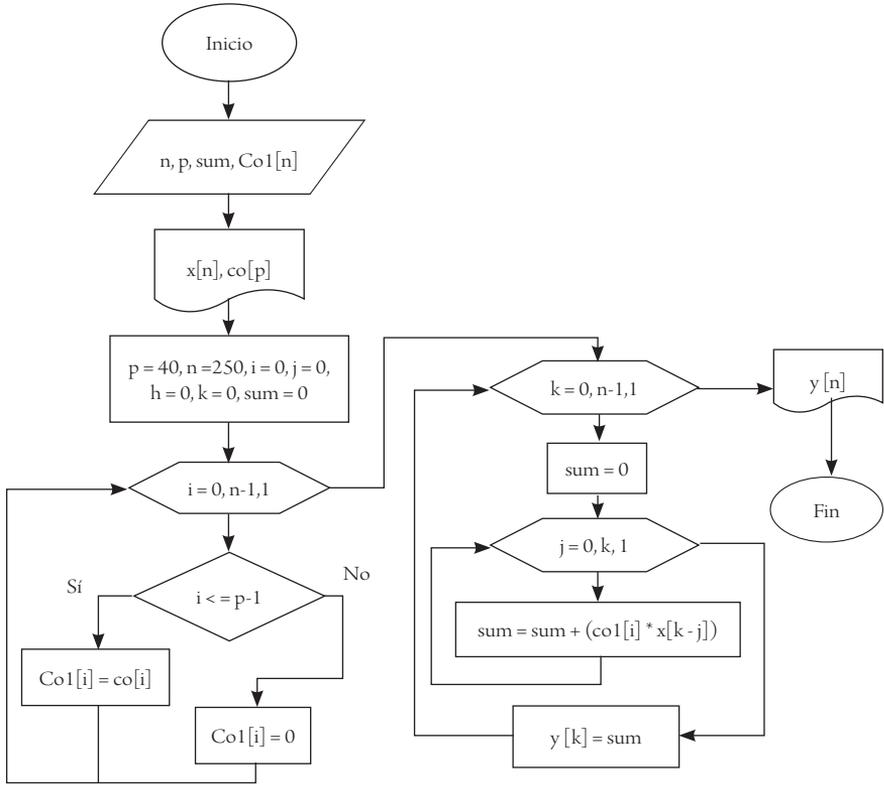


Figura 3. Diagrama de flujo etapa de filtrado

Fuente: elaboración propia.

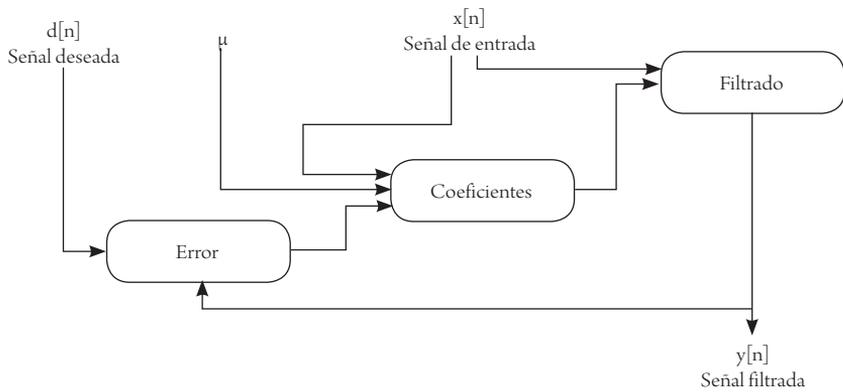


Figura 4. Diagrama de bloques del algoritmo LMS

Fuente: elaboración propia.

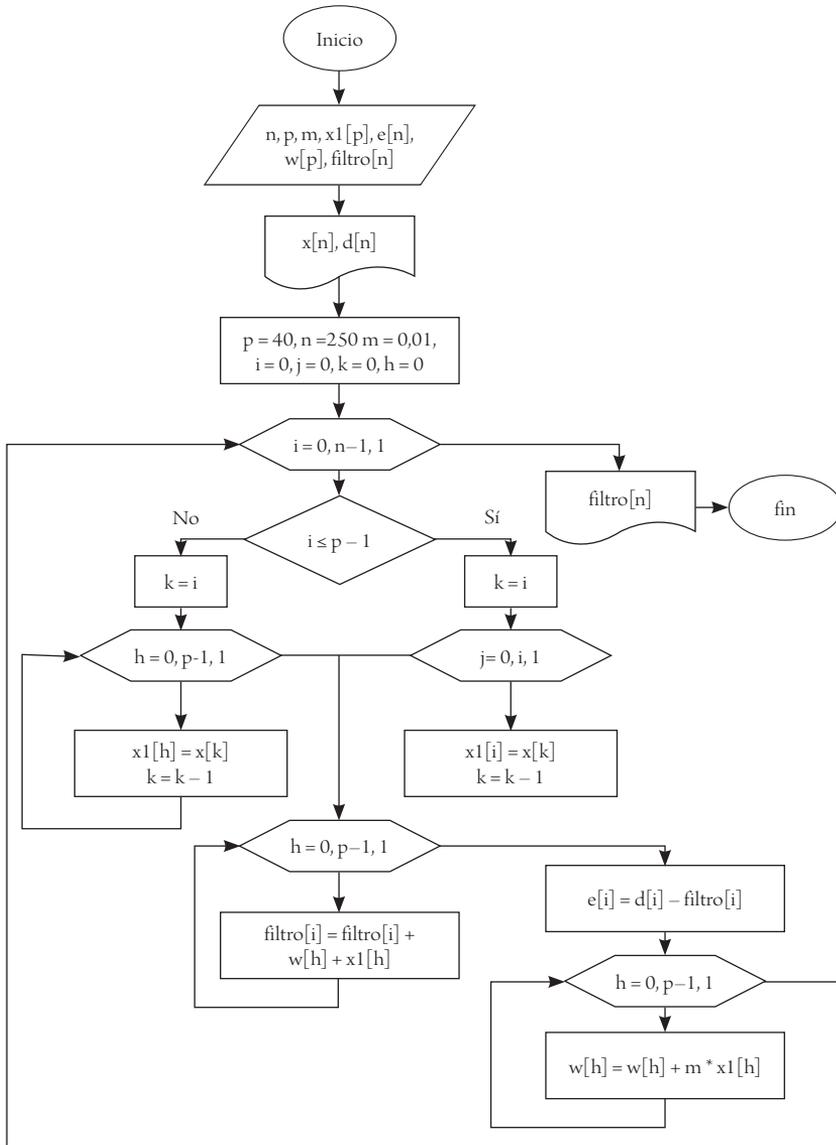


Figura 5. Diagrama de flujo del filtro LMS

Fuente: elaboración propia.

A diferencia del algoritmo del filtro Wiener, este tipo de filtrado no requiere varias funciones para poder realizar el filtrado. El filtrado LMS solo depende de la velocidad de convergencia (μ) y de las señales de entrada $x(n)$ y deseada $d(n)$. Además,

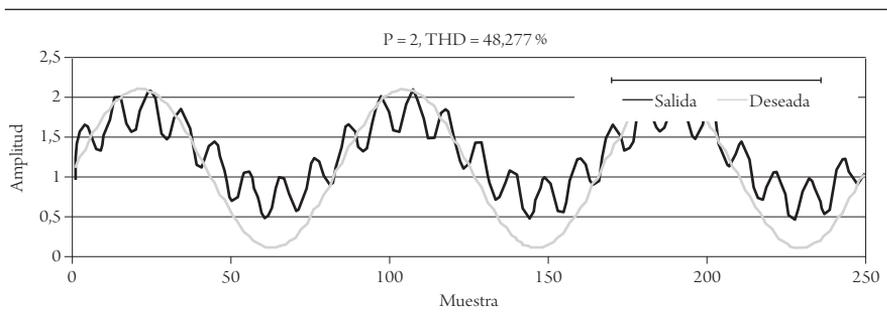
el número de muestras n y el grado del filtro p , que también se seleccionaron con 250 y 40, respectivamente.

Para hallar la salida del filtro LMS, se debe partir del vector de pesos $w(n)$. En esta medida, es necesario conocer su valor inicial $w(0)$, que en el caso del diseño propuesto corresponde a un valor igual a cero. Ahora, para calcular los valores actuales de los pesos es necesario estimar el error $e(n)$, comparando la salida del filtro $y(n)$ con la señal deseada $d(n)$ y, finalmente, ajustar el valor de los pesos del filtro, todo esto asignando a μ un valor determinado.

Determinación de parámetros

Orden del filtro

Inicialmente, se realizaron pruebas para conocer un valor adecuado del orden del filtro p , con un rango de valores, donde N es el número de muestras. Se asignó a N un valor de 250, con el fin de observar varios periodos de una señal de baja frecuencia. Estas pruebas se realizaron con el código del filtro Wiener, donde la entrada corresponde a una señal seno de baja frecuencia (3 Hz) mezclada con una señal de interferencia; en este caso, una señal seno de mayor frecuencia (24 Hz). En la figura 6 es posible notar que a medida que aumenta el valor del orden del filtro, también mejora la calidad de la señal filtrada. En esta medida, se decide tomar un orden de filtro 40, ya que realiza un filtrado razonable. El criterio tomado para esta selección correspondió a una distorsión armónica total (THD) menor al 5%.



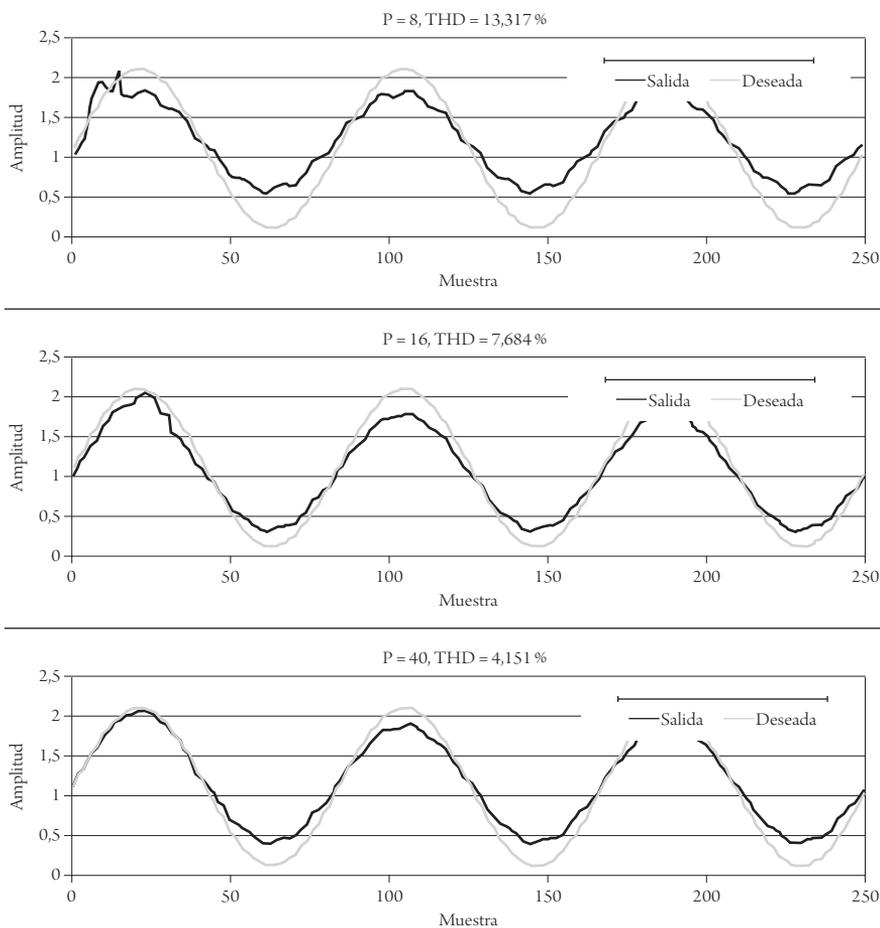


Figura 6. Filtro Wiener (orden: $2 < P < 40$)

Fuente: elaboración propia.

Velocidad de convergencia

Además del orden del filtro, definido en la sección anterior, para realizar el filtrado LMS es necesario conocer el valor de la velocidad de convergencia (μ) óptimo. Para ello, se realiza un análisis gráfico mostrado en la figura 7, donde se obtiene que para el rango entre $0,002 < \mu < 0,007$. A medida que μ aumenta, la señal de salida se asemeja cada vez más a la señal deseada.

A partir de lo anterior se le asigna a la velocidad de convergencia μ un valor de 0,006.

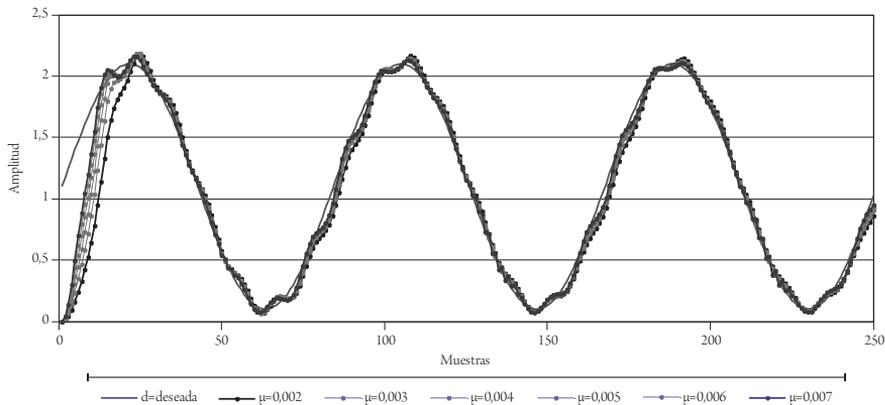


Figura 7. Rango $0,002 < MU < 0,007$ (salida del filtro LMS)

Fuente: elaboración propia.

Resultados

El *hardware* de adquisición de datos implementado en un sistema PSoC™ y el *software* LabView™ son herramientas que permiten la manipulación de los filtros adaptativos programados. Este *software* es una plataforma con un lenguaje de programación gráfico que permite diseñar un conjunto de señales para evaluar la eficiencia de los filtros implementados. En la figura 8 se observa el panel frontal elaborado en LabView™. En este se pueden observar dos secciones: en la parte izquierda, la configuración de las siguientes características: señal deseada, tipo de ruido, tipo de filtro y puerto serial; y en el lado derecho se observan las señales de entrada y el resultado de la señal filtrada.

Después de la implementación del diseño en el sistema PSoC, ya con los parámetros N y p definidos, se realizó una serie de pruebas en dos escenarios: el primero de ellos, tomando como señal de entrada la suma de la señal deseada con una señal seno conocida; y el segundo, con un ruido blanco uniforme. Posteriormente, y mediante simulaciones en el *software* Matlab®, se realizó una serie de comparaciones y análisis para evaluar el desempeño de los filtros.

En un primer escenario se selecciona como ruido una señal seno de mayor frecuencia y menor amplitud que la señal deseada. En la figura 9 se observa la señal de

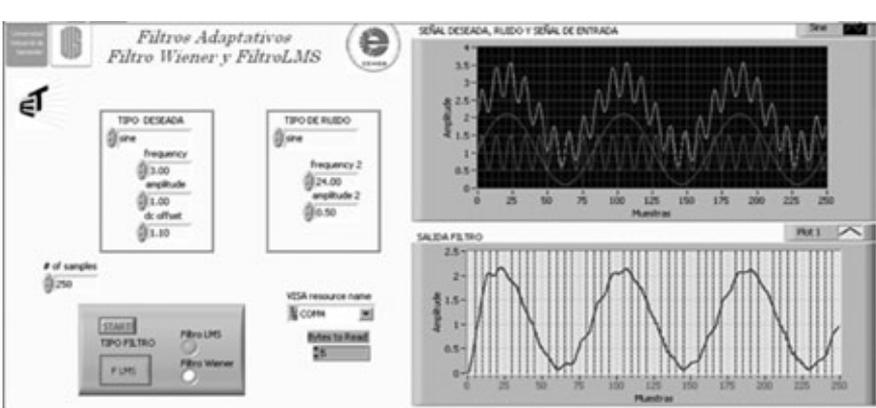


Figura 8. Aplicación para la evaluación de los filtros

Fuente: elaboración propia.

salida del filtro Wiener y del filtro LMS, donde, por inspección, se observa la eliminación de la componente de continua y la atenuación del ruido. Para una segunda prueba, se mantienen las mismas características de la señal deseada, pero esta vez se usa un ruido blanco uniforme de amplitud 0,5. Después de realizar el filtrado de la señal de entrada en el sistema PSoC, se obtienen los resultados de la figura 10.

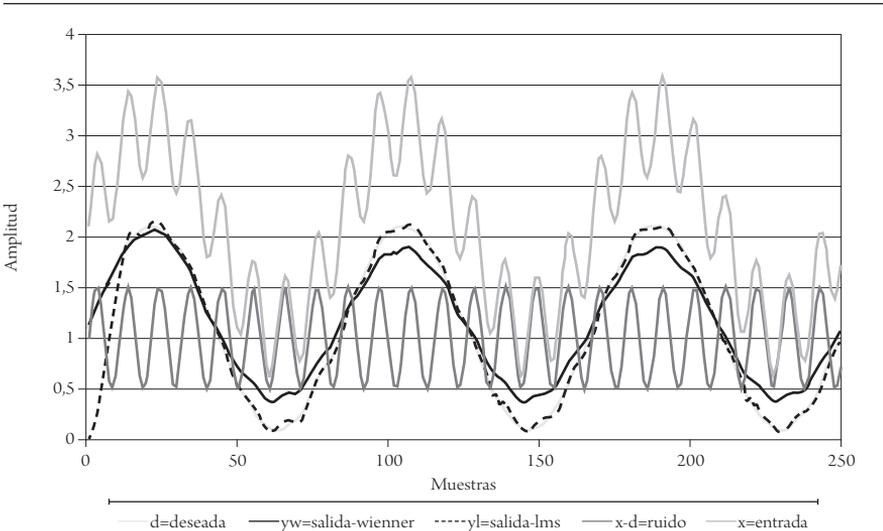


Figura 9. Salida de los filtros Wiener y LMS (ruido sinusoidal)

Fuente: elaboración propia.

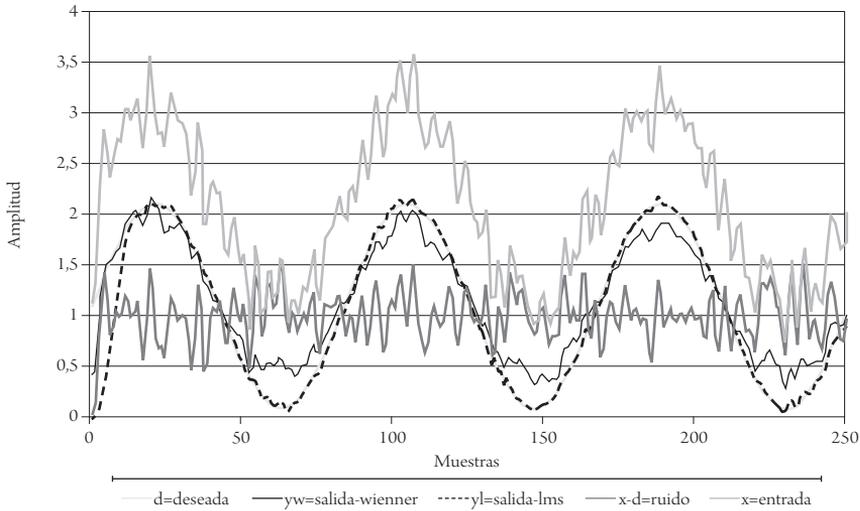


Figura 10. Salida de los filtros Wiener y LMS (interferencia de ruido blanco uniforme)

En la figura 9, el tipo de ruido es conocido y corresponde a una señal seno. El filtro Wiener tiene menor amplitud, pero su forma se asemeja más a la señal deseada. Caso contrario es el del filtro LMS, ya que la amplitud es similar, pero tiene mayores variaciones en la forma de la señal.

Por otra parte, al cambiar el tipo de ruido (figura 10) se obtienen mejores resultados con el filtro LMS, mientras que el filtro Wiener parece guardar las mismas proporciones en amplitud, pero posee una mayor distorsión.

Para medir la eficiencia de los filtros y verificar su rendimiento, se toman las siguientes medidas de desempeño: el error de identificación, el error en la estimación y el porcentaje de error (García, 2008). El error de identificación representa una medida del error en la salida del sistema adaptable con respecto a la salida deseada en presencia de ruido. Se define por la ecuación (17):

$$error = 10 \log_{10} \left[\frac{\sum_{n=1}^N (d(n) - y(n))^2}{\sum_{n=1}^N (d(n) - r(n))^2} \right] [dB] \quad (17)$$

Donde N es el número de muestras, es decir, 250.

El error en la estimación también es una medida de desempeño que está dado por la diferencia entre la señal deseada y la señal estimada. Se define por la ecuación (15). El porcentaje de error está dado por el cociente entre el promedio del valor absoluto del error (la diferencia entre la señal deseada y estimada) y la amplitud de la señal deseada.

Los resultados de dichos errores ante el ruido correspondiente a la interferencia seno y al ruido blanco uniforme se pueden observar en las tablas 1 y 2, respectivamente.

Tabla 1. Errores de identificación y estimación (ruido: señal seno)

	FILTRO WIENER	FILTRO LMS
Error de identificación	-13,0463 [dB]	-12,1466 [dB]
Promedio del error de estimación	-0,0498	0,0350
% de error	14,95	6,73

Tabla 2. Errores de identificación y estimación (ruido: blanco uniforme)

	FILTRO WIENER	FILTRO LMS
Error de identificación	-11,2411 [dB]	-11,8263 [dB]
Promedio del error de estimación	0,0478	0,0410
% de error	17,0	6,71

Conclusiones

- Una característica importante del algoritmo del filtro adaptativo LMS encontrada fue su sencillez y simplicidad, tanto en los requisitos de memoria como en la capacidad computacional, ya que no requiere de las funciones de correlación, inversión de matrices, entre otras. Pero uno de sus aspectos determinantes por tener en cuenta es su dependencia de la velocidad de convergencia (μ), por lo que es necesario fijar su valor teniendo en cuenta el error de salida del filtro y su rapidez de adaptación, para, de esta manera, elegir un valor óptimo y eficiente.
- El error de identificación y el promedio del error de estimación del filtro Wiener (-11,2411 [dB] y 0,0478, respectivamente) son mayores a los del filtro LMS (-11,8263 [dB] y 0,0410, respectivamente) tanto en las pruebas realizadas con el ruido blanco como en las pruebas con el ruido seno de mayor frecuencia.

Esto lleva a concluir que el filtrado LMS, además de tener menores tiempos de cómputo (casi cinco veces menor al filtro Wiener), arrojó mejores resultados, y su señal de salida se acercó un poco más a la señal de referencia. Aunque la salida del filtro Wiener se caracterizó por una curva más suave y con menores sobresaltos que la señal de salida LMS, esta última se acercó más en amplitud a la señal deseada.

- Los sistemas embebidos PSoC ofrecen una alternativa eficiente en el procesamiento de señales, dado que, además de la capacidad de contener algoritmos complejos, brindan la posibilidad de realizar un acondicionamiento análogo de la señal, porque internamente posee módulos de amplificación, comparación y filtrado, entre otros.

Referencias

- Augusto, C. y Meza, A. (2009). *Implementation of the algorithm for noise cancellation on speech using the ARM LPC2378 processor*. Växjö, Suecia: Växjö University.
- Barajas González, J. E. (2005). *Procesamiento digital de señal electrocardiográfica con tecnología DSP orientado al análisis de variabilidad de la frecuencia cardiaca*. Bucaramanga: Universidad Industrial de Santander.
- Barbosa Suárez, J. A. y Murcia Sepúlveda, N. (2013). *Diseño e implementación de filtros adaptativos mediante un sistema de desarrollo PSoC*. Bucaramanga: Universidad Industrial de Santander.
- Blanco, E. F. y Ruiz, J. (2009). Implementación de filtros digitales en FPGA. *Bioingeniería y Física Médica Cubana*, 10(2), 9-13.
- Borisagar, K. y Kulkarni, G. R. (2010). Simulation and comparative analysis of LMS and RLS algorithms using real time speech input. *Global Journal of Researches in Engineering*, 10(5), 44-47.
- Chapra, S. C. y Canele, R. P. (2007). Descomposición LU e inversión de matrices. En *Métodos numéricos para ingenieros* (5.ª ed.; pp. 284-299). México: McGraw-Hill.
- Durán, C. A. (2010). *Implementación del algoritmo NLMS en un DSP*. México: Instituto Politécnico Nacional.
- García Hernández, M. (2008). *Propuesta y evaluación de un algoritmo de filtrado adaptable*. Iztapalapa, México: Universidad Autónoma Metropolitana Iztapalapa.
- Geoffrey, A. (2010). Adaptive IIR filters. En *Digital Signal Processing Fundamentals* (pp. 1-21). Nueva York: Taylor and Francis Group.

- Gray, R. M. (2005). Toeplitz and Circulant matrices: a review. *Foundations and Trends in Communications and Information Theory*, 2(3), 155–239. Doi:10.1561/0100000006
- Iglesias Martínez, M. E. (2012). *Técnicas de filtrado adaptativo y estadística de orden superior para la implementación de algoritmos de cancelación de ruido sobre una arquitectura reconfigurable*. Recuperado de <http://site.ebrary.com/id/10609448?ppg=24>
- Jones, D. L. (2008). Adaptive filters. *Solutions*. Recuperado de <http://cnx.org/content/col10280/1.1/>
- Manolakis, D. G., Ingle, V. K. y Kogon, S. M. (2005). *Statistical and adaptive signal processing*. Recuperado de http://www.lr.ttu.ee/~ameister/materjale/Dig_spe_anal/Statistical%20and%20Adaptive%20Signal%20Processing.pdf
- Oppenheim, A. V., Schafer, R. W. y Buck, J. R. (1997). The convolution theorem. En *Discrete-time signal processing* (2.^a ed.; pp. 60-61). Nueva Jersey: Prentice-Hall.
- Physionet (2013). Filtrado lineal óptimo. Recuperado de <http://physionet.cps.unizar.es/~eduardo/docencia/tds/librohtml/wiener1.htm#filtros>
- Vera, M, Y. (2008). *Cancelación activa de ruido*. Bogotá: Universidad de Los Andes.

