



**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL ACTIVO  
DE RUIDO ACÚSTICO DE BAJA FRECUENCIA PARA COMPRESOR  
NEUMÁTICO MARSHAL INDUCTIVE MOTOR 931373**

**CRISTIAN ALBERTO FRANCO CUBILLOS  
OSCAR DAVID SÁNCHEZ BOCANEGRA**

**UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS  
FACULTAD TECNOLÓGICA  
INGENIERÍA EN CONTROL  
BOGOTÁ, D.C.  
2015**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL ACTIVO DE  
RUIDO ACÚSTICO DE BAJA FRECUENCIA PARA COMPRESOR NEUMÁTICO  
MARSHAL INDUCTIVE MOTOR 931373**

CRISTIAN ALBERTO FRANCO CUBILLOS  
OSCAR DAVID SÁNCHEZ BOCANEGRA

Monografía de Grado  
Para optar al título de  
Ingeniero en Control

Director de Proyecto  
Ing. Miguel Pérez Pereira

**UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS  
FACULTAD TECNOLÓGICA  
INGENIERÍA EN CONTROL  
BOGOTÁ, D.C.  
2015**

## INDICE DE CONTENIDOS

LISTADO DE TABLAS Y FIGURAS.....	i
LISTADO DE ABREVIATURAS.....	iv
PÁGINA DE APROBACIÓN.....	vi
1. INTRODUCCIÓN.....	1
1.2 OBJETIVOS.....	2
2. CONTROL ACTIVO DE RUIDO.....	3
2.1 CONTROL DE RUIDO POR MÉTODOS PASIVOS.....	3
2.2 PRINCIPIO DE YOUNG.....	4
2.3 CONTROL ACTIVO DE RUIDO.....	6
2.4 SISTEMAS DE CONTROL.....	6
2.5 PROCESAMIENTO DE SEÑALES (Filtrado FIR).....	8
3. METODOLOGÍA.....	11
3.1 ALGORITMOS DE CONTROL.....	12
3.1.1 FILTRO ADAPTATIVO DE LA MÁXIMA PENDIENTE.....	12
3.1.2 LMS MEDIOS MÍNIMOS CUADRÁTICOS.....	14
3.1.3 LMS NORMALIZADO .....	15
3.1.4 LEAKY LMS.....	17
3.1.5. ALGORITMO FILTERED-X LMS.....	17
3.1.6. ARITMÉTICA DE PUNTO FIJO.....	18
3.1.7. LIMITACIONES EN TIEMPO REAL.....	19
3.1.8. CONTROL FEEDFORWARD.....	20
3.1.9 ALGORITMOS PARA SISTEMAS DE BANDA CORTA (filtro Notch).....	24
3.1.10. ALGORITMO GENÉTICO.....	24

3.2 ELECCIÓN DEL TRANSDCUTOR.....	25
3.3 ETAPA DE ACONDICIONAMIENTO.....	28
3.3.1 CONSOLA MIXER 8 CANALES BEHRINGER XENYX 802....	29
3.4 REALIZACIÓN DE ESTRUCTURA PARA EL COMPRESOR.....	30
3.5 CARACTERIZACIÓN DE LA SEÑAL DE RUIDO ACÚSTICO DEL COMPRESOR NEUMÁTICO MARSHAL INDUCTIVE MOTOR 931373.....	31
3.6 DESARROLLO DEL SISTEMA DE CONTROL DE RUIDO ACÚSTICO DE BAJA FRECUENCIA.....	36
3.6.1 ESCENARIO DE ATENUACIÓN.....	37
3.6.2 LOCALIZACIÓN DE SENSORES.....	39
3.6.2.1 LOCALIZACIÓN DEL SENSOR DE ERROR.....	39
3.6.2.2 LOCALIZACIÓN DEL SENSOR DE REFERENCIA.....	41
3.6.3 ELECCIÓN DEL ALTAVOZ DE SALIDA.....	42
3.6.4 ELECCIÓN DEL DSP.....	43
3.6.4.1 CODE COMPOSER STUDIO V6.....	45
3.6.5 DISEÑO DEL ALGORITMO ADAPTATIVO.....	46
3.6.5.1. ALGORITMO DE MINIMIZACIÓN DE ERROR: FILTRO ADAPTATIVO FIR LMS.....	47
3.6.5.2. Algoritmo FILTERED-X LMS (FxLMS).....	50
3.6.5.2.1 IDENTIFICACIÓN DEL CAMINO SECUNDARIO.....	50
3.6.5.2.2 IMPLEMENTACIÓN DE CONTROL ACTIVO DE RUIDO FEEDFORWARD.....	53
3.6.5.3. ALGORITMOS PARA SISTEMAS DE BANDA CORTA (Filtro Notch).....	57
3.6.5.4. ALGORITMO GENÉTICO.....	59

4.	RESULTADOS.....	60
4.1	INTERFAZ.....	60
4.2	PARÁMETROS DEL ALGORITMO ADAPTATIVO DE CONTROL...64	
4.3	PRUEBA DE CONTROL CON PARLANTES DE COMPUTADOR...65	
4.4	PRUEBA DE CONTROL CON RUIDO DEL COMPRESOR.....67	
4.5	FXLMS.....	68
4.6	FXLMS-NA.....	70
4.7	FXNLMS.....	71
4.8	ALGORITMO ADAPTATIVO NOTCH.....	72
4.9	ALGORITMO GENÉTICO.....	71
5.	CONCLUSIONES.....	75
	BIBLIOGRAFÍA.....	77
	ANEXOS.	

## LISTADO DE TABLAS Y FIGURAS

Tabla 3.1. Especificaciones micrófono condensador Behringer ECM8000.

Tabla 3.2. Velocidad del sonido en diferentes medios y temperaturas. (Tomada de Pérez Vega, 2001).

Tabla 4.1. Resultados de atenuación obtenidos con los diferentes algoritmos de prueba.

Tabla 4.2. Resultados de atenuación obtenidos con los diferentes algoritmos implementados para el sistema de control de ruido acústico.

Figura 2.1. Recinto acústico aislado con barrera acústica.

Figura 2.2. Principio de Young. Interferencia de ondas generadas a través de dos orificios.

Figura 2.3. Atenuación de la presión acústica en un punto por interferencia destructiva de ondas de presión acústica en oposición de fase. Variaciones de atenuación vs. Diferencias de fase y de amplitud.

Figura 2.4. Principio de cancelación de ruido, interferencia de ruido y antiruido.

Figura 2.5 Elementos de un sistema de control activo de ruido.

Figura 2.6. Estructura de un filtro digital tipo FIR.

Figura 2.7. Diagrama de bloques de los filtros FIR en el dominio del tiempo y el dominio de  $z$ .

Figura 3.11 Diagrama de bloques de un filtro adaptativo.

Figura 3.12. Un ejemplo del MSE  $\mathcal{E}(n) = E\{e(n)^2\}$  de los dos pesos del filtro se ilustra por el gráfico de superficie a la izquierda y el gráfico de contorno a la derecha. La dirección del descenso más rápido apunta hacia el mínimo de la superficie de error (opuesta a la del gradiente).

Figura 3.13. Diagrama de bloques del algoritmo Filtered-x LMS.

Figura 3.14. Control Feedforward, lazo abierto.

Figura 3.15. Diagrama de bloques del control Feedforward.

Figura 3.16. Diagrama de bloques sin realimentación acústica y con la función de transferencia entre la señal de ruido  $R(z)$  y el micrófono de referencia igual a uno.

Figura 3.17. Frecuencia fundamental y múltiplos de la frecuencia fundamental que contienen la mayor parte de la energía de la señal.

Figura 3.21. Respuesta en frecuencia del micrófono Behringer ECM8000 según el fabricante.

Figura 3.31. Consola Mixer 8 Canales Behringer Xenyx 802.

Figura 3.32. Características de la consola Mixer 8 Canales Behringer Xenyx 802.

Figura 3.41 Estructura con material triplex de 1.8cm para el compresor.

Figura 3.51. Análisis en frecuencia de la señal del compresor sin la estructura. A) mayor rango de frecuencias. B) zoom en frecuencias inferiores a 500Hz.

Figura 3.52. Análisis en frecuencia de la señal del compresor con la estructura.

Figura 3.53. Perfil armónico del compresor con la estructura. A) después de 5 minutos. B) Después de 9 minutos.

Figura 3.61. Configuración del sistema de control en el escenario de atenuación.

Figura 3.62. Configuración del sistema de control en el escenario real de atenuación.

Figura 3.63. Disminución de la potencia acústica emitida por la fuente de ruido en función de la distancia en longitudes de ondas con la fuente de control.

Figura 3.64. Retardos de la señal a 4, 15 y 32cm.

Figura 3.65. Señal del compresor y la señal opuesta generada con un algoritmo adaptativo, comprobando así, la acertada ubicación de los micrófonos.

Figura 3.66 Amplificador de potencia. (Fuente secundaria de control).

Figura 3.67 Respuesta en frecuencia del amplificador de potencia. (Fuente secundaria de control).

Figura 3.68. Arquitectura CPU familia C6000 de Texas Instruments.

Figura 3.69. Características y distribución de los componentes de la tarjeta TMS320C6416.

Figura 3.610. Generación de código mediante software code composer studio 6.1

Figura 3.611. Diagrama de bloques de la configuración general de un filtro adaptativo.

Figura 3.612. Diagrama de flujo Filtro adaptativo FIR LMS.

Figura 3.613. Diagrama de bloques del sistema de control en lazo abierto feedforward con todas las funciones de transferencia involucradas.

Figura 3.614. Diagrama de bloques sistema feedforward simplificado.

Figura 3.615. Diagrama de bloques sistema feedforward simplificado.



Figura 3.616. Diagrama de bloques de identificación del camino secundario.

Figura 3.617. Diagrama de flujo camino secundario.

Figura 3.618. Diagrama de bloques, Sistema de control Feedforward analizado.

Figura 3.619. La identificación realizada con la interfaz gráfica implementada.

Figura 3.620. Diagrama de flujo FxLMS.

Figura 3.621. Interfaz gráfica del algoritmo FxLMS.

Figura 3.622. Convergencia del algoritmo de identificación con el programa Code Composer Studio v6.1.

Figura 3.623. Diagrama de bloques algoritmo con filtro Notch.

Figura 4.1 En la interfaz se muestran los parámetros más importantes a configurar.

Figura 4.2. Algoritmo FxLMS en la interfaz.

Figura 4.3. Algoritmo Notch en la interfaz.

Figura 4.4. Algoritmo de máxima pendiente optimizado con algoritmo genético en la interfaz.

Figura 4.5. Resultados de atenuación con prueba de parlantes.

Figura 4.6 Cancelación Notch con parlantes, donde la señal CH1 es de control y la señal CH2 es la señal de error.

Figura 4.7. Parlantes sp-p110, utilizados para simular la fuente de ruido.

Figura 4.8. Gráfico de atenuación de acuerdo a los diferentes algoritmos probados.

Figura 4.9. Resultado del algoritmo FXLMS. CH1 es la señal de error y CH2 la señal de control.

Figura 4.10. Resultado del algoritmo FXLMS generando una onda de 120Hz. CH1 es la señal de error y CH2 la señal de control.

Figura 4.11. Resultado del algoritmo FXLMS-NA. CH1 es la señal de error y CH2 la señal de control.

Figura 4.12. Resultado del algoritmo FXNLMS. CH1 es la señal de error y CH2 la señal de control.

Figura 4.13. Resultados del algoritmo NOTCH. CH1 es la señal de error y CH2 la señal de control.

Figura 4.14. Resultados del algoritmo genético. CH1 es la señal de error y CH2 la señal de control.

## LISTADO DE ABREVIATURAS

$T$	Tiempo de muestreo.
$n$	Variable en tiempo discreto.
$X(z)$	Transformada $z$ de $x(n)$ .
$\omega$	Frecuencia Angular.
$f_s$	Frecuencia de muestreo.
$x(n)$	Señal muestreada como una función en tiempo discreto.
$d$	Señal deseada.
$e$	Señal de error.
*	Convolución.
$\beta$	Paso normalizado usado en NLMS.
$\mu$	Paso de adaptación utilizado en LMS.
$\nabla$	El gradiente. En este caso las derivadas parciales con respecto al el filtro de pesos.
$w(n)$	$[w_n(0), w_n(1), \dots, w_n(p)]^T$ Vector de pesos del filtro adaptativo.
$z$	Variable compleja, $z = re^{j\omega}$
$w_n$	Pesos del filtro adaptativo.
$x$	Señal de entrada.
$X(z)$	Señal de entrada en el dominio de $z$ .
$y$	Señal de salida.
$Y(z)$	Señal de salida en el dominio de $z$ .
$j$	Unidad imaginaria.
$p$	Orden del filtro.
<i>ANC</i>	Active Noise Control
<i>Codec</i>	Coding and decoding
<i>DSP</i>	Digital Signal Processor
<i>FIR</i>	Finite Impulse Response

*FxLMS* Filtered X Least Mean Square

*LMS* Least Mean Square

*MSE* Mean Square Error

*NLMS* Normalized Least Mean Square

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL ACTIVO DE  
RUIDO ACÚSTICO DE BAJA FRECUENCIA PARA COMPRESOR NEUMÁTICO  
MARSHAL INDUCTIVE MOTOR 931373”**

**HOJA DE APROBACIÓN**

**Observaciones**

---

---

---

---

---

---

---

---

---

**Ingeniero** **Evaluador 1**

---

**Ingeniero** **Evaluador 2**

**2015-08-10**

## 1. INTRODUCCIÓN

La contaminación auditiva es una problemática que se ha ido incrementando en el sector industrial, por lo cual se han hecho avances en técnicas que ayuden a su atenuación y control. Aunque se han desarrollado métodos pasivos para tal fin, su utilización eficaz no cubre todo el espectro de ruido, ya que para bajas frecuencias su implementación debe ser con estructuras masivas que usualmente no cumplen a cabalidad con ésta función, por lo que se han ido desarrollando nuevos métodos, como el control activo de ruido acústico.

El control activo de ruido es la atenuación de ruido acústico mediante la superposición de ondas de presión en contrafase generadas por la fuente de ruido y algún controlador; esta técnica se remonta a los años 30 cuando Paul Lueg patentó los esquemas en los que se fundamenta el control activo de ruido. Esta idea ha estado durante décadas en fase de transición permanente entre el concepto acústico, la experimentación en el laboratorio y la transición entre el laboratorio y el desarrollo de sistemas de control comerciales aplicables a cualquier escenario de cancelación acústica. En la actualidad, estos sistemas han tomado mayor relevancia debido a los adelantos tecnológicos que permiten realizar lo que hace algunos años sólo podía quedar en un contexto teórico. La exigencia del problema requiere de bastantes recursos a nivel computacional para ejecutar una solución fiable.

En este proyecto se abordó esta temática y se continuó con la línea de investigación de las técnicas de controlar y reducir el ruido externo al que está expuesto un trabajador durante su actividad laboral, ya sea proveniente de la máquina que opera, o del entorno que lo rodea. Aprovechando los adelantos de los procesadores digitales de señales (DSP) se aplicará control activo de ruido acústico de baja frecuencia en el compresor neumático Marshal inductive motor 931373, el cual es una fuente de ruido utilizada con frecuencia en el sector mecánico, buscando la optimización de los sistemas de control activo y el manejo de los principios acústicos de estos, que representará, en últimas, el éxito del proyecto.

## **1.2. OBJETIVOS**

### **General.**

Diseñar e implementar un sistema de control activo de ruido de baja frecuencia, para el compresor neumático Marshal Inductive Motor 931373, el cual atenúe los ruidos nocivos que se produzcan durante su utilización en el sector industrial.

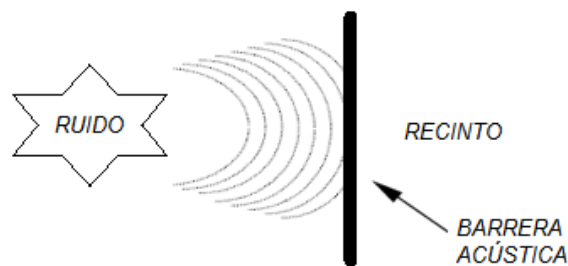
### **Específicos.**

- Caracterizar la señal de ruido del compresor Marshal Inductive Motor 931373.
- Desarrollar un sistema de control que atenúe la señal de ruido no deseado del compresor Marshal Inductive Motor 931373, en un punto en el espacio.
- Realizar las pruebas objetivas para comprobar el funcionamiento del sistema de control y los niveles de atenuación de ruido del compresor Marshal Inductive Motor 931373.

## 2. CONTROL DE RUIDO

### 2.1. CONTROL DE RUIDO POR MÉTODOS PASIVOS.

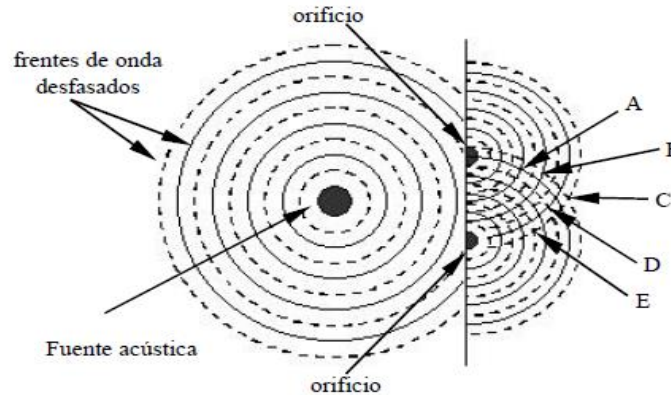
Cuando se desea atenuar o eliminar el ruido que genera una fuente acústica o más inmediato es actuar allí donde se encuentra dicha fuente, y en caso de no ser posible, actuar entonces sobre el lugar donde se requiere una disminución de ruido. En ambos casos, es necesario una barrera que impida el paso de la energía acústica desde la fuente al recinto de interés (figura 2.1).



**Figura 2.1. Recinto acústico aislado con barrera acústica.**

### 2.2. PRINCIPIO DE YOUNG.

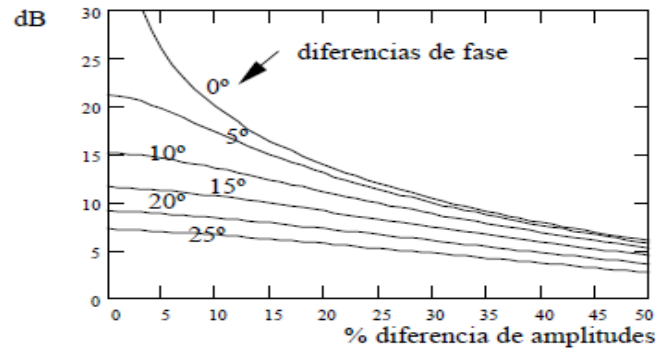
La interferencia es un fenómeno característico de todo tipo de ondas que transportan alguna energía en un determinado medio de propagación y que se produce en regiones del medio donde se cruzan frentes de ondas diferentes. Este principio, particularizado para las ondas de presión acústica, establece que una onda de presión acústica de una determinada frecuencia, que se propaga en un medio elástico, puede cancelarse en un determinado punto de dicho medio, mediante la superposición de otra onda de presión acústica de la misma frecuencia que se encuentre en oposición de fase en ese punto. Este principio fue demostrado por Thomas Young en 1802 mediante frentes de ondas de luz generadas en dos orificios de una superficie. (Mínguez Olivares, 1998).



**Figura 2.2. Principio de Young. Interferencia de ondas generadas a través de dos orificios.**

En la figura 2.2 se muestran diferentes frentes de ondas. En los puntos A, C y E (intersección de línea continua con línea continua) las ondas procedentes de los dos orificios llegan en fase y se produce un aumento de intensidad (interferencia constructiva). En otros puntos, tales como B y D (intersección de línea continua con línea discontinua), las ondas están desfasadas 180 grados y se cancelan entre sí (interferencia destructiva). Los sistemas de control activo de ruido optimizan su funcionamiento para situar zonas de interferencia destructiva en los lugares deseados. Se generan ondas de presión acústica que cancelan a otras ondas de presión en aquellas zonas objeto de control acústico. La idea puede parecer simple pero su realización práctica no lo es en absoluto. La onda de presión acústica generada ha de ser una imagen exacta pero invertida, de la onda de presión acústica a cancelar. Si existen variaciones de amplitud, fase y, por supuesto, en frecuencia, la cancelación se pierde rápidamente. A continuación, en la figura 2.3, se presenta una gráfica que muestra la atenuación de la presión acústica que se obtiene en función de las diferencias de amplitud y fase de dos ondas de presión acústica de una determinada frecuencia. En el caso ideal la cancelación es total (4 dB). (Mínguez Olivares, 1998)



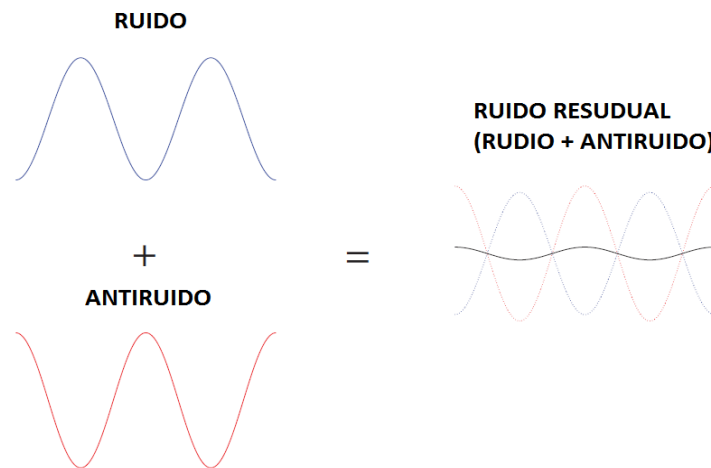


**Figura 2.3. Atenuación de la presión acústica en un punto por interferencia destructiva de ondas de presión acústica en oposición de fase. Variaciones de atenuación vs. Diferencias de fase y de amplitud.**

Se puede apreciar que para conseguir atenuaciones de presión acústica superiores a 20 dB es necesario que la variación de la fase entre las ondas sea inferior a 5 grados ( $180^\circ \pm 5^\circ$ ) y que la variación en amplitud sea inferior al 5%. Si se desea cancelar una onda de presión acústica completamente, es necesario cancelar en todos y cada uno de los puntos del medio elástico por el que se propaga dicha onda. Lógicamente, esto es prácticamente imposible, ya que cada fuente acústica tiene su propio diagrama de radiación y es imposible realizar tal superposición en todos los puntos. En situaciones reales no se fijan objetivos de cancelación tan exigentes. Generalmente, se pretenden conseguir pequeñas zonas de silencio o de atenuación, en aquellos lugares más críticos, donde una persona pueda percibir la sensación de atenuación acústica, es decir, en una zona de dimensión tal, que englobe la cabeza. Este objetivo puede cumplirse hoy en día. La dimensión máxima de esta zona virtual de cancelación, depende de la frecuencia máxima a cancelar o a atenuar. Resultados experimentales en un campo acústico difuso, de ondas de presión de una determinada frecuencia, demuestran que pueden conseguirse valores de atenuación superiores a 10 dB para una esfera espacial de diámetro la décima parte de la longitud de onda acústica. Por ejemplo, una frecuencia de 100 Hz tiene una longitud de onda de 3,4 metros y la zona de cancelación tendría unas dimensiones equivalentes a la de una esfera de diámetro unos 34 centímetros. Lógicamente, conforme se aumenta la frecuencia estas dimensiones se van haciendo más pequeñas y no resulta práctico para frecuencias superiores a 500 Hz. En cualquier caso, hay que recordar que la técnica de control activo de ruido sólo es necesaria para las bajas frecuencias, justamente para las cuales los elementos pasivos pierden efectividad. (Mínguez Olivares, 1998).

### 2.3 CONTROL ACTIVO DE RUIDO.

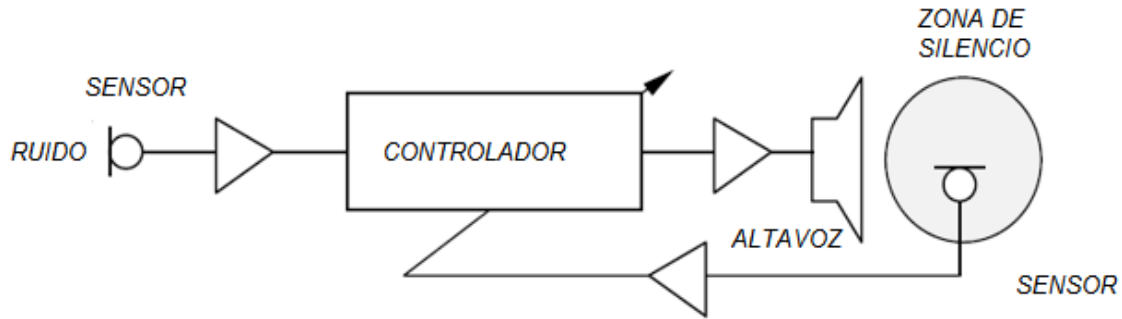
El control activo de ruido está basado en el principio de superposición. Cuando dos ondas interactúan el resultado es una forma de onda que depende de la frecuencia, la amplitud y la fase. Una cancelación total ocurrirá si el ruido y el antiruido tienen la misma fase pero en amplitud opuesta. El principal obstáculo está en encontrar una rápida, estable y apropiada solución para identificar el ruido y generar el antiruido en tiempo real.



**Figura 2.4. Principio de cancelación de ruido, interferencia de ruido y antiruido.**

### 2.4. SISTEMAS DE CONTROL.

Para poder realizar una disminución de la presión acústica en un recinto es necesario un sistema de control que adapte su respuesta, ante las posibles variaciones del ruido y del entorno acústico de cancelación. Dicho sistema de control está formado por los siguientes elementos: Sensores, actuadores, planta, controlador. Los resultados de atenuación acústica que pueden conseguirse dependen directamente de las características de los elementos que componen el sistema de control. En primer lugar, los transductores electroacústicos no son ideales y sus imperfecciones modifican la conversión electro/mecánica/acústica que se realiza en el sistema de control. (Beltrán, 1995)



**Figura 2.5 Elementos de un sistema de control activo de ruido.**

Los sensores utilizados suelen ser micrófonos y acelerómetros. Los primeros entregan una señal eléctrica función de la presión acústica que incide sobre ellos y los segundos transforman las aceleraciones de las vibraciones de los materiales en señales eléctricas. Como obviamente no son dispositivos ideales, las referencias del campo acústico sufren algún tipo de degradación (distorsión, ruido, modificación espectral en realce o atenuación, retardos, etc.). Esta degradación, influye en el funcionamiento óptimo del controlador. Las características de los sensores deben ser excelentes en el margen de frecuencias de interés ( $< 1.000$  Hz es más que suficiente), es decir, en términos de sensibilidad, bajo ruido, respuesta en frecuencia y fase. Por ejemplo, en la mayoría de las aplicaciones los micrófonos que se utilizan son de bajo coste, tipo electret, ya que sus características son suficientes para lograr una buena señal de referencia. (Elliott, Syothers, Nelson, McDonald, Quinn y Saunders, 1988)

La correlación entre el campo acústico y las señales eléctricas de los sensores determina la máxima atenuación que puede alcanzarse.

El entorno acústico de cancelación, también condiciona la dimensión máxima de la zona de atenuación. Si el sistema a controlar, es un sistema donde se propagan ondas acústicas planas, por ejemplo en un tubo, se obtienen resultados bastante buenos de cancelación para grandes distancias. Sin embargo, para el caso de ondas acústicas esféricas se reduce drásticamente la zona de control, a unos márgenes de la décima parte de la longitud de onda acústica, entre 7 y 34 cm para frecuencias entre 500 y 100 Hz respectivamente. (Mínguez Olivares, 1998).

El diseño de los controladores es bastante complejo ya que deben ser capaces de seguir los cambios acústicos que se puedan producir a lo largo del tiempo (nivel, frecuencia, fase, función de transferencia acústica del recinto, etc.). La implementación de un controlador con

tales exigencias, sólo es posible mediante algoritmos digitales adaptativos, desarrollados sobre un procesador digital de señal. En estos dispositivos se puede implementar cualquier sistema, por complejo que este sea, y además modificando sus características dinámicamente. Existen dos tipos de sistemas de control que se aplican en el control activo de ruido: control de lazo cerrado (feedback) y control de lazo abierto (feedforward). El primero de ellos sólo cancela ruido periódico y el segundo cancela tanto ruido de banda estrecha -periódico- como ruido de banda ancha (aleatorio). (Widrow y Walach, 1996).

## 2.5 PROCESAMIENTO DE SEÑALES (Filtrado FIR)

Una respuesta infinita al impulso (FIR) es un filtro de procesamiento de señales con una respuesta al impulso la cual es cero en un intervalo finito. Si el filtro es causal, en cada muestra de la secuencia de salida  $y(n)$  sólo depende del valor actual y el valor pasado de la entrada  $x(n)$ ,  $x(n - 1)$ ,  $x(n - 2)$ , etc.

Si los coeficientes de la respuesta al impulso del filtro son  $h(k) = 0$  para  $k < 0$  y  $k \geq p$ , entonces la señal de salida puede ser expresada como una combinación lineal de entradas pasadas y presentes.

$$y(n) = \sum_{k=0}^p h(k)x(n - k)$$

Donde  $p$  es el orden del filtro.

El filtro FIR también puede ser expresado como el producto punto usando notación de vectores. Por lo tanto la señal de salida  $y(n)$  puede ser expresada con el vector de forma:

$$y(n) = h^T x(n)$$

Donde T denota la transpuesta del vector, y el vector de pesos del filtro es:

$$h = \begin{bmatrix} h(0) \\ h(1) \\ \vdots \\ h(N-1) \end{bmatrix}$$

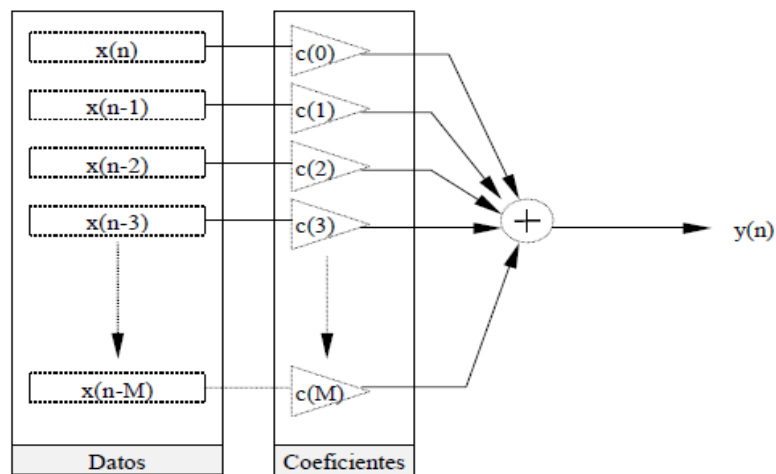
Y la señal del vector de entrada

$$x(n) = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(n-p) \end{bmatrix}$$

Una notación comúnmente utilizada en procesamiento de señales es la convolución. El filtrado FIR en forma de convolución lineal discreta se expresa:

$$y(n) = h(n) * x(n)$$

Donde \* significa convolución.



**Figura 2.6. Estructura de un filtro digital tipo FIR.**

Otra forma para hacer el filtrado FIR es con la transformada de las señales desde el dominio del tiempo discreto al dominio de la frecuencia compleja  $z$ . La transformada  $z$  bilateral de  $X(z)$  de una señal discreta  $x(n)$  se define como:

$$X(z) = Z\{x(n)\} = \sum_{n=-\infty}^{\infty} x(n)z^{-n}$$

Y la transformada  $z$  inversa es:

$$x(n) = Z^{-1}\{X(z)\} = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{jw})e^{jnw} dw$$

Donde  $z = e^{jw}$  es una variable compleja,  $j$  es la unidad imaginaria y  $w$  es el ángulo de  $z$ .

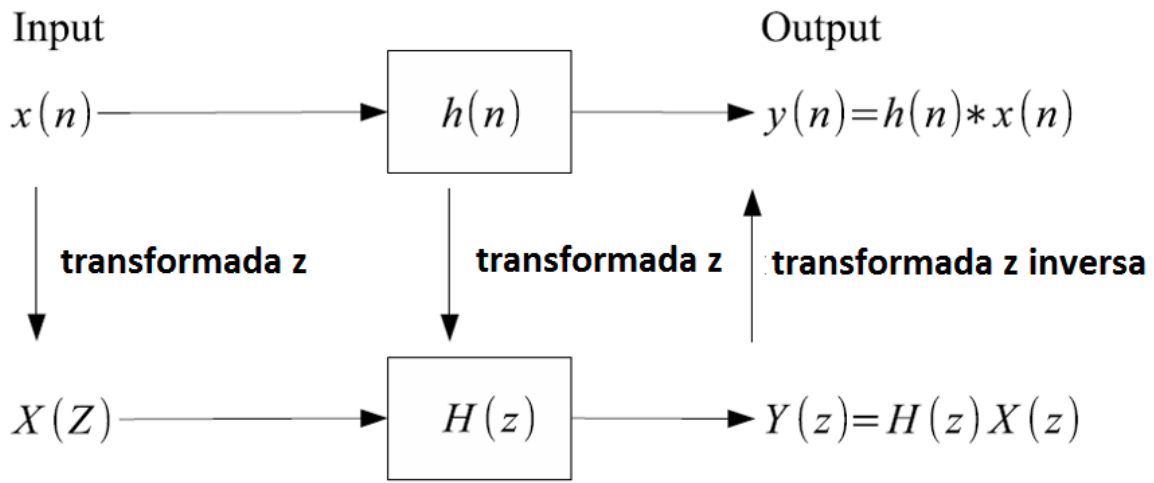
El teorema de la convolución en el dominio  $z$  será una multiplicación en el dominio del tiempo discreto.

Esta propiedad hace posible calcular el filtrado FIR en el dominio  $Z$ :

$$Y(z) = Z\{h(n * x(n))\} = H(z)X(z)$$

Y puede ser transformada para el tiempo discreto por medio de la transformada  $z$  inversa.

En la figura 2.7 se ilustra un diagrama de bloques del filtro FIR en ambos dominios (tiempo discreto y transformada  $z$ ) y también la relación entre estos dominios. Los filtros FIR son siempre de entradas acotadas y salidas acotadas (BIBO por sus siglas en inglés). SI un sistema es BIBO y estable, entonces la salida del mismo está acotado por cada entrada que está acotada. Esta propiedad hace los filtros FIR adecuados para los sistemas de control activo de ruido.



**Figura 2.7. Diagrama de bloques de los filtros FIR en el dominio del tiempo y el dominio de z.**

### 3. METODOLOGÍA

Teniendo en cuenta el alcance de los objetivos y en respuesta al carácter investigativo del proyecto, se estableció un orden para la implementación de los bloques que lleven a la solución del problema, valiéndose de simulaciones y experimentos para detectar y resolver problemas técnicos y prácticos. En primer lugar, se hace un breve análisis de los diferentes algoritmos con los que se realizaron pruebas del sistema de control, se muestra la elección de los transductores y la implementación del acondicionamiento de las señales. También se muestra la realización de una estructura para la fuente de ruido y la obtención del perfil armónico de ésta. Se muestra el desarrollo del sistema de control de ruido acústico, la implementación y análisis de los diferentes algoritmos, de los cuales se obtienen, finalmente, los resultados de la monografía.

### 3.1 ALGORITMOS DE CONTROL

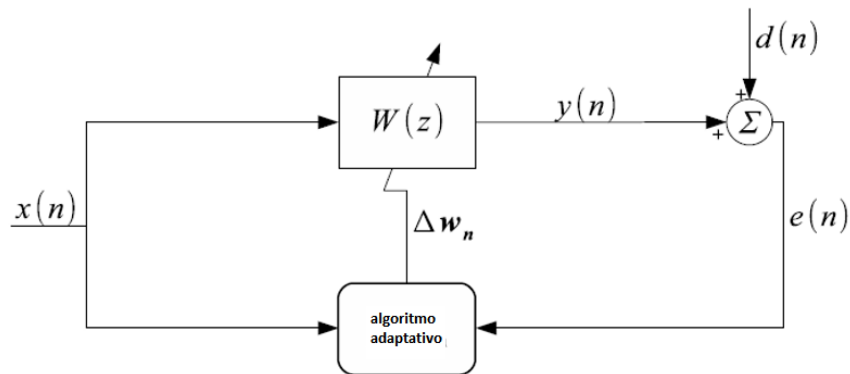
El controlador ha de ser capaz de seguir todas las posibles variaciones que puedan producirse en el campo acústico (variaciones de fase, de frecuencia, de amplitud, de funciones de transferencia acústica, etc.). El controlador digital consiste en una etapa de entrada donde se tiene un filtro paso bajo “anti-aliasing” y un convertidor analógico-digital. En la etapa de salida se encuentra un convertidor digital analógico y un filtro paso bajo de salida anti-imagen. El sistema digital lo constituye un filtro digital (algoritmo numérico), cuyos parámetros son actualizados dinámicamente por medio de un algoritmo adaptativo, de manera que siga cualquier variación de las condiciones acústicas del sistema. Las señales analógicas de los sensores, micrófonos de error y de referencia, se muestrean a una frecuencia bastante superior a la frecuencia de Nyquist. (Allie 1998, Mínguez 1998).

#### 3.1.1 FILTRO ADAPTATIVO DE LA MÁXIMA PENDIENTE

El objetivo del algoritmo es encontrar de manera iterativa el filtro óptimo que minimice la función de mínimo error cuadrático. (MSE por sus siglas en inglés).

$$\mathcal{E}(n) = E\{e(n)^2\}$$

Donde  $e(n)$  es la señal de error en tiempo  $n$ .



**Figura 3.11 Diagrama de bloques de un filtro adaptativo.**



El filtro adaptativo de la figura ajusta los pesos hasta que  $y(n)$  esté en una estimación satisfactoria de la señal  $d(n)$ . La actualización del filtro está basado en la señal de entrada  $x(n)$  y la señal de error.

$$e(n) = d(n) - y(n) = d(n) - w_n^T x(n)$$

Donde el vector de pesos del filtro es

$$w(n) = \begin{bmatrix} w_n(0) \\ w_n(1) \\ \cdot \\ \cdot \\ w_n(p) \end{bmatrix}$$

El fundamento del algoritmo adaptativo es la ajustar dinámicamente el filtro en cada iteración. En cada paso los pesos del filtro son actualizados al agregar un término de corrección que acerca los pesos a una solución óptima. Los términos de corrección apuntan hacia la máxima pendiente de  $\mathcal{E}(n)$ . Matemáticamente la dirección es denotada por el gradiente. El gradiente es el vector de derivadas parciales del MSE con respecto al peso el vector de filtros  $Wn$ . Ya que los puntos del gradiente  $\nabla\mathcal{E}(n)$  apuntan en la dirección de la pendiente ascendente, un cambio en el signo  $-\nabla\mathcal{E}(n)$  da la dirección de gradiente de descenso. Para controlar como la corrección afecta los coeficientes antiguos la variable  $u$  es introducida. Este paso es multiplicado con el gradiente negativo dando la expresión de la ecuación:

$$w_{n+1} = w_n - \mu \nabla\mathcal{E}(n)$$

Donde el gradiente del vector de MSE es:

$$\nabla\mathcal{E}(n) = \nabla E\{e(n)^2\}$$

$$\nabla\mathcal{E}(n) = E\{\nabla e(n)^2\}$$

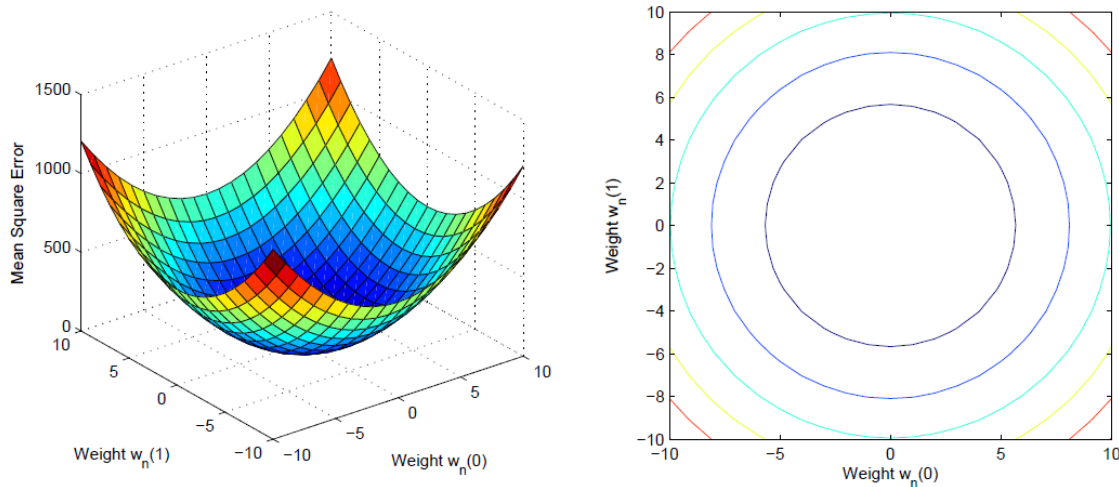
$$\nabla\mathcal{E}(n) = E\{2e(n)\nabla e(n)\}$$

Utilizando la ecuación de error, se tiene que:

$$\nabla \mathcal{E}(n) = E\{2e(n)x(n)\}$$

La cual conduce a la siguiente expresión del algoritmo de máxima pendiente

$$w_{n+1} = w_n - \mu E\{2e(n)x(n)\}$$



**Figura 3.12.** Un ejemplo del MSE  $\mathcal{E}(n) = E\{e(n)^2\}$  de los dos pesos del filtro se ilustra por el gráfico de superficie a la izquierda y el gráfico de contorno a la derecha. La dirección del descenso más rápido apunta hacia el mínimo de la superficie de error (opuesta a la del gradiente).

### 3.1.2 LMS MEDIOS MÍNIMOS CUADRÁTICOS

El algoritmo LMS puede ser visto como un caso especial del filtro de máxima pendiente. Por lo general el término  $\mu E\{2e(n)x(n)\}$  es desconocido y debe ser reemplazado por una estimación del promedio de la muestra:

$$E\{2e(n)x(n)\} = \frac{1}{L} \sum_{l=0}^{L-1} e(n-l)x(n-l)$$

Por tanto el valor para  $w_{n+1}$

$$w_{n+1} = w_n + \frac{2\mu}{L} \sum_{l=0}^{L-1} e(n-l)x(n-l)$$

Se puede simplificar si usamos un promedio de la muestra de un punto ( $L = 1$ ).

$$w_{n+1} = w_n + \mu 2e(n)x(n)$$

Para estos cambios la actualización de la ecuación del vector de pesos está escrita en una forma particularmente simple conocida como el algoritmo LMS.

### 3.1.3 LMS NORMALIZADO

El algoritmo LMS normalizado (NLMS) se introduce para mejorar el tamaño de paso de adaptación  $\mu$  utilizado en el algoritmo LMS. Un paso muy pequeño resulta en una convergencia lenta del sistema, pero el resultado es un error muy pequeño. Un paso muy grande hace al sistema converger más rápidamente, pero el error es generalmente más grande y podrá ocurrir un comportamiento estocástico que pueda hacer divergir la adaptación. Para una señal de entrada variante muy grande  $x(n)$  el algoritmo LMS se torna inestable. Este comportamiento es conocido como el gradiente de amplificación de ruido. Una solución a este problema es usar un paso de variable que dependa de la energía. Por lo tanto en el algoritmo NLMS el tamaño del paso es ajustado al normalizarlo con la energía de la señal de entrada  $x(n)$ .

La energía de la señal de entrada  $x(n)$  puede ser expresada como:

$$\|x(n)\|_{l_2}^2 = \sum_{k=(n-p)}^n |x(k)|^2$$

La energía inversa multiplicada por un coeficiente  $\beta$  se convierte en el tamaño del paso para el algoritmo NLMS.

$$\mu(n) = \frac{\beta}{\|x(n)\|_{l_2}^2}$$

Con el nuevo  $\mu$  colocado en la ecuación de actualización de pesos, resulta la expresión para el algoritmo LMS normalizado (NLMS).

$$w_{n+1} = w_n + \frac{2\beta}{\|x(n)\|_{l_2}^2} e(n)x(n)$$

Aunque el algoritmo NLMS resuelve el problema del gradiente de amplificación de ruido, un problema similar ocurre cuando  $\|x(n)\|_{l_2}^2$  resulta ser muy pequeño. Por lo tanto el algoritmo NLMS es modificado al añadir un número pequeño positivo en el denominador.

$$w_{n+1} = w_n + \frac{2\beta}{\epsilon + \|x(n)\|_{l_2}^2} e(n)x(n)$$

El tamaño de pesos variable mejora la estabilidad y la tasa de convergencia a cambio de un costo mínimo. La única computación extra es evaluar el término de normalización que involucra dos operaciones cuadráticas, una adición y una resta. Ya que el término puede ser calculado recursivamente.

$$\|x(n+1)\|_{l_2}^2 = \|x(n)\|_{l_2}^2 + |x(n+1)|^2 - |x(n-p)|^2$$

El algoritmo converge si la siguiente condición se cumple:

$$0 < \mu < 2$$

Es importante mencionar que la condición de convergencia es considerada con precisión infinita, Ya que los computadores tienen una precisión finita debemos tener esto en mente cuando seleccionamos los parámetros.

### 3.1.4 LEAKY LMS

Introduciendo un factor *leaky* en el algoritmo LMS, se restringirá la energía en los coeficientes del filtro y se estabilizará el algoritmo. El factor *leaky* está dado por  $\nu = 1 - \mu\gamma$ , donde  $\gamma$  es una constante positiva escogida para que  $0 < \nu \leq 1$  se dé. Por lo que la actualización del filtro adaptativo con Leaky LMS es:

$$w_{n+1} = (1 - \mu\gamma)w_n + 2\mu e(n)x(n)$$

El algoritmo Leaky LMS limita el crecimiento del filtro y reduce al mismo tiempo el error numérico en una implementación de punto fijo. Su desventaja es el pequeño error de adaptación sistemático que puede producir una perturbación del filtro.

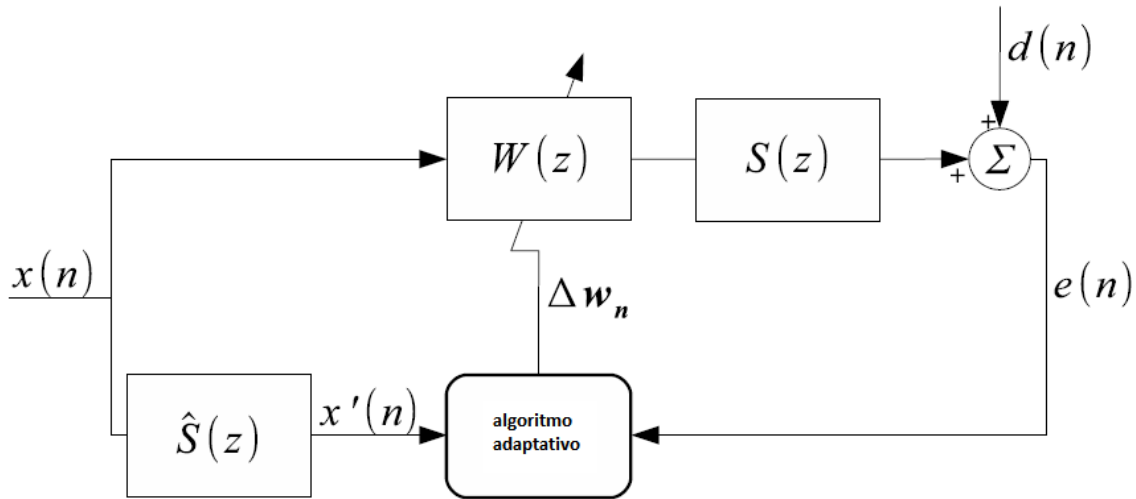
### 3.1.5 ALGORITMO FILTERED-X LMS

Este algoritmo es una alternativa del algoritmo LMS para cuando el filtro adaptativo se calcula con un camino secundario. Este camino secundario se representa por  $S(z)$ . En una situación ideal la señal de error es calculada por la diferencia entre la salida del camino primario y la salida del filtro adaptativo, como el LMS normal.

En la práctica, los sistemas de control activo de ruido involucran un camino secundario el cual incluye los conversores D/A, el amplificador, el parlante, el campo acústico entre el parlante y el micrófono de error, etc. Esto puede causar problemas y el retraso generado necesita compensarse. La solución más obvia es colocar un camino secundario inverso en serie con el normal para reducir su impacto. En general, el camino secundario no es un sistema de fase mínima con una inversa existente. Una mejor solución es compensar por el retraso de la señal de entrada  $x(n)$  al algoritmo adaptativo de control por un estimado del camino secundario.

$$x'(n) = s'(n) * x(n)$$

Donde  $x'(n)$  es la señal filtrada en tiempo  $n$ .



**Figura 3.13. Diagrama de bloques del algoritmo Filtered-x LMS.**

Esta modificación es algo más complicada que el algoritmo LMS normal pero resulta en un mejoramiento en ejecución por una rápida convergencia, mayor estabilidad, y una mayor precisión del algoritmo adaptativo.

### 3.1.6 ARITMÉTICA DE PUNTO FIJO.

Un procesador de 16 bit, como en nuestro caso la tarjeta TMS320C6416 de Texas Instruments, almacena números en un formato entero de este tamaño. Esto significa que puede representar 65535 valores, típicamente enteros desde -32768 a 32767.

El rango dinámico (diferencia entre los valores mínimo y máximo) para una cuantización de 16 bits es

$$20 \log_{10}(2^{15}) = 90 \text{dB}$$

Y la mejor relación entre señal a ruido SNR para una señal sinusoidal es:

$$\frac{10 \log_{10} \left( 2^{\frac{15.2}{2}} \right)}{1/12} = 98 \text{dB}$$

El ruido de cuantización es pequeño para señales de amplitud alta, por lo tanto la potencia de la señal debe ser tan grande como sea posible sin desbordamiento, el desbordamiento ocurre cuando una operación aritmética excede la capacidad del registro utilizado para el resultado. La adición de dos números de 16 bits producirá una suma de 16 bits, por lo tanto no hay error de redondeo introducido. El problema ocurre cuando una suma se salga del rango preestablecido. La manera más efectiva para prevenir esto es disminuir la magnitud escalando con un valor menor a 1 antes del desbordamiento y después escalándolo a su valor original cuando el riesgo de desbordamiento desaparezca. Esto se necesita hacer cuidadosamente ya que al hacerlo pueden producir errores de redondeo.

Por ejemplo si una señal es reducida a la mitad el error de redondeo se incrementa 6dB y es equivalente a perder un bit de la señal que está siendo representada.

Por estas limitaciones en rango dinámico se tiene que aplicar factores que previnieran el desbordamiento aritmético lo cual es complejo. Pero en general los dispositivos de procesamiento de punto fijo son más económicos y rápidos que los de punto flotante ya que usa menos silicio y tiene menos pines. Así que cuando se escoge entre estos se debe tomar lo anterior en consideración.

### **3.1.7 LIMITACIONES EN TIEMPO REAL**

La velocidad del procesador y la complejidad del algoritmo determinan el tiempo usado para muestrear las señales analógicas. Un DSP en tiempo real tiene que tener en cuenta estas limitantes, dado que el tiempo de procesamiento debe ser menor al tiempo de muestreo.

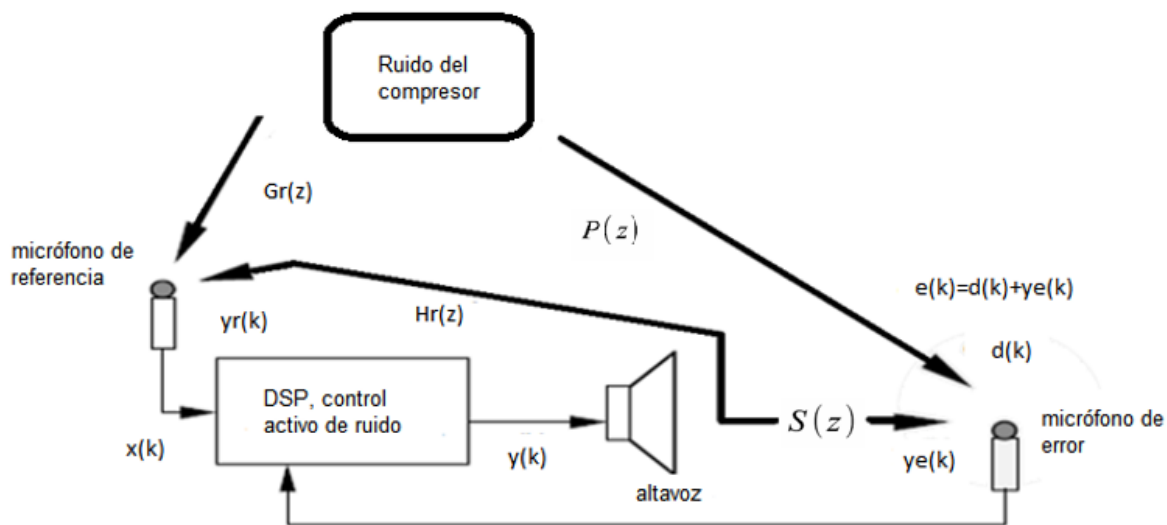
Esto restringe la máxima tasa de muestreo en el dispositivo, la mínima tasa permitida se determina por el teorema de muestreo de Nyquist.

Aunque los sistemas DSP se están volviendo cada vez más veloces esto aún es una limitación. Y es más evidente aun cuando se involucra el costo del mismo. Así que se debe tener un equilibrio entre estos factores.

Los métodos más comunes usados en sistemas de control activo de ruido son feedforward y feedback. El método que se implementó fue el feedforward. El procedimiento tomado es primero identificar el camino secundario y usarlo su estimación en el algoritmo.

### 3.1.8 CONTROL FEEDFORWARD.

Cuando se dispone de una señal que da una referencia del ruido que se desea cancelar, la configuración del sistema de control es de lazo abierto.



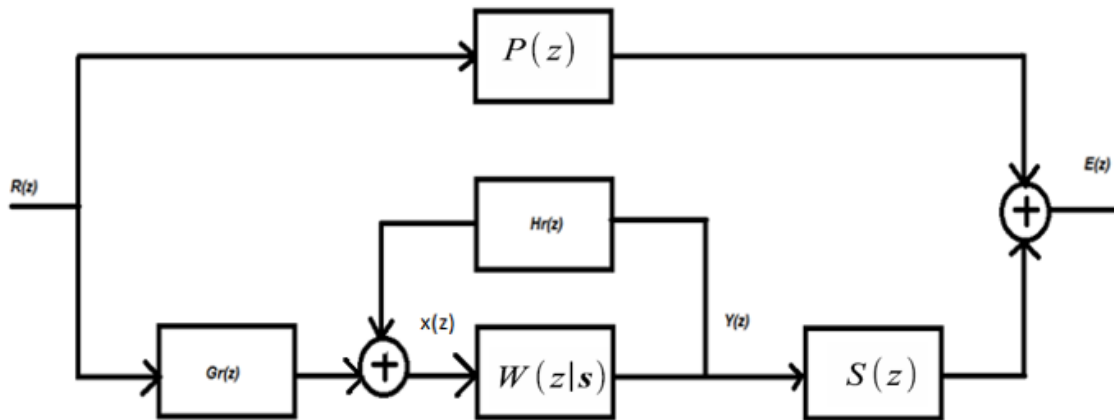
**Figura 3.14. Control Feedforward, lazo abierto.**

A partir de una referencia del ruido  $x(k)$ , el controlador genera una señal de salida  $y(k)$ , de manera que en el micrófono de error se encuentre invertida en fase  $y_e(k)$ , con respecto a la señal que llega al micrófono de error,  $d(k)$ , procedente de la fuente de ruido. De esta forma, se consigue una disminución de la presión acústica en el entorno de este micrófono. Las variaciones acústicas del medio perturban la función de transferencia del sistema de control. Cuando las condiciones acústicas cambian, hay que modificar la respuesta del controlador. Es imprescindible dar información al controlador de la señal de error  $e(k)$ , para que este modifique su respuesta, cuando cambien las condiciones de la planta. De esta forma, el controlador intentará en todo momento hacer nula la señal que se recoge por el micrófono de



error, esto es, el nivel de presión acústica en el entorno del micrófono. (Mínguez Olivares, 1998).

El sistema de control puede representarse mediante un diagrama de bloques con todas las funciones de transferencia electro/acústicas involucradas:



**Figura 3.15. Diagrama de bloques del control Feedforward.**

Las señales y funciones de transferencia son las siguientes:

$R(z)$  = Ruido.

$E(z)$  = Señal de error.

$Y(z)$  = Señal de salida del controlador.

$P(z)$  = Función de transferencia electro-acústica entre la fuente de ruido y el micrófono de error.

$Gr(z)$  = Función de transferencia electro-acústica entre la fuente de ruido y el micrófono de referencia.

$S(z)$  = Función de transferencia electro-acústica entre el altavoz y el micrófono de error.

$Hr(z)$  = Función de transferencia electro-acústica entre el altavoz y el micrófono de referencia.

$W(z)$  = Función de transferencia del sistema de control.

El controlador estima  $W(z)$  para hacer que la señal de error sea cero, de acuerdo al análisis de bloques se tiene:

$$E(z) = R(z) \left[ P(z) + Gr(z) \frac{W(z)}{1 - W(z)Hr(z)} S(z) \right] = 0$$

Resolviendo:

$$\left[ P(z) + Gr(z) \frac{W(z)}{1 - W(z)Hr(z)} S(z) \right] = 0$$

$$Gr(z) \frac{W(z)}{1 - W(z)Hr(z)} S(z) = -P(z)$$

$$\frac{Gr(z)W(z)S(z)}{1 - W(z)Hr(z)} = -P(z)$$

$$Gr(z)W(z)S(z) = [-P(z)][1 - W(z)Hr(z)]$$

Despejando  $W(z)$ :

$$W(z) = -\frac{P(z)}{Gr(z)S(z) - P(z)Hr(z)}$$

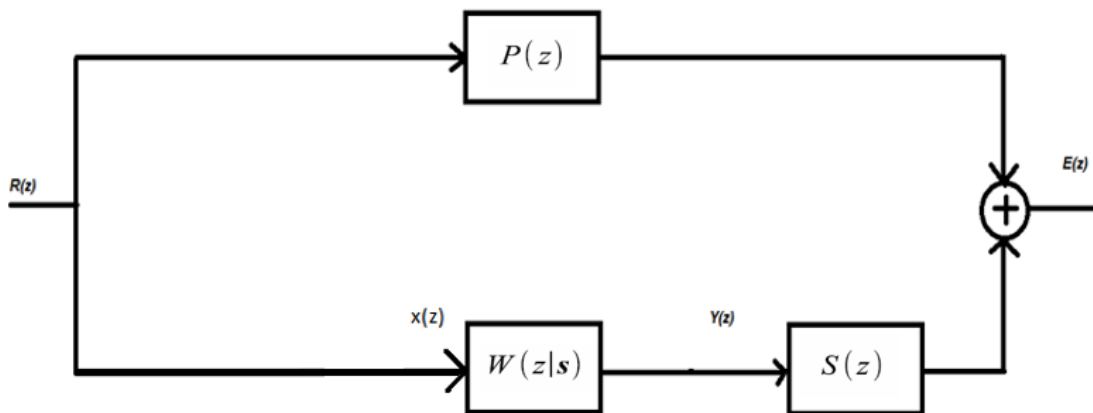
La atenuación acústica que se consigue es inferior a aquella que se logra cuando no existe realimentación acústica. En este caso, la función de transferencia  $Hr(z)$  (función de transferencia electro-acústica entre el altavoz y el micrófono de referencia) se iguala a cero:

$$C(z) = -\frac{P(z)}{Gr(z)S(z) - P(z)(0)}$$

Por tanto, la función de transferencia del controlador tiene la siguiente expresión:

$$C(z) = -\frac{P(z)}{Gr(z)S(z)}$$

Dado que el micrófono de referencia se encuentra tan cerca de la fuente de ruido no se tiene en cuenta la función de transferencia  $Gr(z)$  se aproxima a un valor de uno.



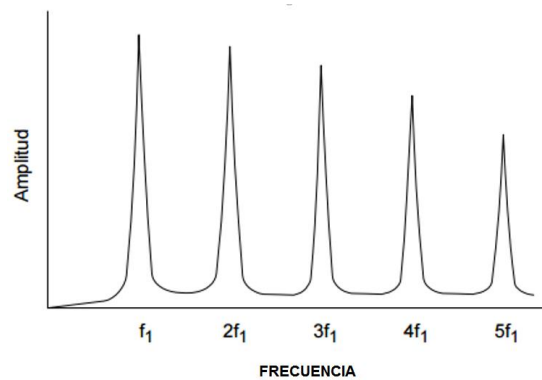
**Figura 3.16. Diagrama de bloques sin realimentación acústica y con la función de transferencia entre la señal de ruido  $R(z)$  y el micrófono de referencia igual a uno.**

Por tanto la función de transferencia para  $W(z)$  es:

$$W(z) = -\frac{P(z)}{S(z)}$$

### 3.1.9. ALGORITMOS PARA SISTEMAS DE BANDA CORTA (Filtro Notch)

Cuando el ruido primario es producido por máquinas rotativas y la señal es periódica, hay un método alternativo, el cual, estima la señal acústica mediante una medición indirecta de un sensor no acústico en lugar del micrófono de referencia.



**Figura 3.17. Frecuencia fundamental y múltiplos de la frecuencia fundamental que contienen la mayor parte de la energía de la señal.**

Esta situación se asemeja a las condiciones dadas por el compresor, ya que de la caracterización de la señal se sabe exactamente la frecuencia que aporta más energía al ruido producido.

### 3.1.10. ALGORITMO GENÉTICO

Los algoritmos genéticos ofrecen un diseño mucho más simple para las configuraciones de sistemas de control activo multicanal. Con estos algoritmos no es necesario estimar las funciones de transferencia de cancelación ya que la adaptación del sistema de control se realiza mediante una búsqueda aleatoria de los coeficientes de cada uno de los filtros que constituyen el sistema de control multicanal. La búsqueda de coeficientes se realiza tanto en estructuras tipo FIR como tipo IIR, sin peligro de quedarse en estas últimas atrapado en un mínimo local.

### 3.2 ELECCIÓN DEL TRANSDUCTOR.

Tanto las señales de entrada como las de salida representan el vínculo de unión entre el controlador y campo acústico exterior. La conversión del campo acústico a señales eléctricas y viceversa se produce a través de dispositivos electro-acústicos. La calidad de estos dispositivos se valora en términos de linealidad (baja distorsión), en términos de relación señal a ruido, y en términos de retardo de conversión. Estos tres parámetros de calidad se valoran en el margen de frecuencias de interés, que en el caso de control activo de ruido no superan los 500 Hz.

La linealidad es imprescindible para obtener una señal eléctrica que represente fielmente las ondas de presión acústica que se recogen, por ejemplo, a través de micrófonos o que se generan a través de altavoces. Cuando un dispositivo no es lineal genera armónicos no deseados de la señal y estos medidos porcentualmente con respecto al nivel de la frecuencia fundamental expresan el grado de distorsión. Un nivel elevado de los armónicos en los sensores electro-acústicos (sensor de referencia o sensor de error) enmascara la señal eléctrica que debe procesar el controlador, impidiendo en muchos casos alcanzar el funcionamiento óptimo del mismo. En el caso de los altavoces, dispositivos electro-acústicos encargados de producir las ondas acústicas de presión, la distorsión de los mismos crea en el escenario acústico de control nuevas ondas de presión acústica (armónicos de la señal ideal) que no van a cancelar ningún ruido, ya que de este ruido el controlador no tiene referencia y no está sujeto a control, y lógicamente se va a producir un incremento del ruido acústico. Los valores de distorsión mínimamente aceptables para los dispositivos electro-acústicos tienen que ser inferiores al 1%. (Mínguez Olivares, 1998).

La relación señal/ruido (S/N) de los transductores electro-acústicos también determina el buen funcionamiento del sistema de control. Al igual que la distorsión, el ruido de fondo propio del transductor enmascara la señal ideal de entrada al controlador o desvirtúa la conversión electro-acústica en la salida (altavoz). El ruido suele ser de naturaleza aleatoria y de características similares al ruido blanco. La suma de este ruido en la señal de referencia actúa como un factor "leakage" o fuga, en el algoritmo adaptativo de control. Si este ruido es de bajo nivel, al menos 40 dB por debajo de la señal, puede ser incluso deseable ya que el

sistema adaptativo de control se comporta de manera más estable. A medida que la relación S/N disminuye, el algoritmo adaptativo es incapaz de encontrar los coeficientes óptimos que garanticen una disminución del ruido acústico exterior, ya que los valores de estos oscilan alrededor de los valores óptimos y esta oscilación es tanto mayor cuanto menor sea la relación S/N. De la misma forma que ocurría con la distorsión, el ruido del altavoz provoca la radiación de un ruido acústico no sujeto a control, incapaz de ser atenuado por el controlador, aumentando el ruido acústico residual en el exterior. (Mínguez Olivares, 1998).

La influencia del retardo de los dispositivos electro-acústicos determina la causalidad del sistema de control. El controlador cambia la amplitud y la fase de la señal eléctrica de entrada para conseguir en el micrófono de error una réplica perfecta invertida de la señal del ruido acústico primario, y así mediante superposición lograr la cancelación acústica. El cambio de fase que realiza el controlador se lleva a cabo mediante retardos unitarios múltiplos del periodo de muestreo de la digitalización. De tal forma, si desea realizar esta cancelación la señal del ruido acústico tiene que llegar con suficiente anticipación en el micrófono de referencia respecto al micrófono de error. El camino que recorre la señal que llega por el micrófono de referencia hasta que llega al micrófono de error lleva aparejado un retardo que hay que considerar. Los retardos a tener en cuenta son:

- Sensor de referencia.
- Filtro anti-aliasing.
- Convertidor analógico-digital.
- Controlador.
- Convertidor digital-analógico.
- Filtro anti-imagen restructor de la señal analógica.
- Altavoz.
- Camino acústico altavoz-micrófono de error.

Las ondas de presión del ruido acústico tienen que llegar al micrófono de referencia antes que al micrófono de error con un retardo igual o inferior a la suma de todos los retardos involucrados. Obviamente, el controlador es el único elemento que tiene un retardo variable para poder modificar la fase de la señal y conseguir el mejor control sobre la misma. En consecuencia, los demás elementos tienen que introducir el mínimo retardo para que el controlador tenga la mayor capacidad de control. Si el retardo de estos elementos es muy

elevado, puede llegar incluso a inhabilitar la acción del controlador. El retardo de los micrófonos suele considerarse despreciable. No ocurre lo mismo con el retardo introducido por los altavoces, especialmente a bajas frecuencias, por lo que debe valorarse este parámetro en la selección del mismo. (Mínguez Olivares, 1998).

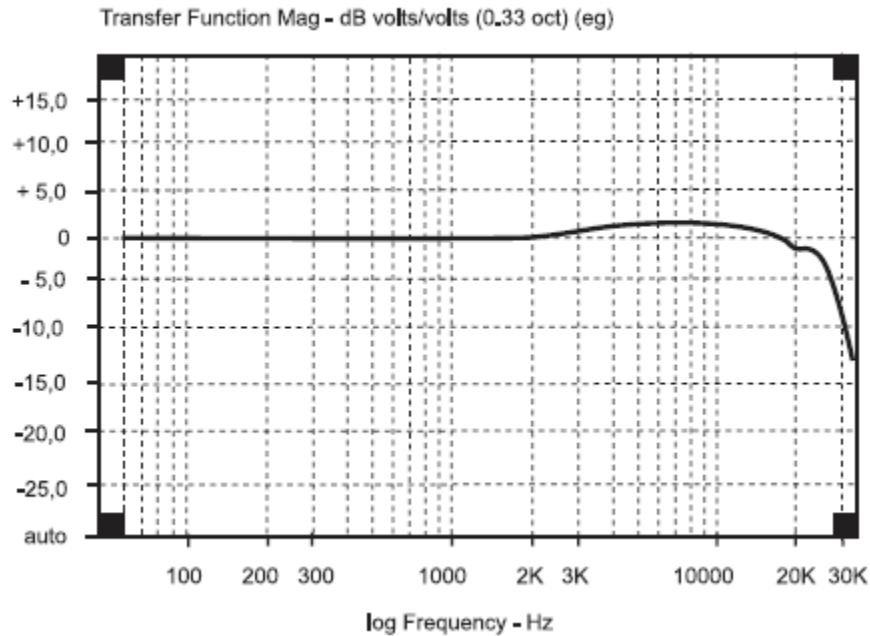
Existen muchas situaciones donde la fuente de ruido esta accesible y en estos casos es recomendable utilizar acelerómetros, situados directamente sobre la fuente de ruido, los cuales van a entregar una señal eléctrica directamente proporcional a la aceleración de las vibraciones de la fuente de ruido. (Mínguez Olivares, 1998).

Como se vio en el capítulo 2.3, los micrófonos tipo electret condensador pueden lograr una buena señal de referencia, además de tener un bajo costo. Para el proyecto, se utilizó un micrófono tipo electret condensador Behringer ECM8000, tanto para la señal de referencia como para la señal de error. En la tabla 3.1 se muestran las especificaciones técnicas de este micrófono.

Type	electr. condenser
Polar Pattern	omnidirectional
Impedance	200 Ohms
Sensitivity	70 dB
Freq. Response	20-20000 Hz
Connector	gold-plated XLR
Phantom Power	+15 to +48 V
Weight	136 g

**Tabla 3.1. Especificaciones micrófono condensador Behringer ECM8000.**

En la figura 3.21 se puede ver la respuesta en frecuencia, para el rango de interés, el cual son frecuencias menores a 500Hz, la respuesta tiende a ser plana, es decir, ninguna clase de distorsión que atenúe o amplifique las señales de referencia y error.



**Figura 3.21. Respuesta en frecuencia del micrófono Behringer ECM8000 según el fabricante.**

### 3.3 ETAPA DE ACONDICIONAMIENTO.

Los niveles de las señales de entrada que entregan los sensores de referencia y sensores de error, deben ser acomodados a los niveles de tensión que admite el convertidor A/D de la tarjeta. Los transductores electro-acústicos entregan unos niveles muy bajos, del orden de milivoltios, y es necesario aplicar una ganancia considerable para obtener al menos una señal de 200mV. Si es mayor la ganancia, se pueden saturar los valores y por tanto puede haber un mal procesado de las señales en el DSP. La ganancia a aplicar a cada uno de los micrófonos se determina ajustando la tensión de salida del pre-amplificador cuando los niveles de ruido acústico exterior son los máximos posibles a los que va estar sometido el sensor.



### 3.3.1 CONSOLA MIXER 8 CANALES BEHRINGER XENYX 802

En primer lugar, para la preamplificación de las señales de referencia y error, se utilizó un amplificador basado con BJT's. Sin embargo, debido a la interferencia por ruido electromagnético proveniente de la red eléctrica, se optó por mejorar esta etapa utilizando una consola que, además de brindar la alimentación adecuada a los micrófonos, (Phantom Power 48Vdc) brinda una amplificación más acorde, con baja distorsión para que las señales necesarias sean perfectamente procesadas por los conversores de la tarjeta DSP.



**Figura 3.31. Consola Mixer 8 Canales Behringer Xenyx 802**

Algunas de las características más importantes del amplificador, tomadas de la hoja de datos del fabricante, son:

<b>Frequency response</b>	<10 Hz - 150 kHz (-1 dB), <10 Hz - 200 kHz (-3 dB)
<b>Gain range</b>	+10 to +60 dB
<b>Max. input level</b>	+12 dBu @ +10 dB gain
<b>Impedance</b>	approx. 2.6 k $\Omega$ balanced
<b>Signal-to-noise ratio</b>	110 dB / 112 dB A-weighted (0 dBu In @ +22 dB gain)
<b>Distortion (THD+N)</b>	0.005% / 0.004% A-weighted
<b>Line input</b>	
<b>Type</b>	1/4" TRS connector electronically balanced
<b>Impedance</b>	approx. 20 k $\Omega$ balanced 10 k $\Omega$ unbalanced
<b>Gain range</b>	-10 to +40 dB
<b>Max. input level</b>	+22 dBu @ 0 dB Gain
<b>Fade-out attenuation<sup>1</sup> (Crosstalk attenuation)</b>	
<b>Main fader closed</b>	90 dB
<b>Channel muted</b>	89.5 dB
<b>Channel fader closed</b>	89 dB
<b>Frequency response</b>	
<b>Microphone input to main out</b>	
<10 Hz - 90 kHz	+0 dB / -1 dB
<10 Hz - 160 kHz	+0 dB / -3 dB

**Figura 3.32. Características de la consola Mixer 8 Canales Behringer Xenyx 802.**

### **3.4 REALIZACIÓN DE ESTRUCTURA PARA EL COMPRESOR.**

Dado que los motores no pueden encerrarse completamente debido a la ventilación que requieren por el calor generado durante el roce de los carbones con el colector del rotor, se realizó una estructura que permita su ventilación y al mismo tiempo sirva como atenuador pasivo de ruido. El material utilizado fue Triplex de 1.8cm con las siguientes medidas: 47.8cm de ancho, 41.4cm de largo, 45.5cm de alto. En la figura 3.41 se muestra la estructura realizada.



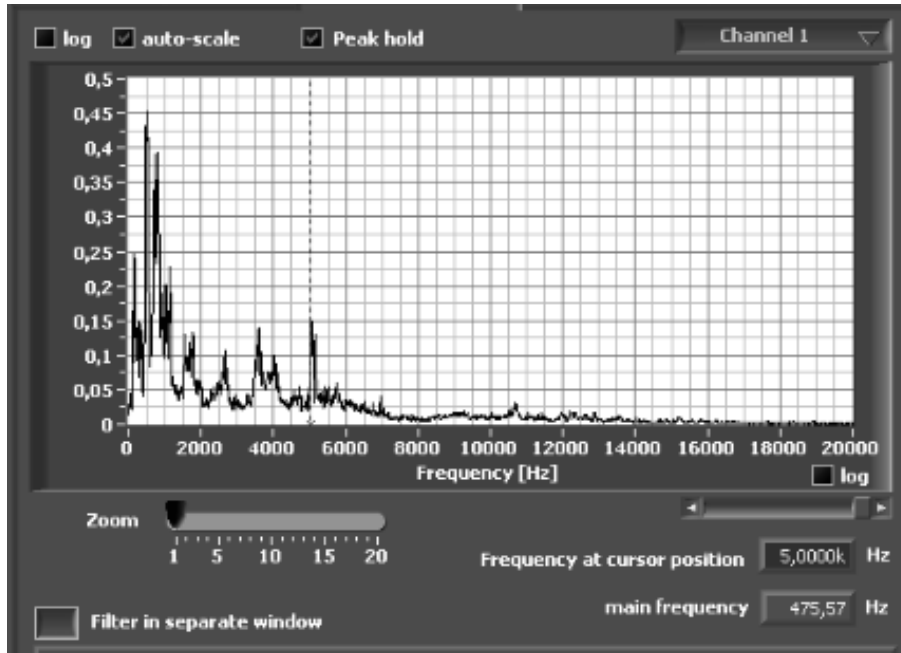
**Figura 3.41 Estructura con material triplex de 1.8cm para el compresor.**

La presión del aire debido al movimiento rotatorio y el soporte del compresor hacen que vibren las moléculas del material que lo cubre, y como consecuencia, el sonido se transmite por vibración al medio ambiente. La barrera pasiva realizada tiene la finalidad de atenuar frecuencias superiores a los 500Hz para obtener un mayor ancho de banda de atenuación. En el punto 3.5 se observa el espectro con el compresor sin la estructura de atenuación pasiva y con la estructura, debido a que con el material que se realizó, no hay tablas de referencia para obtener su nivel de atenuación de ruido.

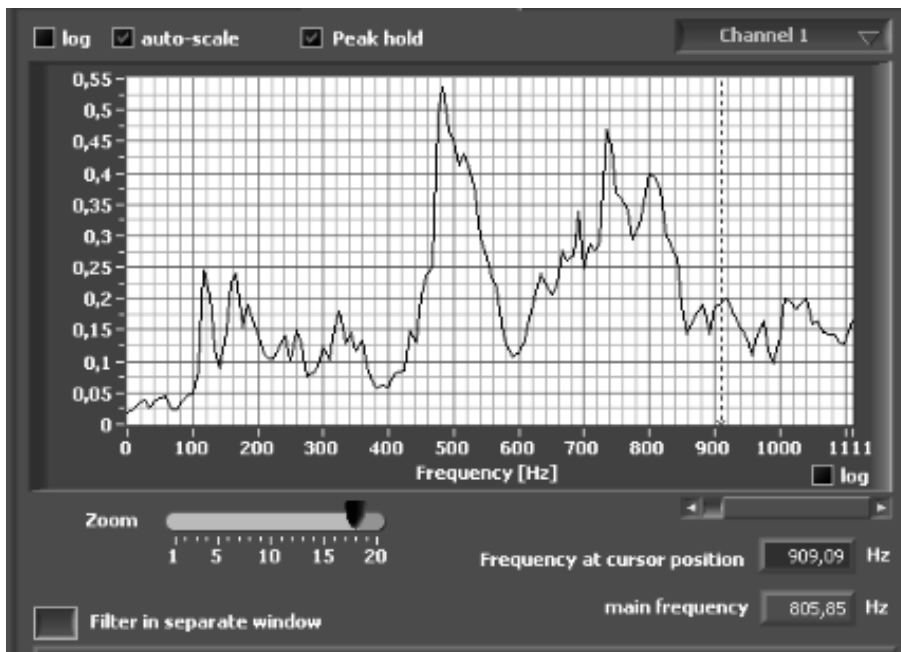
### **3.5 CARACTERIZACIÓN DE LA SEÑAL DE RUIDO ACÚSTICO DEL COMPRESOR NEUMÁTICO MARSHAL INDUCTIVE MOTOR 931373**

Ya que se desconoce el perfil armónico de las señales del compresor Marshal, es necesario elegir una frecuencia de muestreo para el procesamiento de la señal. Por lo general, se tiene la idea de que a mayor frecuencia de muestreo se obtendría mejor calidad en la onda capturada, sin embargo, esto no es completamente cierto. Si se aumenta la frecuencia de muestreo se capturan las frecuencias más altas de la onda original, pero las frecuencias bajas se siguen capturando con la misma calidad, este aumento de frecuencia trae un mayor costo computacional, convertidores analógicos/digitales más caros, mayor necesidad de memoria y mayor tiempo de procesado, por lo que no se deben capturar muestras a más frecuencia de lo estrictamente necesario. El criterio de Nyquist sostiene que el muestreo de una onda se debe realizar al menos al doble de su frecuencia máxima. Si se muestrea por debajo, la información de las altas frecuencias se pierde, si se muestrea por encima del doble, se

aumenta el costo informático sin ampliar el conjunto de frecuencias capturadas del sonido original, por lo que es necesario utilizar la frecuencia de muestreo oportuna de cada caso.



A)



B)

**Figura 3.51. Análisis en frecuencia de la señal del compresor sin la estructura. A) mayor rango de frecuencias. B) zoom en frecuencias inferiores a 500Hz.**

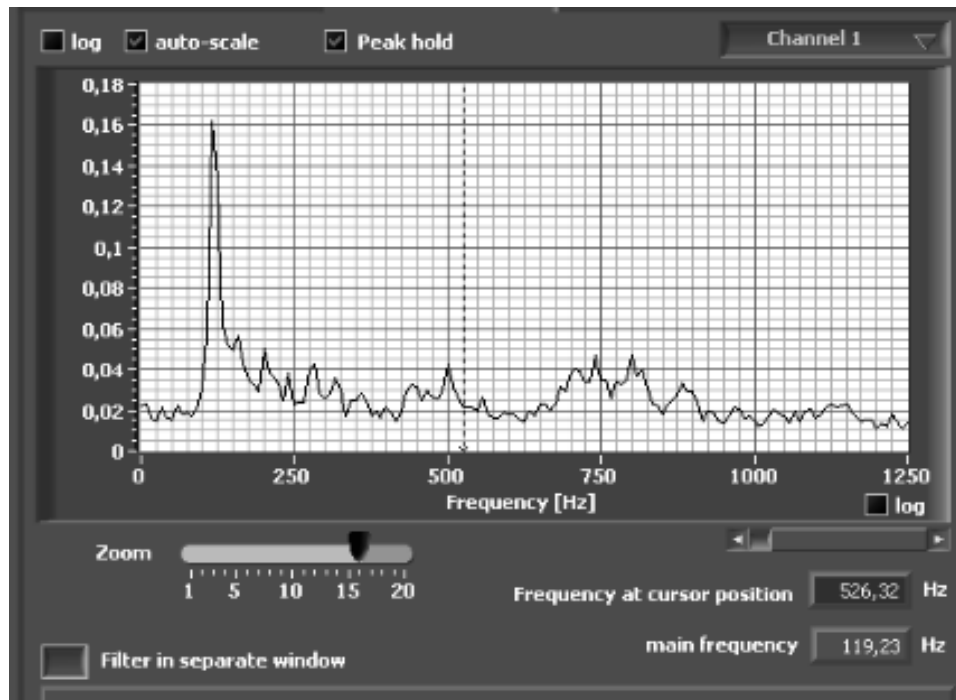
Para realizar la medición de la señal de ruido del compresor, se utilizó el software Portable Soundcard Oscilloscope v1.32 que incluye la función de análisis en frecuencia, teniendo en cuenta que para dicha medición se utilizó el micrófono condensador Behringer ECM8000 y la tarjeta de sonido del computador. En el mismo punto se utilizó el sonómetro Digital Sound level meter referencia S01 (ver anexo 3) para obtener el ruido producido por el compresor durante su operación habitual con la estructura y sin la estructura (ver anexo 4). Las mediciones se realizaron entre 1 y 10 minutos de operación del compresor (ver figura 3.53).

En la figura 3.51A se puede observar el espectro de frecuencias para la señal del compresor cuando no tiene la estructura. Se puede notar que hay una mayor concentración de energía alrededor de 480 y 720Hz. También se puede observar que el rango de sonidos significativos del compresor no es mayor a 8000Hz. Teniendo esta información de la señal de ruido del compresor Marshal, se puede obtener la óptima frecuencia de muestreo la cual está dada por la siguiente ecuación:

$$f_s = 2f$$

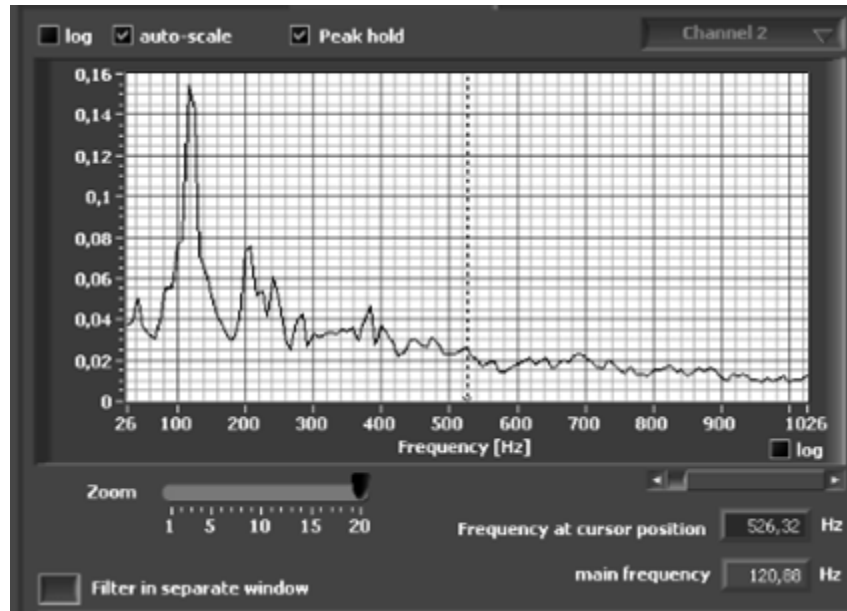
$$f_s = 2(8000Hz) = 16000Hz$$

Donde  $f$  es la máxima frecuencia contenida en el perfil armónico. La frecuencia obtenida sirve como base para realizar el muestreo y posterior desarrollo de los algoritmos en el DSP.

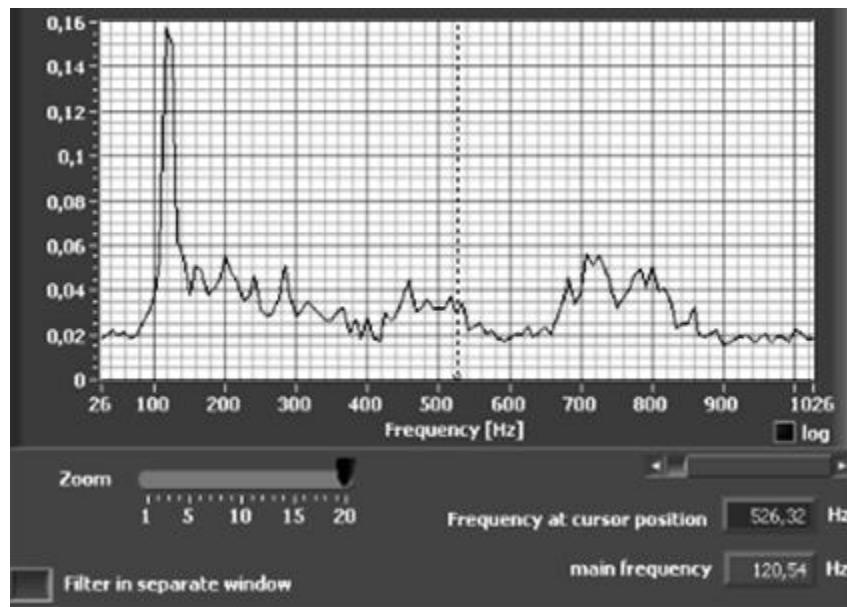


**Figura 3.52. Análisis en frecuencia de la señal del compresor con la estructura.**

En esta figura 3.52 se observa la atenuación pasiva que ofrece la estructura realizada (ver anexo 10), la cual, atenúa efectivamente las frecuencias altas, dejando un residuo de una frecuencia aproximada de 120Hz. Esta frecuencia, que se toma como la más importante, es emitida por el motor monofásico de 60Hz produciendo dos picos de torque en cada revolución. Por lo tanto, el ruido producido por el mismo en corriente alterna de interiores ocurre en esta frecuencia y posibles armónicos de la misma.



A)

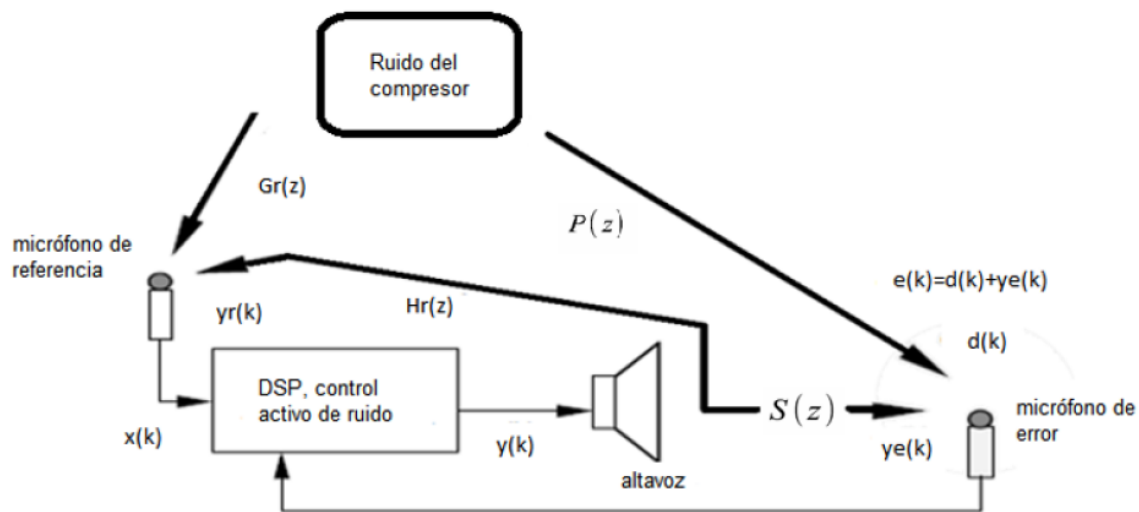


B)

**Figura 3.53. Perfil armónico del compresor con la estructura. A) después de 5 minutos. B) Después de 9 minutos.**

### 3.6 DESARROLLO DEL SISTEMA DE CONTROL DE RUIDO ACÚSTICO DE BAJA FRECUENCIA.

Las pruebas desarrolladas fueron probadas en un escenario de atenuación, en el cual, la fuente primaria (ruido del compresor) y la fuente de control van a estar muy próximos entre sí, variando la distancia para medir el resultado más óptimo. Para el compresor se implementó una estructura que permite encerrar la fuente de ruido, atenuando frecuencias por encima de 500Hz y disminuyendo el efecto que produce la fuente de control en el micrófono de referencia (realimentación acústica). El sensor de error debe situarse en un punto del recinto donde estén dispuestos tanto la fuente de control como la de ruido. Se ha escogido esta configuración debido a su simplicidad y teniendo en cuenta el interés en el funcionamiento del propio controlador electrónico.

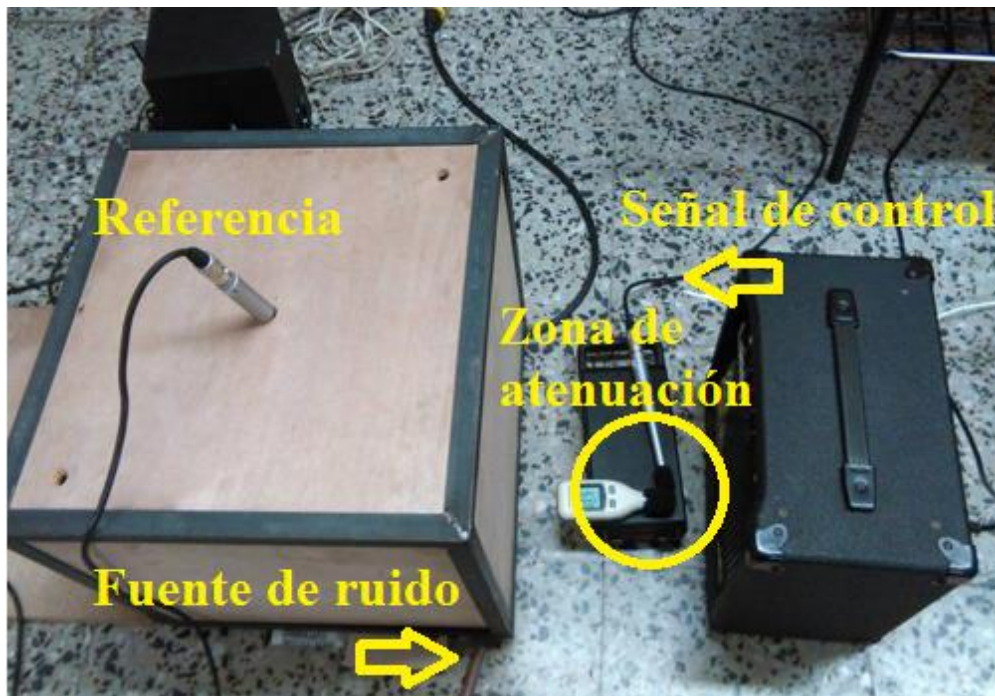


**Figura 3.61. Configuración del sistema de control en el escenario de atenuación.**

En la figura 3.61 en forma generalizada se bosqueja la cancelación de ruido acústico. La fuente de ruido primaria (ruido del compresor) se obtiene por el micrófono de referencia, esta señal se acondiciona a través del amplificador diseñado, se realiza la conversión A/D para pasar a través de un filtro FIR donde los coeficientes se actualizan mediante el algoritmo. Posteriormente la señal que sale del filtro,  $y(n)$ , se pasa a un convertidor D/A, pasa por un filtro RC (que es el filtro reconstructivo) se amplifica y se transmite mediante un altavoz. El sonido residual se captura mediante un micrófono de error colocado a cierta distancia el cual



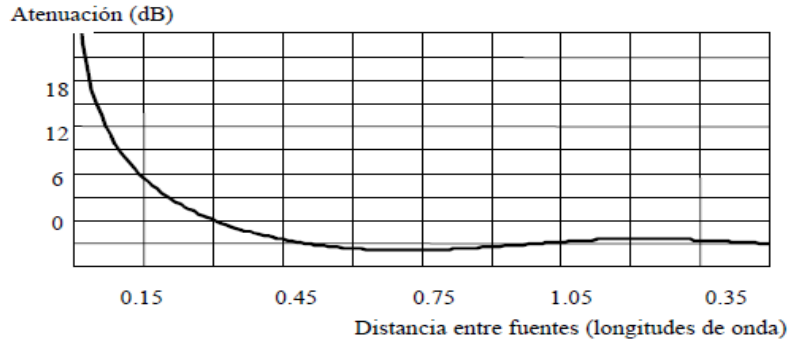
lo ingresa al DSP mediante una amplificación y una conversión A/D para ajustar los coeficientes.



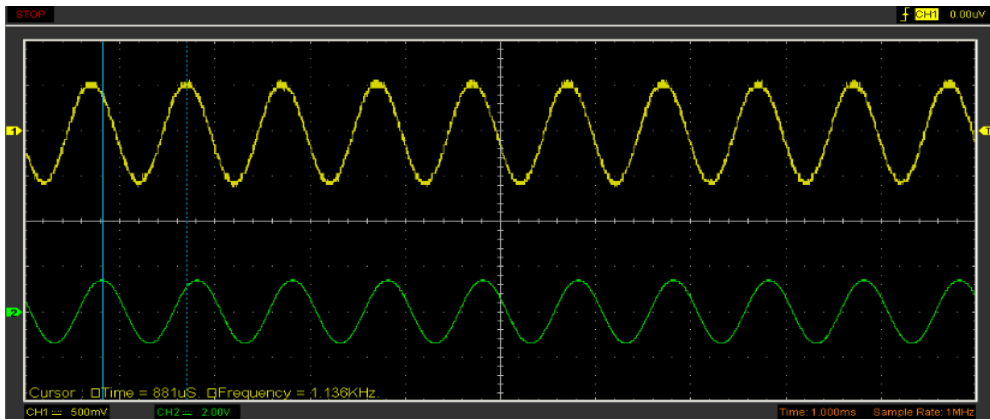
**Figura 3.62. Configuración del sistema de control en el escenario real de atenuación.**

### 3.6.1 ESCENARIO DE ATENUACIÓN

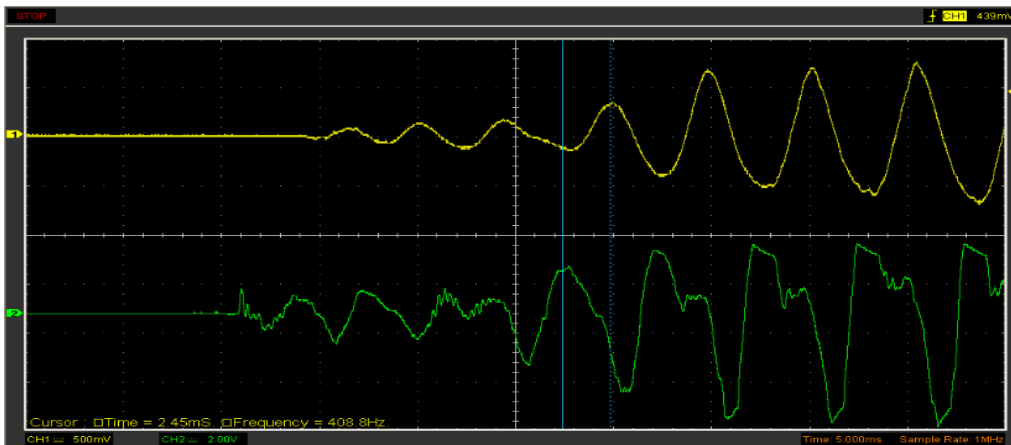
En estas condiciones, el controlador siempre atenuará la presión acústica en el punto donde se encuentre situado el sensor de error (figura 3.62). Si la fuente de ruido está muy separada de la fuente de control la única atenuación acústica que se produce es por superposición de ondas y la dimensión de la zona de cancelación es aproximadamente una esfera virtual de diámetro la décima parte de la longitud de la onda acústica. Por otra parte, cuando las dos fuentes están muy próximas entre si se produce un acoplamiento acústico entre ellas, modificándose la impedancia de radiación de la fuente de ruido debido a la fuente de control y disminuyéndose la potencia acústica radiada por la propia fuente de ruido. Esta disminución depende de la separación entre fuentes como se muestra en la figura 3.63. Por ejemplo, para una frecuencia de 100 Hz la longitud de la onda acústica es de 3.4 metros y por consiguiente, una separación entre altavoces inferior a 34 cm producirá valores importantes de atenuación.



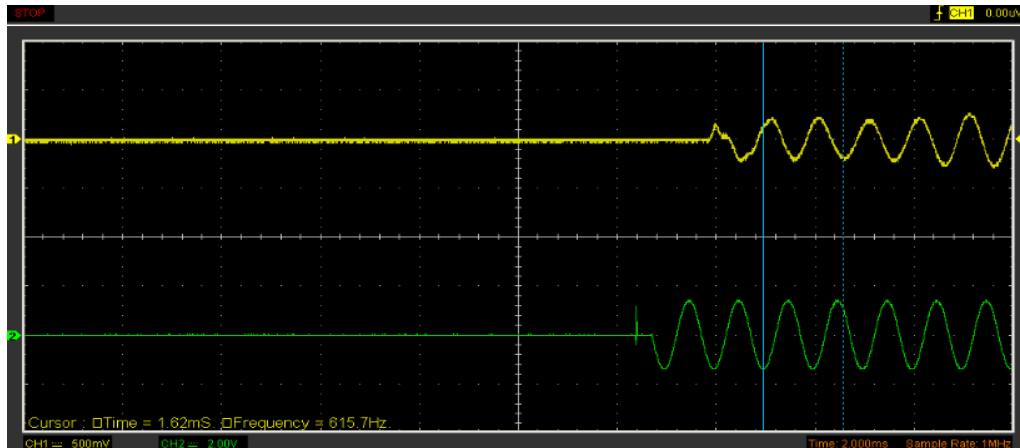
**Figura 3.63. Disminución de la potencia acústica emitida por la fuente de ruido en función de la distancia en longitudes de ondas con la fuente de control.**



**a) Retardo con una señal de referencia a 4cm.**



**b) Retardo con una señal de referencia a 15cm.**



### c) Retardo con una señal de referencia a 32cm.

**Figura 3.64. Retardos de la señal a 4, 15 y 32cm.**

La distancia entre la fuente del ruido y el altavoz cancelador es importante ya que de ello dependen los tiempos que tenga el DSP para poder llevar a cabo el ajuste de los coeficientes y obtener la señal anti fase. Se debe conocer el tiempo de procesamiento del DSP además de los retardos que se puedan producir en los sensores, amplificador y el altavoz. En la figura 3.64 se pueden ver los retardos que se producen a diferentes frecuencias. Esta medición fue realizada con el micrófono Behringer ECM8000, la etapa de acondicionamiento y el osciloscopio HANTEK 6022BE (ver anexo 4). Al realizar las medidas se determinó que el sistema tiene un retardo aproximado de 760us, más el retardo generado por la distancia de la fuente secundaria (altavoz al micrófono de error).

## 3.6.2 LOCALIZACIÓN DE SENSORES

### 3.6.2.1 LOCALIZACIÓN DEL SENSOR DE ERROR

La ubicación de los sensores de error determina, la máxima atenuación posible que puede alcanzar el sistema de control. Un algoritmo de regresión múltiple permite seleccionar la mejor configuración entre varias candidatas de las cuales se seleccionará aquella que produzca un mayor valor de correlación entre el campo acústico originado por las fuentes de control y el originado por la fuente de ruido primaria. En la selección de posibles puntos candidatos donde colocar los micrófonos hay que tener en cuenta las circunstancias prácticas en el funcionamiento real del sistema de control. Aunque se consideran todos los puntos del espacio acústico, no todos ellos ofrecen los mismos resultados de atenuación con un sistema

de control real. Esta circunstancia va a limitar el margen de colocación de los micrófonos. Por ejemplo, los peores resultados de atenuación se obtienen cuando los micrófonos se encuentran mucho más próximos de la fuente primaria de ruido que de las fuentes de control. Tampoco es aconsejable colocarlos muy alejados de las fuentes de control ya que la estabilidad del sistema de control está altamente influenciada por el retardo entre la fuente de control y el micrófono de error, siendo más inestable cuanto mayor es este retardo. Otro detalle a tener en cuenta es el ruido que pueda recoger el micrófono ajeno a la fuente primaria de ruido, por ejemplo, cuando se coloca el micrófono en un conducto de aire y el movimiento de este enmascara la señal de ruido. Esta contaminación de la señal afecta al funcionamiento del algoritmo adaptativo de control y hace que este sea incapaz de alcanzar la máxima atenuación posible.

Medio	Velocidad (m/s)	Velocidad Relativa al aire seco a 0°C
Vacío	No se propaga	-----
Caucho	54	0,16
Aire seco a 0°C	331	1,00
Aire seco a 20°C.	340	1,03
Aire seco a 100°C.	390	1,18
Corcho	500	1,51
Agua	1483	4,48
Cobre	3500	10,57
Madera	3850	11,63
Acero	5060	15,29

**Tabla 3.2. Velocidad del sonido en diferentes medios y temperaturas. (Tomada de Pérez Vega, 2001)**

Según la tabla 3.35, la velocidad del sonido en el aire a una temperatura aproximada de 20°C se transmite a 340m/s. Si se realiza una regla de tres para obtener la distancia recorrida en 76us, se tiene:

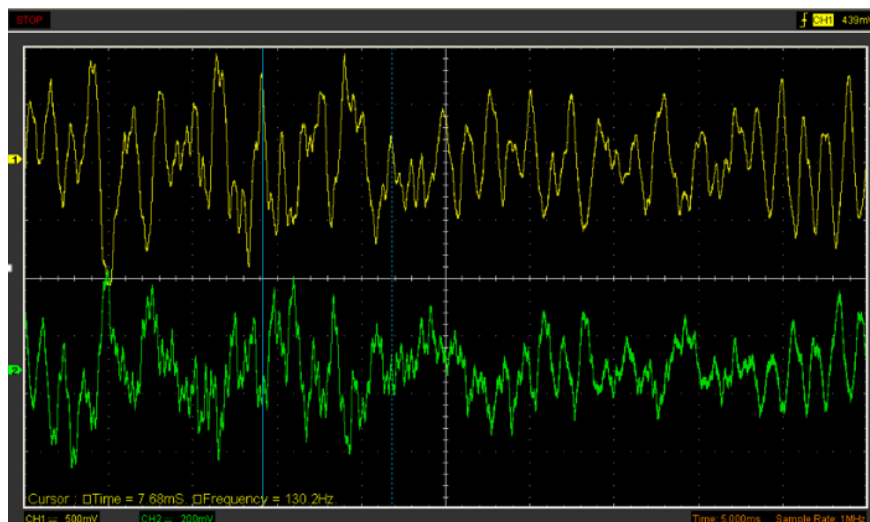
$$\frac{340m}{s} (0.00076s) = 0.25m = 2,58cm$$

Si el micrófono es colocado a 2,58cm de la fuente de ruido, se tendrían 75us para enviar la señal anti fase del altavoz atenuador, este también tendría que estar a 2,58cm de la fuente.

Con este análisis es posible determinar la distancia del micrófono de error, sin embargo, el punto de ubicación exacto si es necesario obtenerlo por medio de experimentos prácticos.

### 3.6.2.2 LOCALIZACIÓN DEL SENSOR DE REFERENCIA.

La ubicación del sensor de referencia va a determinar tanto la atenuación total que pueda lograr el sistema de control como la estabilidad del mismo. Los dos parámetros a considerar son: la contaminación de la señal de referencia y la relación señal a ruido (S/N) que ofrece el sensor. La contaminación de la señal del sensor de referencia puede producirse por una realimentación acústica desde una o varias fuentes de control. El sensor tiene que colocarse de tal manera que los efectos de realimentación sean prácticamente nulos, por tal motivo, el sensor de referencia se ubica dentro de la estructura, opuesto a la fuente de control. En la figura 3.66 se puede ver la señal de referencia (CH1) y la señal de control (CH2) generada por el algoritmo en el DSP. Existe una pequeña realimentación acústica en la señal de referencia sin embargo el sistema de control aún sigue siendo estable y la señal no es tan afectada por la señal de control proveniente del altavoz.



**Figura 3.65. Señal del compresor y la señal opuesta generada con un algoritmo adaptativo, comprobando así, la acertada ubicación de los micrófonos.**

### 3.6.3 ELECCIÓN DEL ALTAVOZ DE SALIDA.

Otro elemento del sistema de control son los altavoces (fuente secundaria o de control). Siguiendo en la línea de dispositivos de bajo coste se utilizó un amplificador de potencia debido a su buena respuesta en bajas frecuencias. Algunas características se muestran a continuación.



**Figura 3.66 Amplificador de potencia. (Fuente secundaria de control).**

Voltaje de alimentación: 100V, 120V, 220V, 230V, 240V 50/60Hz.

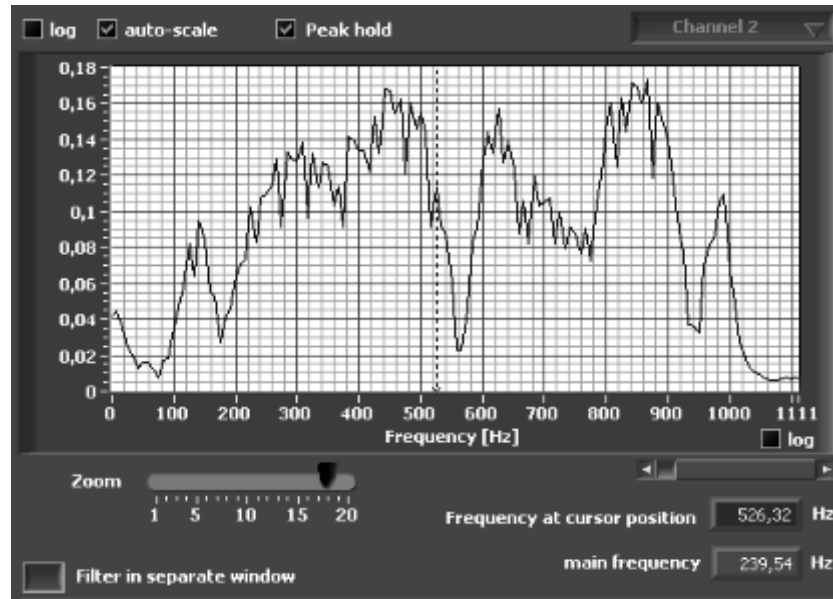
Potencia: 30W

Rango de potencia de salida: 20W

Altavoz de 8 pulgadas.

Impedancia de entrada: 1M $\Omega$ /47pF

Debido a que el amplificador no presenta especificaciones técnicas en cuanto a respuesta en frecuencia, se realizó un barrido de frecuencias desde 0 a 1000Hz. Para realizar la medición de la señal de ruido del compresor, se utilizó el software Portable Soundcard Oscilloscope v1.32 que incluye la función de análisis en frecuencia y la función de generador de señales, también se utilizó el micrófono Behringer ECM8000 y la tarjeta de sonido del computador.



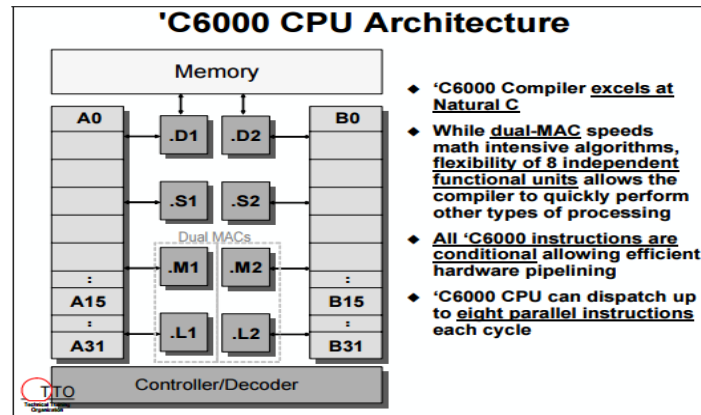
**Figura 3.67 Respuesta en frecuencia del amplificador de potencia. (Fuente secundaria de control).**

En la figura 3.67 se observa que el amplificador presenta unas variaciones en cuanto a su respuesta en baja frecuencia, sin embargo entre 100Hz y 500Hz la variación es de 1.4dB lo cual no afecta en gran medida el comportamiento del sistema de control.

### 3.6.4 ELECCIÓN DEL DSP

La complejidad del algoritmo a implementar será un factor determinante en la elección del modelo del DSP, pero también se debe tener en cuenta la frecuencia de muestreo y, en consecuencia, la frecuencia de trabajo de éste. Un dispositivo acorde con las necesidades del problema es el DSP de la familia C6000 de Texas Instruments. Esta familia ofrece el máximo rendimiento del DSP acoplándolo a una variedad de periféricos que permiten una fácil comunicación con las memorias, coprocesadores y otros dispositivos.

C6000 logra un gran rendimiento con su CPU, la cual, está construida para enviar 8 instrucciones por ciclo. La tasa de un ciclo se ejecuta aproximadamente en 1ns.



**Figura 3.68. Arquitectura CPU familia C6000 de Texas Instruments.**

El compilador C6000 desarrolla en C natural.

Mientras Dual-Mac acelera los intensivos algoritmos matemáticos, la flexibilidad de 8 unidades independientes funcionales alcanza para que el compilador realice rápidamente otros procesos.

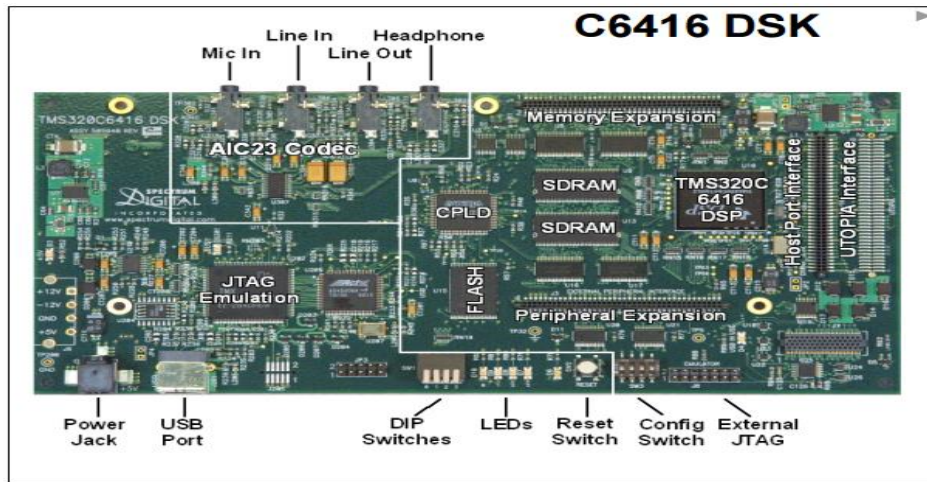
Todas las instrucciones del C6000 son condicionales permitiendo una canalización eficiente del hardware.

La CPU C6000 puede enviar hasta 8 operaciones en paralelo en cada ciclo.

Características del DSK C6416 / C6713:

- TMS320C6416 DSP: 600MHz, punto fijo, 1M Byte RAM interna.
- SDRAM Externa: 16M Bytes, C6416 – 64-bit interface.
- Flash Externa: 512K Bytes, 8-bit interface.
- AIC23 Codec: Stereo, 8 KHz–96 KHz tasa de muestreo, 16 a 24-bit samples; conectores para mic, line-in, line-out y speaker.
- CPLD: Programmable "glue" logic.
- 4 LEDs de usuario: permisos de escritura a través de CPLD.
- 4 DIP Switches de usuario: legibles a través de CPLD.
- 3 Switches de configuración: Selects power-on configuration and boot modes.
- Daughtercard Expansion I/F: Allows user to enhance functionality with add-on daughtercards.
- HPI Expansion Interface: Permite una rápida comunicación con otro DSP.
- Embedded JTAG Emulator: Provides high speed JTAG debug through widely accepted USB host interface.

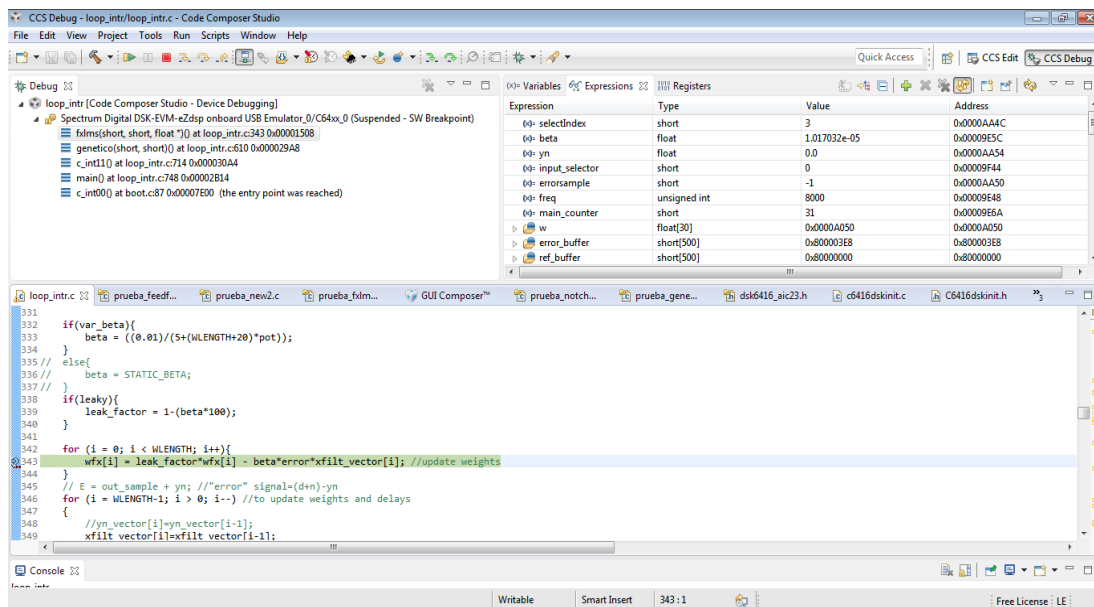




**Figura 3.69. Características y distribución de los componentes de la tarjeta TMS320C6416.**

### 3.6.4.1 CODE COMPOSER STUDIO V6

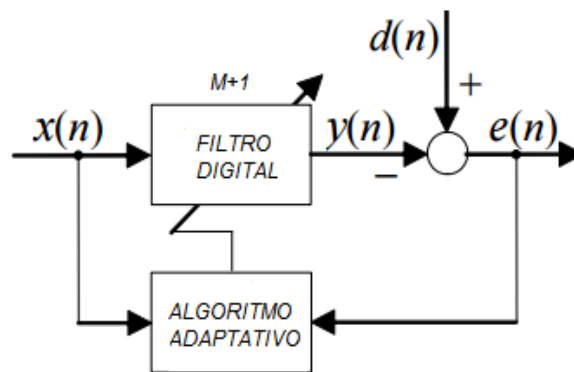
Para la programación de los algoritmos involucrados en el control activo de ruido se utilizó este software el cual tiene varias opciones de depuración y generación de código tanto en assembler como en C/C++ para la generación de software en las tarjetas de Texas instruments en este caso la tarjeta utilizada en el proyecto tms320c6416.



**Figura 3.610. Generación de código mediante software code composer studio 6.1**

### 3.6.5 DISEÑO DEL ALGORITMO ADAPTATIVO.

Una vez elegido el procesador, teniendo listo el acondicionamiento (con los respectivos sensores incluidos), la ubicación de la fuente de ruido, el altavoz de salida y conociendo las características de la señal que se desea trabajar, en este caso, la señal de ruido del compresor neumático marshal inductive motor 931373, se procede a diseñar un sistema de control adaptable. En principio, lo que se desea es que la función de transferencia del controlador siga las variaciones que surgen en el entorno acústico de cancelación, variando de alguna manera el valor de los coeficientes  $c(k)$  de un filtro. Esta es la misión del algoritmo adaptativo mostrado en la figura 3.611. A partir de las señales de entrada del sensor de referencia,  $x(n)$ , y del sensor de error,  $e(n)$ , se realizan los cálculos precisos para ir actualizando los coeficientes muestra a muestra, minimizando el nivel de la señal de error.



**Figura 3.611. Diagrama de bloques de la configuración general de un filtro adaptativo.**

$d(n)$  Respuesta deseada.

$y(n)$  Salida actual del filtro.

$x(n)$  Señal de referencia.

$e(n)$  Error.

### 3.6.5.1. ALGORITMO DE MINIMIZACIÓN DE ERROR:

#### FILTRO ADAPTATIVO FIR LMS (Medios mínimos cuadráticos).

El algoritmo LMS es el más relevante en el proyecto, ya que se necesitan ajustar los coeficientes (pesos) del filtro digital que se utiliza en la adaptación para que sea inverso al del campo acústico afectado por el ruido.

La secuencia de operaciones del algoritmo se muestra en el siguiente diagrama de flujo, así como después se muestra la secuencia matemática para un mejor entendimiento.

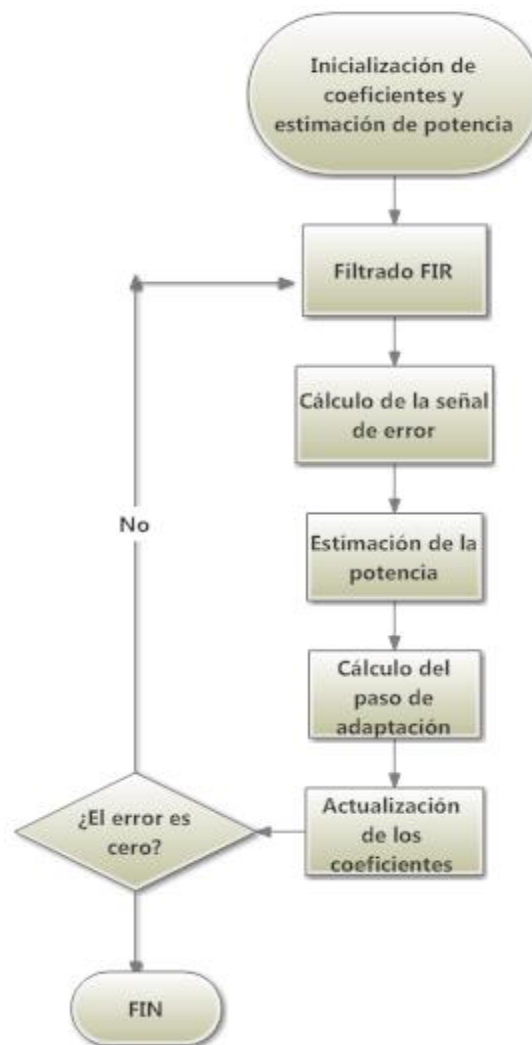


Figura 3.612. Diagrama de flujo Filtro adaptativo FIR LMS.

Inicialización de coeficientes y estimación de potencia:

$$w(k)_0 = 0$$

$$\sigma^2(0) = 1$$

Filtrado FIR:

$$y(n) = \sum_{k=0}^M w(k)_n x(n-k)$$

Cálculo de la señal de error:

$$e(n) = d(n) - y(n)$$

Estimación de la potencia:

$$\sigma^2(n) = \frac{1}{M} \sum_{k=0}^M x^2(n-k)$$

Cálculo del paso de adaptación:

$$\mu(\%) = \frac{1}{(M+1)\sigma^2} \frac{\%}{100}$$

Actualización de coeficientes:

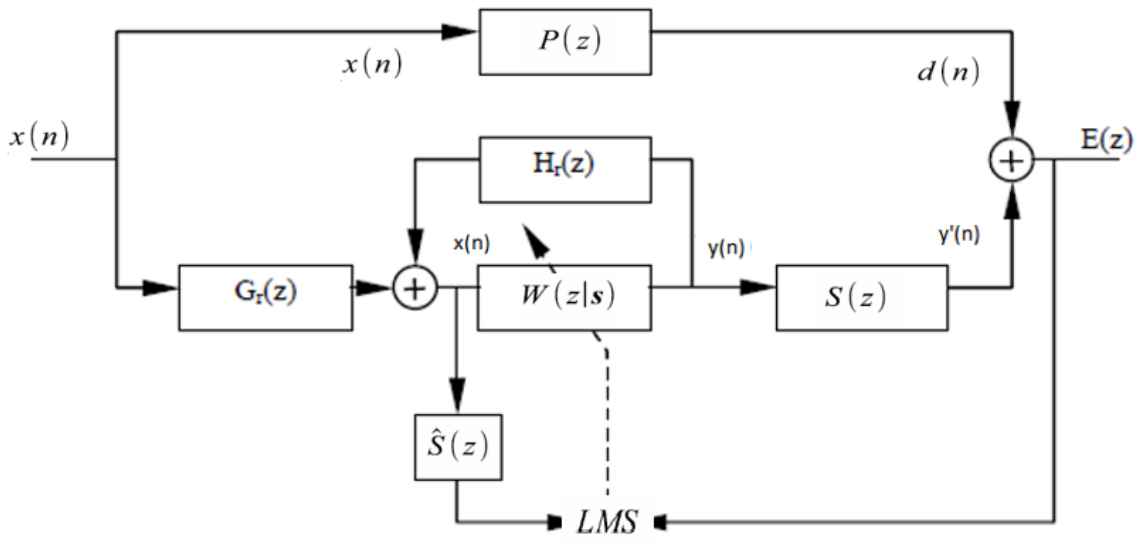
$$c(k)_{n+1} = c(k)_n - 2\mu e(n)x(n-k)$$

Se toma un signo negativo ya que las ondas se superponen (suman) unas a otras en el campo acústico, así que para obtener la cancelación se necesita obtener una realimentación negativa de la misma. Se debe volver a realizar el filtrado FIR y repetir la secuencia de pasos hasta llegar a la solución deseada.

### 3.6.5.2. Algoritmo FILTERED-X LMS (FxLMS)

#### 3.6.5.2.1 IDENTIFICACIÓN DEL CAMINO SECUNDARIO

El sistema de control se muestra en la figura 3.613. Las funciones de transferencia electroacústica involucradas, según la figura son:  $G_r(z)$ ,  $P(z)$ ,  $S(z)$ ,  $H_r(z)$ .



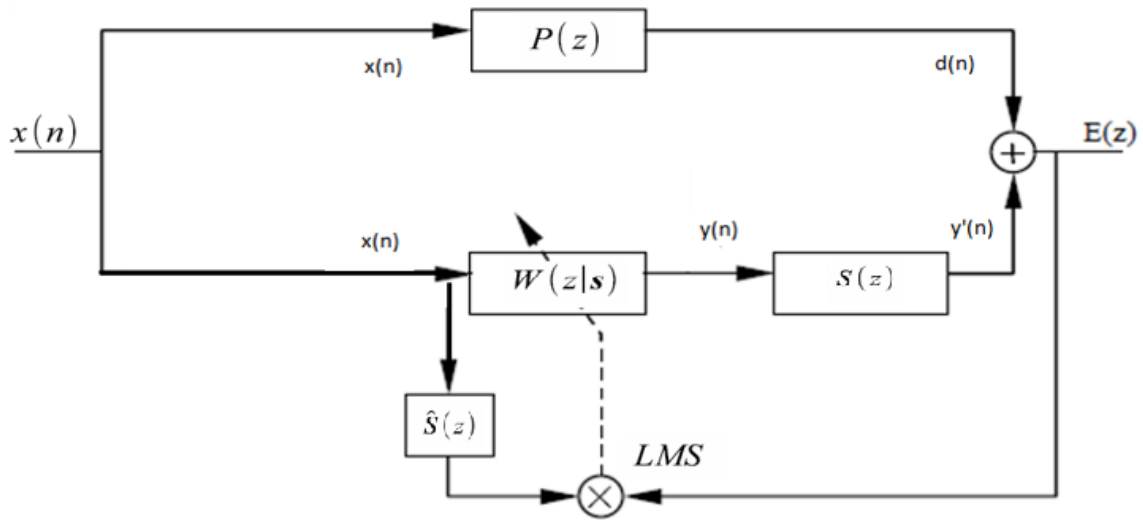
**Figura 3.613. Diagrama de bloques del sistema de control en lazo abierto feedforward con todas las funciones de transferencia involucradas.**

El algoritmo adaptativo filtered-X LMS adapta los coeficientes del filtro FIR  $W(z)$  para hallar la función de transferencia de la siguiente expresión:

$$W(z) = -\frac{P(z)}{G_r(z)S(z) - P(z)H_r(z)}$$

Dadas las condiciones anteriormente analizadas, la función de transferencia es:

$$W(z) = -\frac{P(z)}{S(z)}$$

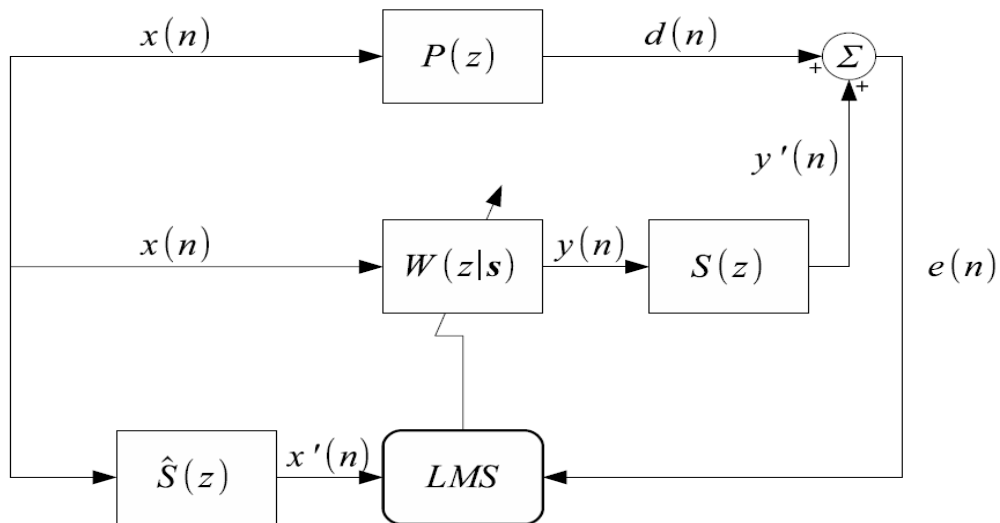


**Figura 3.614. Diagrama de bloques sistema feedforward simplificado.**

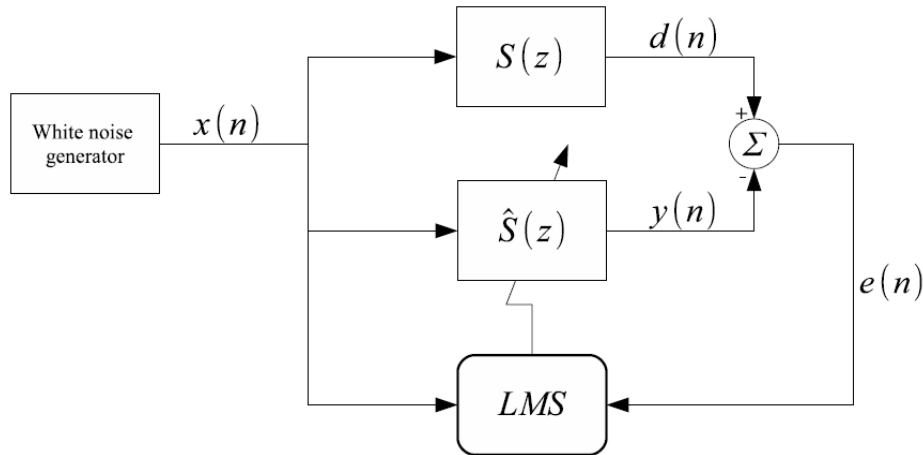
Dado que:

$$S(z)W(z) = -P(z)$$

En la figura 3.615 se puede apreciar el diagrama de bloques de una forma más explícita.



**Figura 3.615. Diagrama de bloques sistema feedforward simplificado.**



**Figura 3.616. Diagrama de bloques de identificación del camino secundario.**

El camino secundario  $S$  puede ser estimado off-line antes de que el sistema de control activo de ruido sea usado. Los coeficientes de  $S'$  son adaptados para coincidir con la función de transferencia desconocida  $S$ . Esto es realizado por una secuencia desconocida de datos de entrenamiento como salida del parlante y medido por el micrófono de error del sistema.

La estimación del proceso es realizada por un estimado adaptativo mediante el algoritmo LMS. La entrada a la ecuación de actualización es la señal de entrenamiento y el error. El error  $e(n)$  es la diferencia entre la señal verdadera y el estimado de la señal, los cuales son calculados:

$$e(n) = s(n) * x(n) - s'(n) * x(n)$$

La señal de entrenamiento debe contener frecuencias sobre todo el espectro de interés. Una señal comúnmente usada es ruido blanco, la cual tiene una potencia constante sobre todas las frecuencias.

Descripción del algoritmo.

1. Generar una señal de entrenamiento  $x(n)$ .
2. Toma la señal deseada  $d(n)$  del sensor.
3. Calcular la salida del filtro  $S'(z)$

$$y(n) = s'(n) * x(n)$$

4. Calcular la señal de error

$$e(n) = d(n) - y(n)$$

5. Actualizar el filtro FIR  $S'(z)$  usando el algoritmo LMS

$$s_{n+1} = s'_n + \mu e(n)x(n)$$

6. Repetir los pasos 2 a 5 hasta que el filtro del camino secundario converja a una solución óptima (hasta que el MSE sea minimizado). Después de la identificación del camino secundario la función de transferencia  $S'(z)$  puede ser usada en el sistema de control activo de ruido.

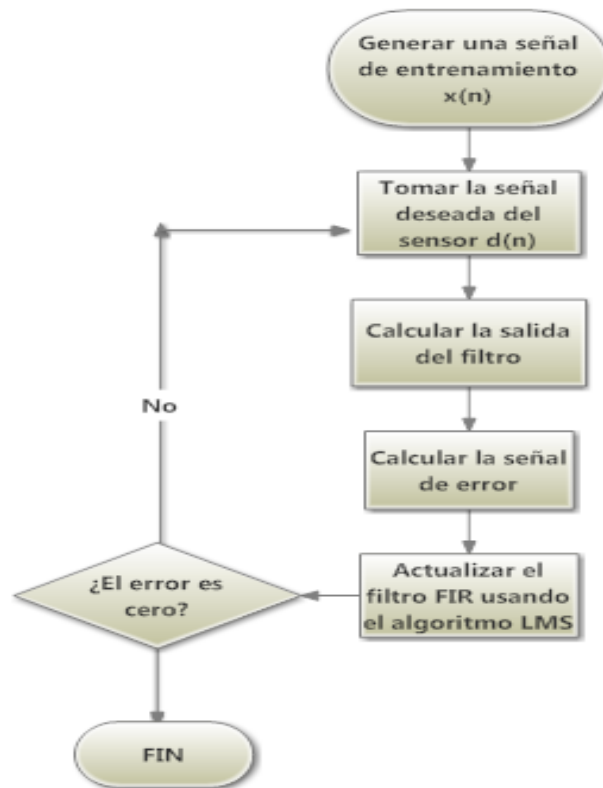
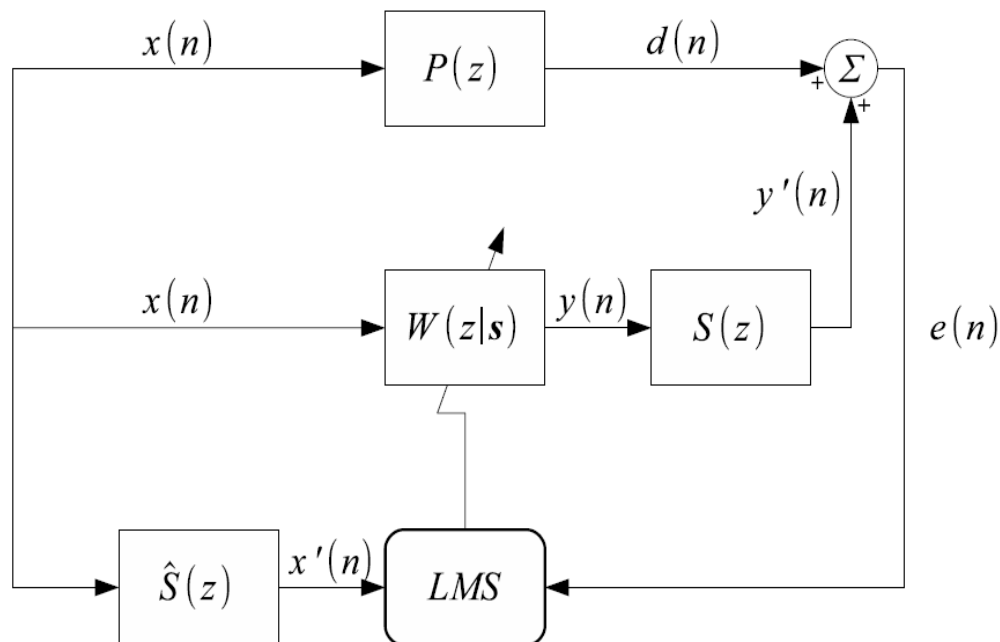


Figura 3.617. Diagrama de flujo camino secundario.



### 3.6.5.2.2 IMPLEMENTACIÓN DE CONTROL ACTIVO DE RUIDO FEEDFORWARD

Este control comprende dos micrófonos (sensores) y un parlante (fuente de control). Un micrófono de referencia es ubicado cerca de la fuente de ruido para obtener la señal coherente antes que se propague hacia el parlante. El micrófono de error es ubicado cerca a la fuente de control para crear la zona de silencio deseada. Las señales desde los micrófonos son usados como entradas para la adaptación del filtro  $w$  para determinar la salida hacia la fuente de control. Entonces el error de micrófono mide el ruido y el micrófono de error mide el rendimiento del control activo de ruido.



**Figura 3.618. Diagrama de bloques, Sistema de control Feedforward analizado.**

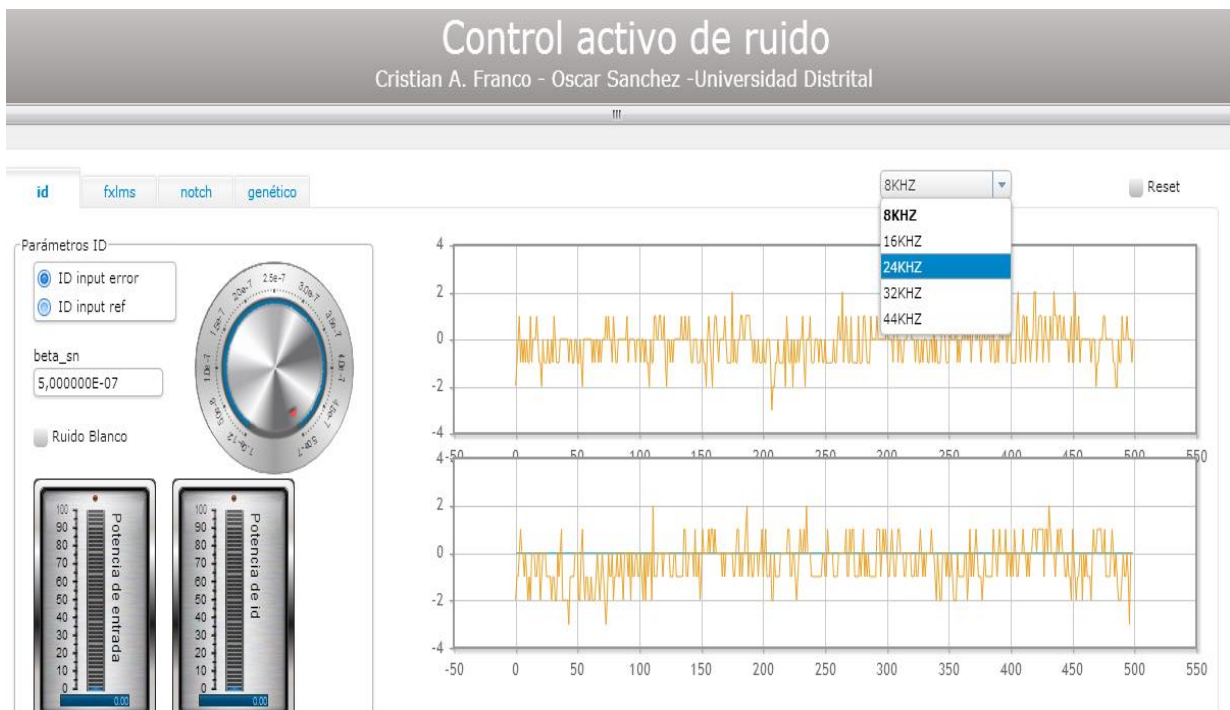
$P$  – camino primario. El camino primario  $P$  es la función de transferencia acústica desde el sensor de referencia (cerca de la fuente de ruido) al sensor de error.

$S$  camino secundario. El camino secundario  $S$  es la función de transferencia desde la salida  $y(n)$  a  $y'(n)$  e incluye el convertidor análogo digital, parlante camino acústico desde el parlante hacia el micrófono de error, convertidor digital análogo, entre otros.

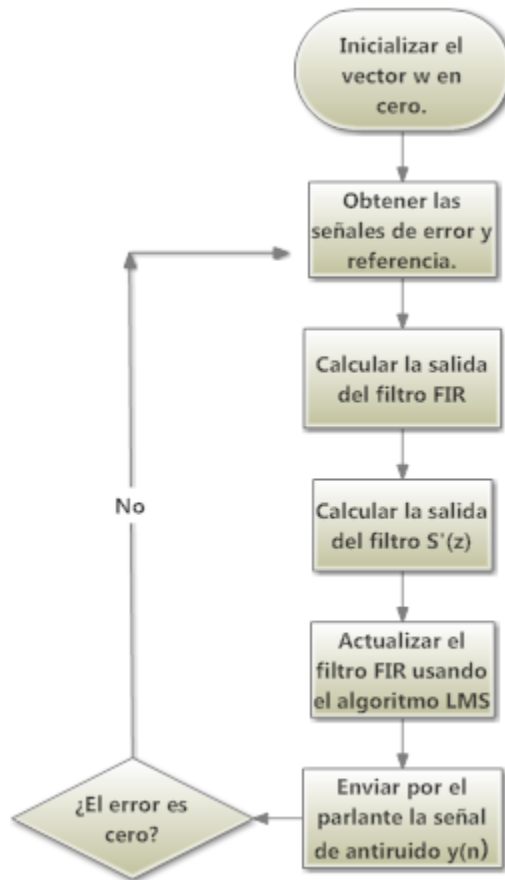
$W$  – filtro adaptativo. El objetivo del filtro adaptativo  $W$  es minimizar el ruido medido por el sensor

La función de transferencia  $P$  (camino) primario nos da información acerca de lo que sucede desde la fuente de error de error hacia el punto de atenuación en el micrófono de referencia. Ya que esta función es un filtro físico, éste no puede ser determinado con exactitud. En cambio la función de transferencia es estimada al reproducir una señal de ruido blanco desde el parlante cerca al motor hacia este micrófono.

La función de transferencia  $S$  contiene información acerca del camino secundario hacia el micrófono de error, ya que este filtro es un filtro físico no puede ser determinado con precisión. En su lugar, una función de transferencia es estimada con la señal de ruido blanco desde el parlante al micrófono.



**Figura 3.619. La identificación realizada con la interfaz gráfica implementada.**



**Figura 3.620. Diagrama de flujo FxLMS.**

Algoritmo:

1. Inicializar el vector  $w$  con ceros.
2. Obtener las señales de los micrófonos de error y referencia.
3. Calcular la salida del filtro FIR  $W(z)$ .

$$y(n) = w_n(n) * x(n)$$

4. Calcular la salida del filtro  $S'(z)$ .

$$x'(n) = s'(n) * x(n)$$

5. Actualizar el filtro FIR usando el algoritmo LMS.

$$w_{n+1} = w_n + \mu e(n)x'(n)$$

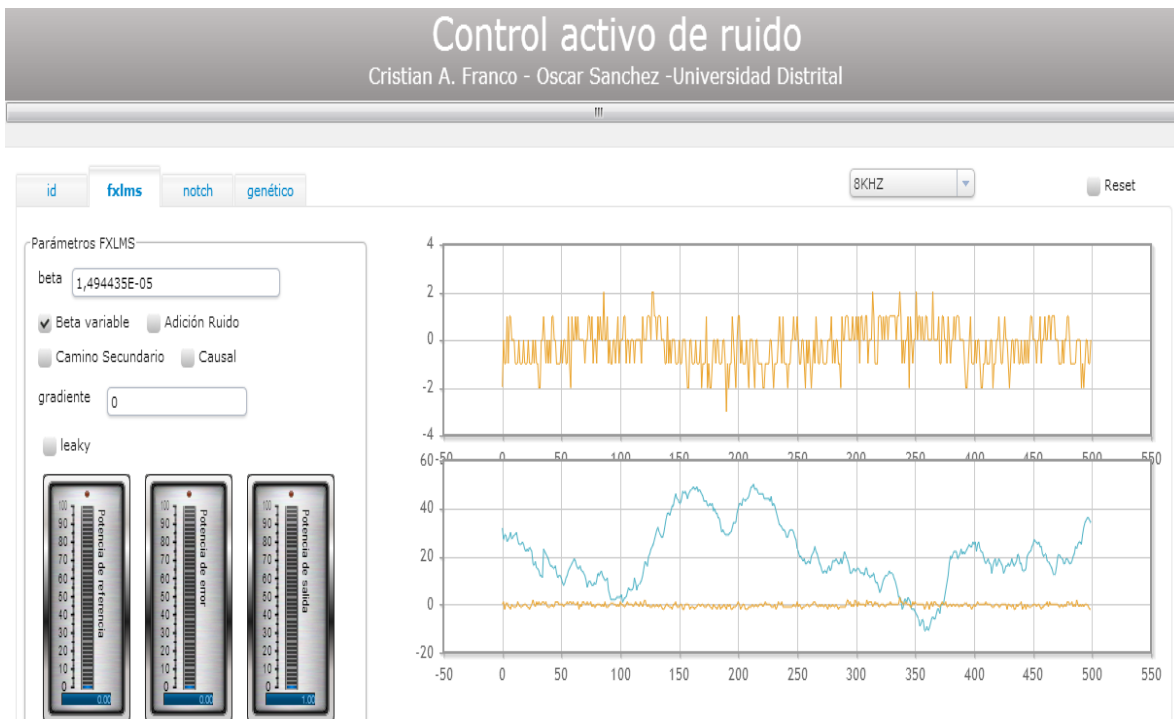
Donde  $e(n) = d(n) + y'(n)$  es la señal del micrófono de error.

6. Enviar por el parlante la señal de antiruido  $y(n)$ .

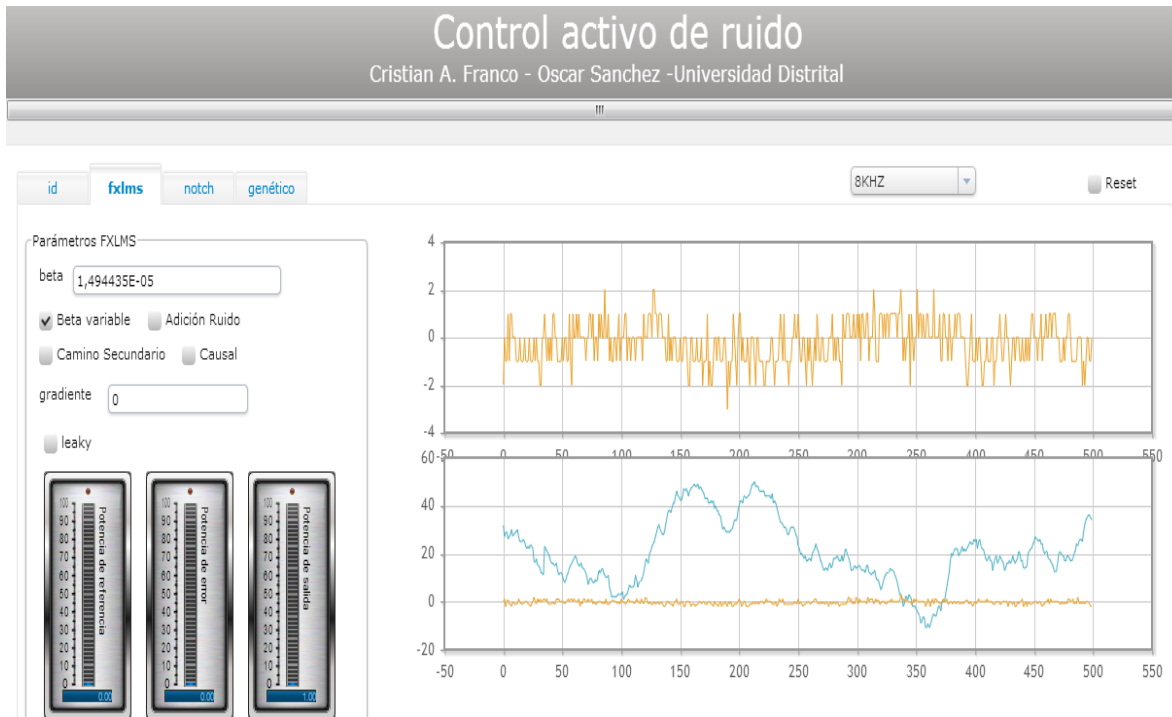
7. Repetir el paso dos al seis mientras el algoritmo esté en ejecución.

Para evaluar que tan bien funciona el sistema de control de ruido se hicieron pruebas con señales tonales, aparte de las hechas con el compresor.

En esta tesis un prototipo de control activo de ruido fue exitosamente implementado y evaluado. Se ha adquirido un conocimiento más a fondo sobre filtrado adaptativo, aritmética de punto fijo y sistemas de procesamiento de señales.



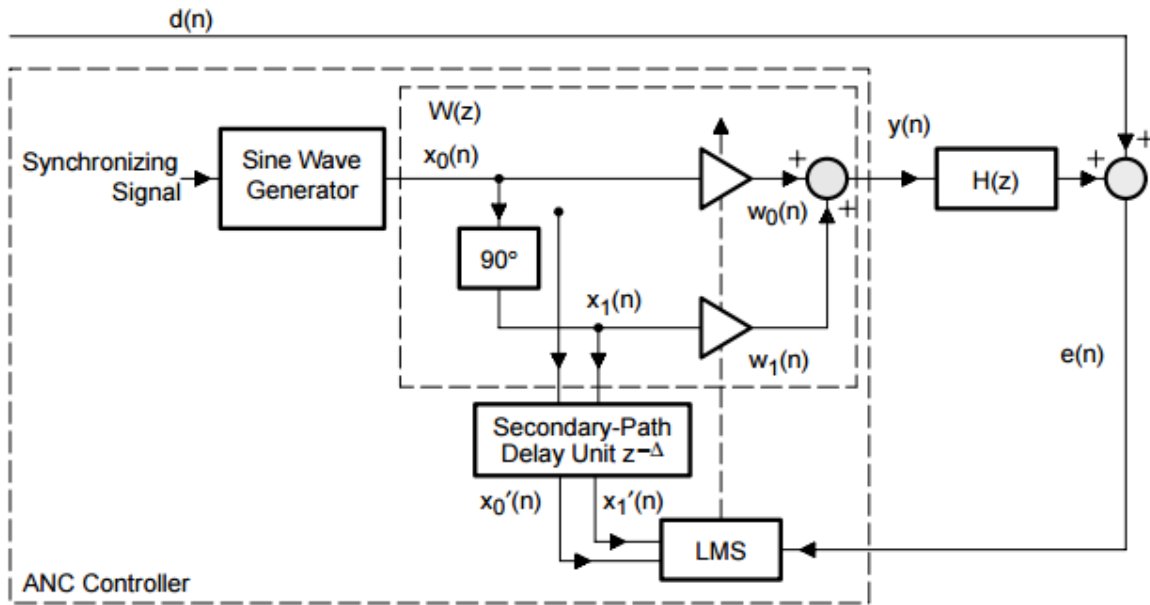
**Figura 3.621. Interfaz gráfica del algoritmo FxLMS.**



**Figura 3.622. Convergencia del algoritmo de identificación con el programa Code Composer Studio v6.1.**

### 3.6.5.3. ALGORITMOS PARA SISTEMAS DE BANDA CORTA (Filtro Notch)

Una referencia utilizada en el sistema de cancelación de ruido de banda estrecha es una onda sinusoidal con la misma frecuencia que el ruido de banda estrecha a ser cancelada. Cuando se emplea una onda sinusoidal como la entrada de referencia, el algoritmo de LMS se convierte en un filtro notch adaptativo (ver anexo 6) para eliminar los componentes espectrales primarios dentro de dicha banda centrada alrededor de la frecuencia de referencia. Por lo general se desea filtrar un rango estrecho en el espectro de frecuencias es donde se encuentra dicho ruido sin distorsionar la señal en general. El filtro notch adaptativo ofrece un fácil control de ancho de banda y la capacidad de hacer seguimiento a la frecuencia exacta de la interferencia.



**Figura 3.623. Diagrama de bloques algoritmo con filtro Notch.**

Diagrama de bloques de este sistema de banda estrecha con dos pesos adaptativos

$$x_0(n) = A \cos(k\omega_0 n)$$

$$x_1(n) = A \sin(k\omega_0 n)$$

Donde  $\omega_0$  es la frecuencia fundamental,  $k$  es el índice de armónicos,  $A$  es la amplitud de la señal de referencia, y  $n$  es el índice de tiempo. El generador de onda sinusoidal puede ser implementado por un vector con los datos de dicha onda o por un resonador digital.

Estas dos señales se ponderan por separado y después se suman para producir la señal de cancelación  $y(n)$ :

$$y(n) = w_0(n)x_0(n) + w_1(n)x_1(n)$$

La magnitud y la fase de esta señal de referencia se ajustan en el controlador, el cual alimenta el altavoz que actúa como fuente de control para cancelar los componentes de ruido correspondientes. El algoritmo LMS actualiza los pesos del filtro para minimizar el error residual  $e(n)$ :

$$w_0(n+1) = w_0(n) - \mu e(n)x_0(n)$$

$$w_1(n+1) = w_1(n) - \mu e(n)x_1(n)$$

### 3.6.5.4. ALGORITMO GENÉTICO

El algoritmo genético utilizado se ha simplificado respecto al algoritmo genético general. En el proceso de selección sólo se escoge aquel cromosoma que presente menor error cuadrático medio, es decir, sólo se selecciona un único padre. La población de cromosomas de la siguiente generación se forma mutando este cromosoma, es decir, añadiéndole un pequeño ruido blanco tantas veces como número de cromosomas tenga la población. La ventana de análisis debe ser suficientemente grande para poder evaluar sin error el error cuadrático medio. La realización de este algoritmo se puede ver en el anexo 5. A continuación se realiza su descripción:

Se usó un filtro FIR:

$$y(n) = \sum_{k=0}^M c(k)_n x(n - k)$$

Se genera aleatoriamente la población inicial de cromosomas (conjunto de coeficientes) que pueden minimizar el error.

$$s_i = [c_i(0), c_i(1), \dots, c_i(n - 1)]$$

Evalúa cada grupo de coeficiente, midiendo su aptitud, la cual se determina mediante la medición de su respectivo error cuadrático medio:

$$\sigma_e = \frac{1}{32} \sum_{k=0}^M e_i^2$$

Obteniendo estos resultados se escoge el que menor error cuadrático tenga. Luego se muta este conjunto de coeficientes de la siguiente manera:

$$c'_i(k) = c_i(k) + \sigma_i D$$

Donde D es la amplitud de la mutación y  $\sigma_i$  es un número aleatorio entre [-1,1]. Este proceso se repite hasta la obtener convergencia deseada, si es necesario se varían las variables D y  $\sigma_i$  hasta obtener un resultado óptimo.

## **4. RESULTADOS.**

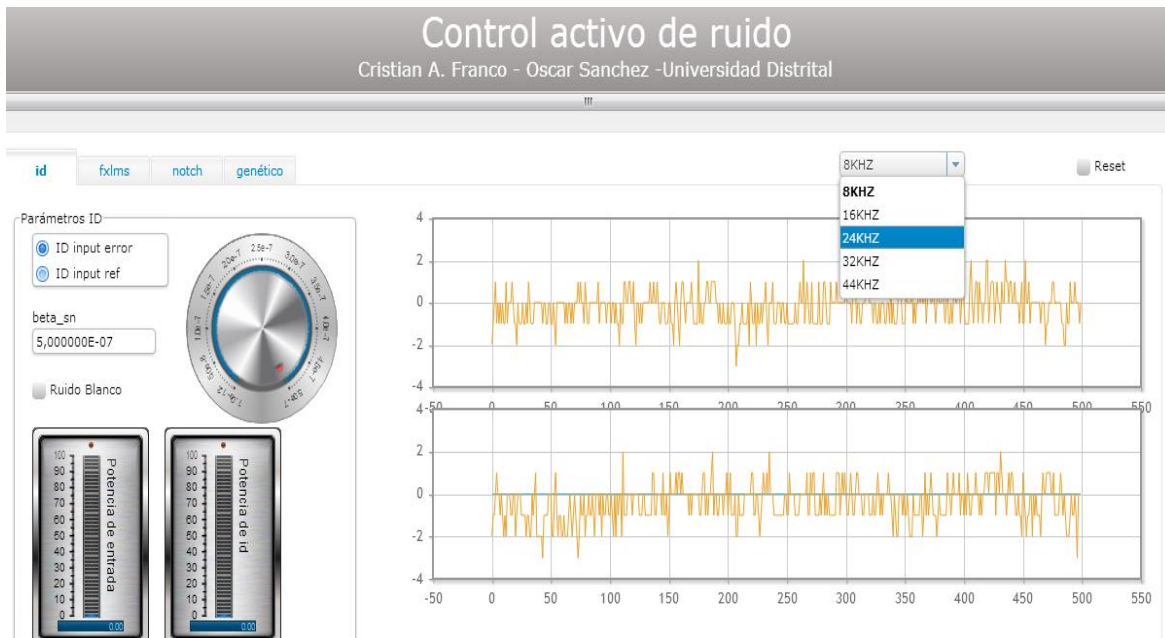
Buscando implementar un sistema práctico para la cancelación de ruido acústico del compresor Marshal Inductive Motor, se desarrolló un sistema adaptativo utilizando la tarjeta TMS320C6416 de Texas Instruments. Como se vio en el capítulo anterior el sistema adaptativo desarrollado en algunos casos tenía que incluir la identificación del camino secundario. Se optó por la identificación offline, ya que el entorno de medición y cancelación es estático, pues el procesador interactúa con el mundo real a través de los convertidores Análogo/Digital y Digital/Análogo y deja libre el proceso de adaptación requiriendo una menor demanda computacional. La inversión de tiempo y esfuerzo ha sido alta para poder entender y utilizar en forma adecuada el DSP, aunque su campo de aplicación es bastante amplio, con el presente trabajo se logró un avance significativo en el entendimiento de esta herramienta tan útil en el campo del control. En este capítulo se presentan los resultados finales de los algoritmos expuestos.

### **4.1 INTERFAZ**

Como complemento al proyecto de grado se realizó una interfaz gráfica mediante el software Code Composer Studio, con el cual se manejan y visualizan al mismo tiempo los datos y variables pertinentes de los algoritmos realizados para el proyecto en tiempo real, para una mejor visualización y operación del sistema a la hora de realizar las pruebas diagnósticas del mismo.

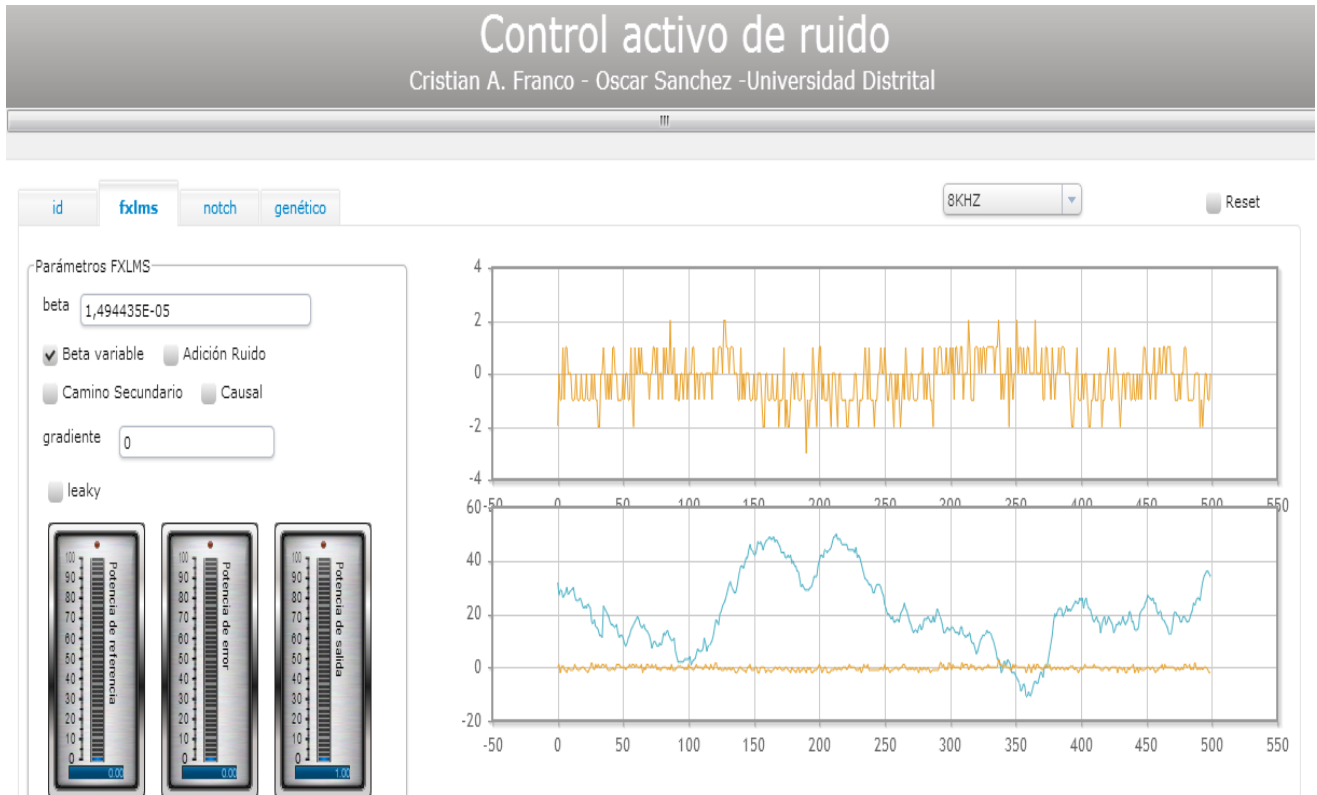
En la interfaz se pueden apreciar los algoritmos que se implementaron para hacer las pruebas comparativas para el control activo de ruido, además de las señales en tiempo real de los dos micrófonos de entrada y la señal de salida generada por los algoritmos implementados en la tarjeta, además de unos parámetros generales para todos los algoritmos los cuales son la frecuencia de muestreo del DSP, y un reset para inicializar a cero las variables de nuevo.





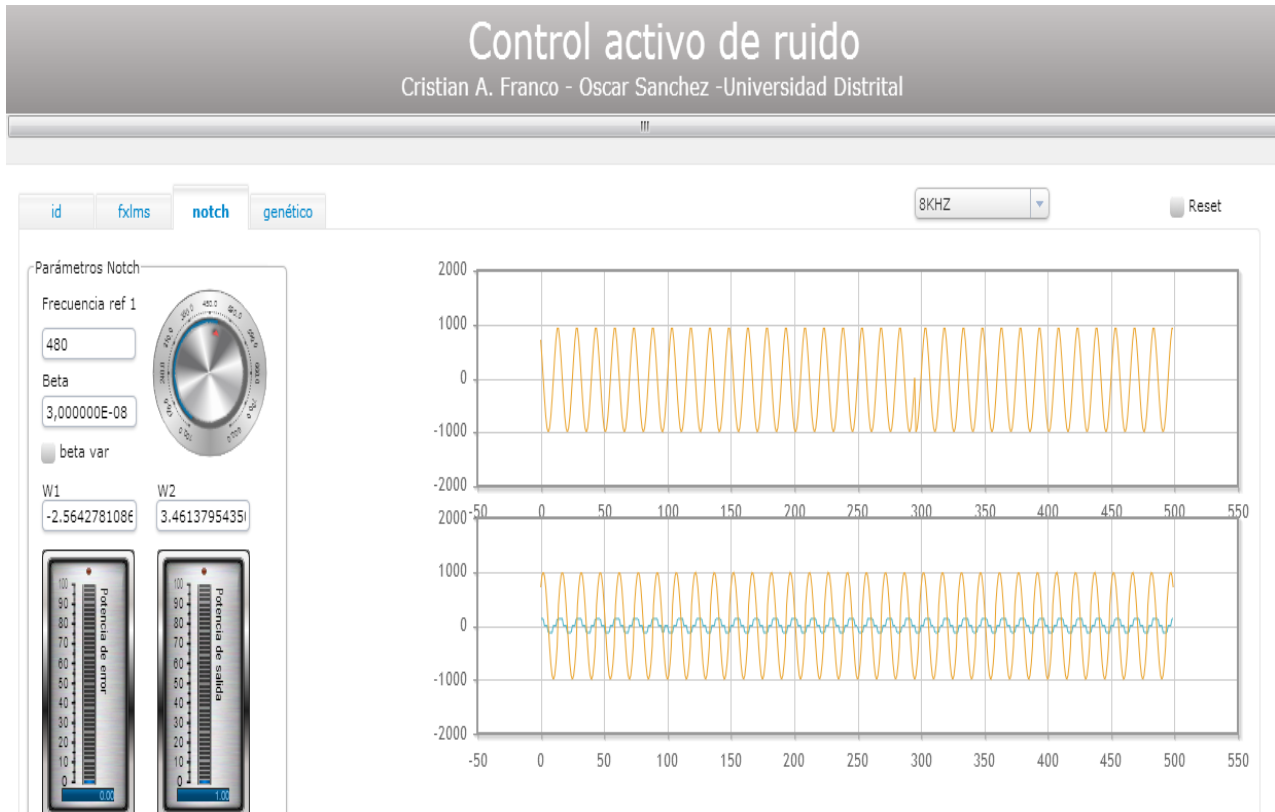
**Figura 4.1** En la interfaz se muestran los parámetros más importantes a configurar.

Lo primero que tenemos que hacer para realizar el control activo de ruido es una identificación del camino secundario el cual comprende el campo acústico del micrófono de error al parlante, y la parte digital que comprende de la recepción del micrófono, la entrada del DSP, tanto como la salida y la acción del parlante que usamos como fuente de control. Al igual que la identificación del campo acústico que afecta la recepción de sonido por el micrófono de referencia a la fuente de control para cálculos con respecto a la posible realimentación que pudiera existir entre los dos micrófonos del sistema. Esto mediante un ruido blanco generado por programación el cual genera un espectro de frecuencias uniforme el cual nos ayuda ver la respuesta de este campo en un rango amplio de frecuencias.



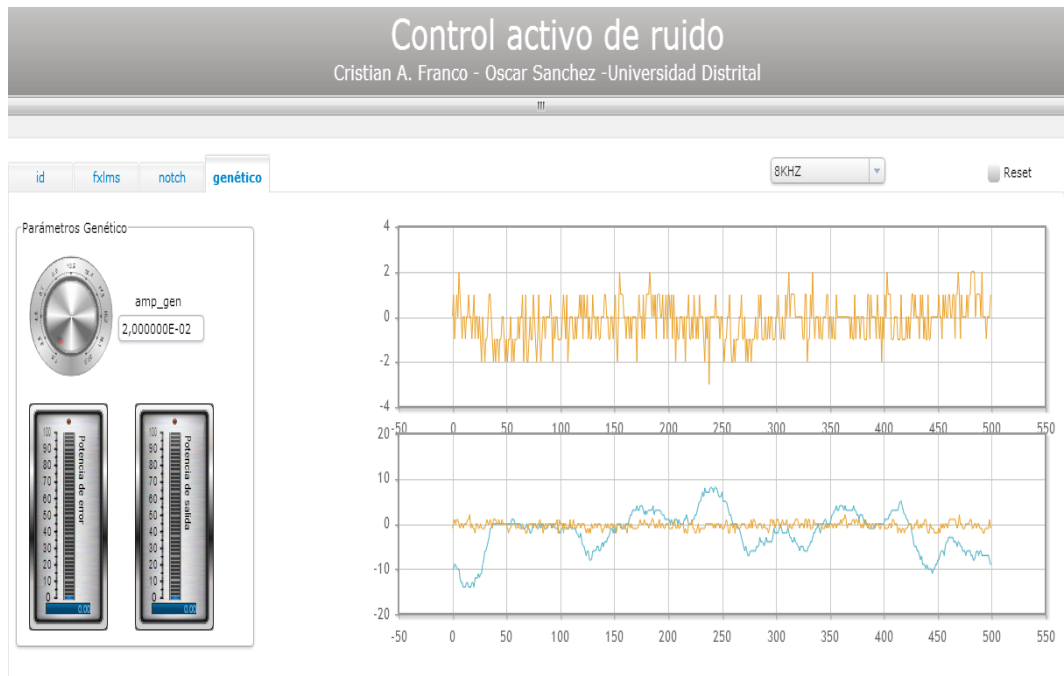
**Figura 4.2. Algoritmo FxLMS en la interfaz.**

Para el algoritmo FXLMS que fue el que dio mejores resultados prácticos, se implementó esta interfaz, con la cual se pueden visualizar tanto la señal de entrada de los dos micrófonos (referencia y error) y salida del DSP, además de varias datos relevantes para el control como la velocidad de convergencia (beta), el cual tiene una opción en la cual se hace su cálculo dinámico según la potencia de la señal de entrada del micrófono de referencia, además de la opción de añadir un ruido de medio cero al cálculo para generar estabilidad al algoritmo (FXLMS-NA), el gradiente de error en tiempo real y un factor “leaky para controlar los efectos producidos por posibles desbordamiento de los datos de procesamiento.



**Figura 4.3. Algoritmo Notch en la interfaz.**

El segundo algoritmo es un filtro Notch en el cual ajustamos la frecuencia a atenuar mediante un dial, en un rango de 100Hz a 800Hz, con la cual se genera una onda seno y coseno como entradas del filtro además de la visualización de las señales, y los pesos que se actualizan en la ejecución del filtro, además del valor de convergencia y la opción de calcular este valor mediante la potencia de la entrada del micrófono.



**Figura 4.4. Algoritmo de máxima pendiente optimizado con algoritmo genético en la interfaz.**

El último algoritmo implementado es un algoritmo de máxima pendiente optimizado con búsqueda de coeficientes por medio de un algoritmo genético, en el cual se buscan los coeficientes que mejor se adapten al control mediante un algoritmo basado en la selección “natural”, o en este caso por selección del vector de coeficientes que generen un menor error cuadrático medio, en un instante dado.

#### **4.2 PARÁMETROS DEL ALGORITMO ADAPTATIVO DE CONTROL.**

Los parámetros que gobiernan el algoritmo global de control y que determinan el funcionamiento del sistema para una aplicación concreta de control activo de ruido son:

Parámetros del filtro adaptativo de control:

- Tipo de filtro FIR o IIR.
- Número de coeficientes del filtro.
- Tipo de algoritmo adaptativo.
- Factor de convergencia del algoritmo adaptativo u otros tipos de parámetros de convergencia.

- Factor "leakage" de los algoritmos adaptativos.

Parámetros en la estimación de la función de transferencia de cancelación:

- Estimación on/off line.
- Estimación de camino secundario con o sin ruido aleatorio auxiliar.
- Longitud del filtro de estimación.
- Factores de convergencia de los algoritmos adaptativos de estimación.
- Factores "leakage".

La selección de los valores para cada uno de estos parámetros:

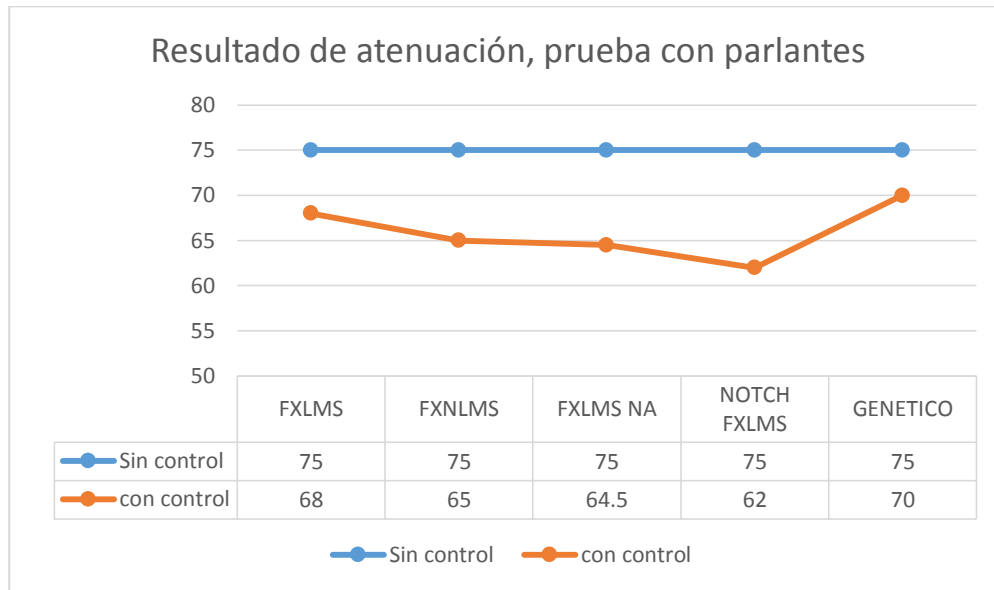
- Tipo de ruido acústico, de banda ancha (aleatorio) o banda estrecha (periódico).
- Tipo de controlador.
- Robustez o estabilidad de funcionamiento.
- Complejidad de diseño (mayor o menor requerimiento computacional).
- Máxima cancelación obtenida.
- Velocidad de adaptación a los cambios de la función de transferencia de cancelación.

### 4.3 PRUEBA DE CONTROL CON PARLANTES DE COMPUTADOR.

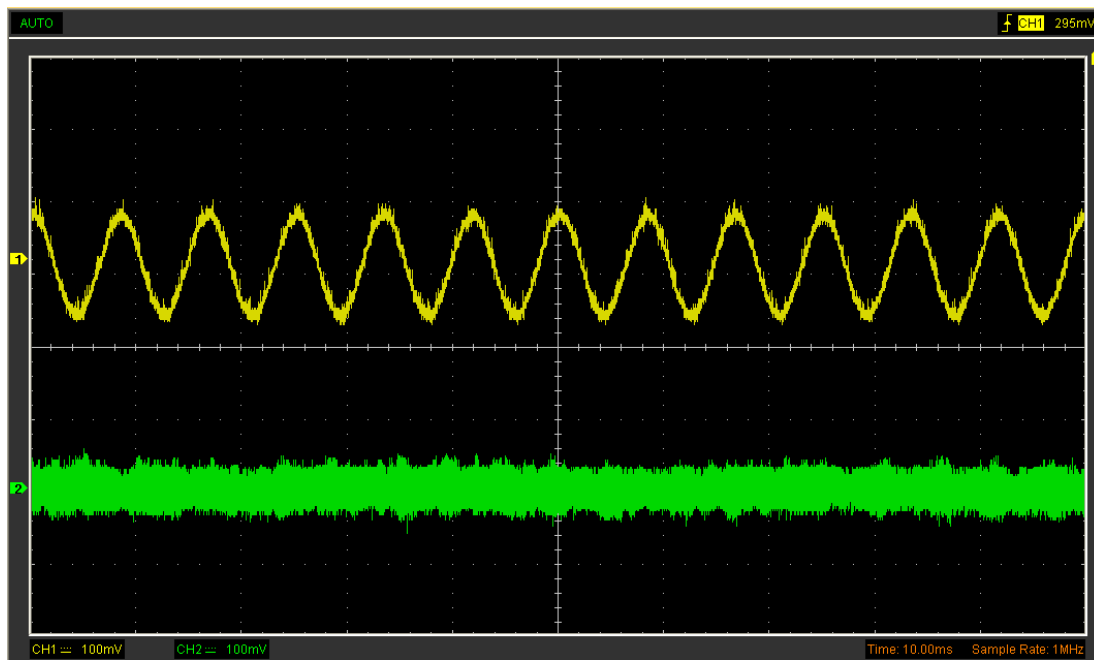
Para la verificación del correcto funcionamiento de los algoritmos y de la implementación de la planta se realizó una prueba piloto con unos parlantes de computador estándar genius SP-P110 en donde se generó una onda de 120Hz. Se obtuvieron los siguientes resultados prácticos con los distintos algoritmos:

	DBA	DBA
ALGORITMO	SIN ATENUAR	ATENUACION
FXLMS	75	68
FXNLMS	75	65
FXLMS NA	75	64.5
NOTCH FXLMS	75	62
GENÉTICO	75	70

**Tabla 4.1. Resultados de atenuación obtenidos con los diferentes algoritmos de prueba.**



**Figura 4.5. Resultados de atenuación con prueba de parlantes.**



**Figura 4.6 Cancelación Notch con parlantes, donde la señal CH1 es de control y la señal CH2 es la señal de error.**



**Figura 4.7. Parlantes sp-p110, utilizados para simular la fuente de ruido.**

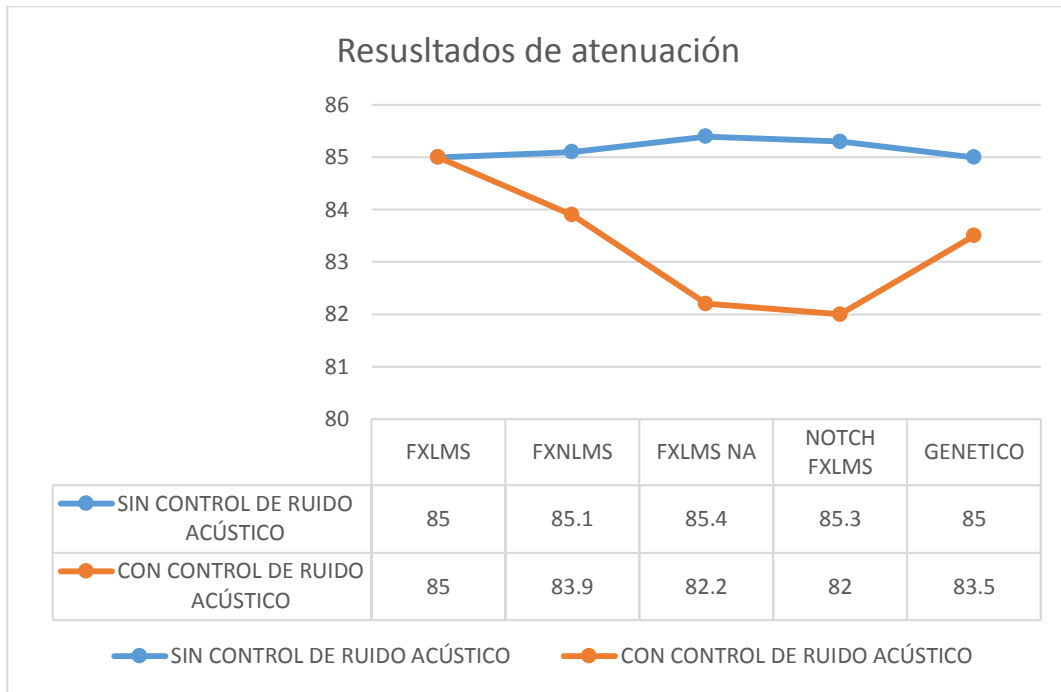
De aquí se tiene un punto de partida muy diciente sobre el comportamiento de los algoritmos en la configuración de la planta y que los mismos algoritmos tienen la actuación deseada (atenuación), sobre frecuencias tonales. Se realizó una prueba también con 440Hz para verificar el funcionamiento al cambio de frecuencias y este tuvo resultados óptimos.

#### **4.4 PRUEBA DE CONTROL CON RUIDO DEL COMPRESOR.**

No existen unos parámetros universales de funcionamiento. Cambiar de la aplicación de los parlantes a la señal real del compresor implica seleccionar aquel algoritmo que mejores resultados ofrece y que se obtuvieron mediante múltiples ensayos de atenuación en condiciones reales de funcionamiento. En las pruebas que se realizaron con los distintos algoritmos, destacan las que aparecen en la tabla, obteniendo los datos mostrados en el punto en el espacio seleccionado:

	<b>DBA</b>	<b>DBA</b>
<b>ALGORITMO</b>	<b>SIN CONTROL DE RUIDO ACÚSTICO</b>	<b>CON CONTROL DE RUIDO ACÚSTICO</b>
<b>FXLMS</b>	85	85
<b>FXNLMS</b>	85.1	83.9
<b>FXLMS NA</b>	85.4	82.2
<b>NOTCH FXLMS</b>	85.3	82
<b>GENETICO</b>	85	83.5

**Tabla 4.2. Resultados de atenuación obtenidos con los diferentes algoritmos implementados para el sistema de control de ruido acústico.**



**Figura 4.8. Gráfico de atenuación de acuerdo a los diferentes algoritmos probados.**

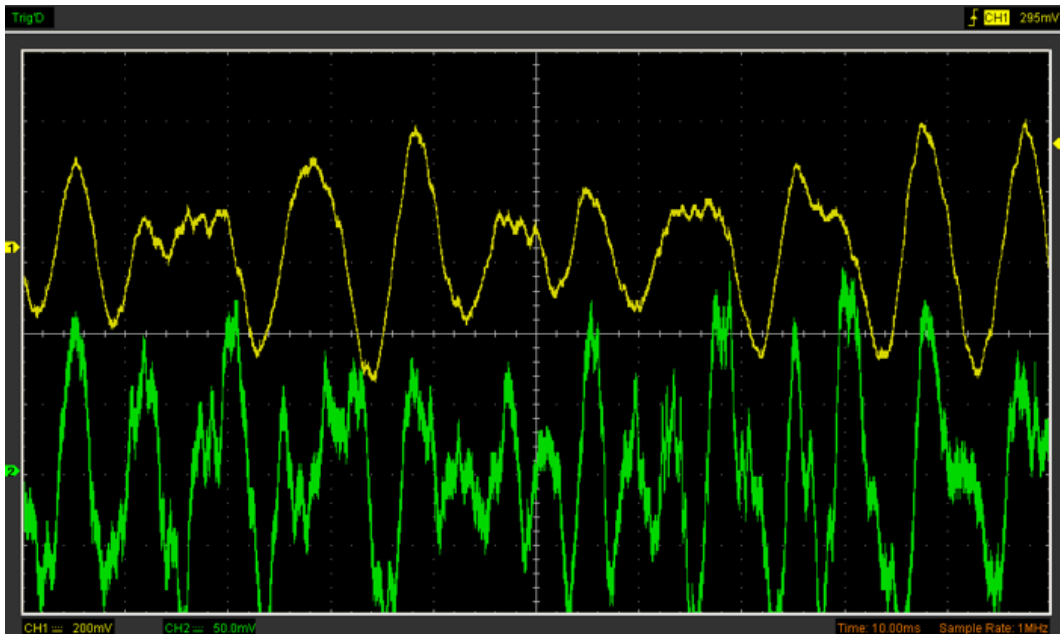
#### 4.5 FXLMS

Se realizó la identificación del campo acústico del camino secundario mediante la estimación de la función de transferencia altavoz/micrófono de error, de manera separada al programa principal (offline). Esto mediante la realización por código de un ruido blanco (ruido con magnitud similar en un rango grande de frecuencias de interés) el cual es enviado y reproducido en el amplificador y cuyo sonido es recogido nuevamente por el micrófono.

Implementando el algoritmo propuesto se obtuvo un vector de pesos de longitud de 30 pesos el cual se almacenó en una variable de programa.

Luego se implementó dicho algoritmo variando el factor de convergencia experimentalmente hasta alcanzar un rango de respuesta deseable.





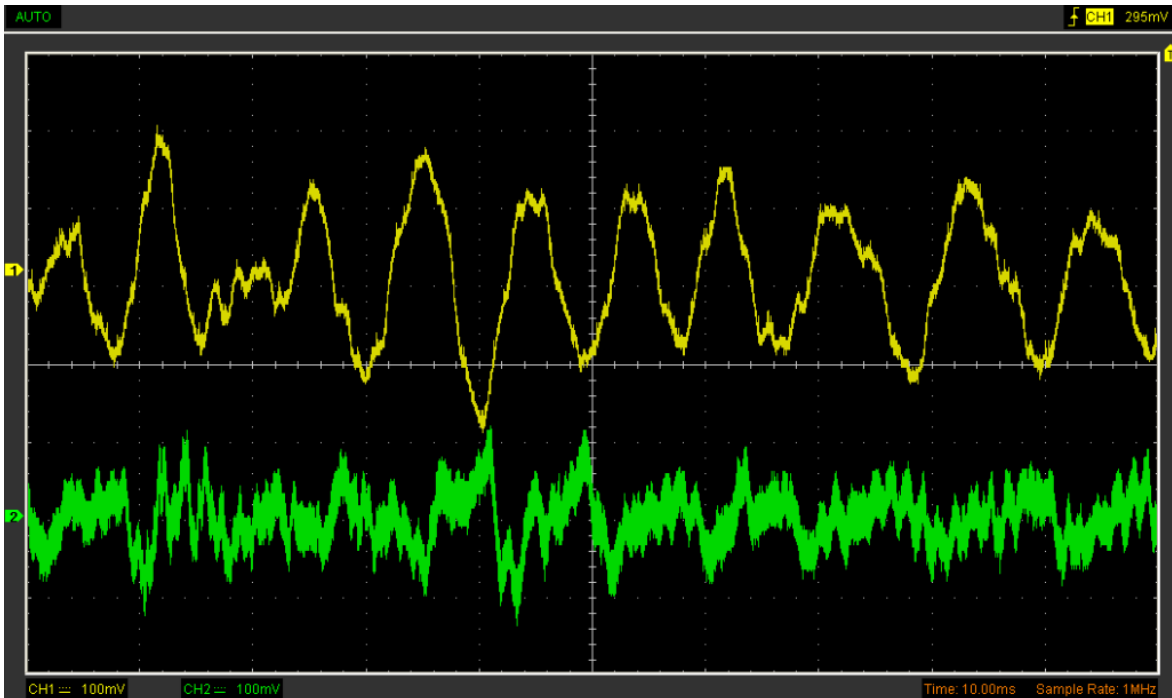
**Figura 4.9. Resultado del algoritmo FXLMS. CH1 es la señal de error y CH2 la señal de control.**

Además se varió la distancia del amplificador para observar las variaciones que el mismo incluía en la adaptación del algoritmo.

En el mejor escenario de cancelación se atenuó algo más de 2.5dBs en campo abierto, teniendo el amplificador al lado de la estructura encerrada. Este algoritmo reacciona con ciertas inestabilidades al cambio acústico y a la disposición de los transductores, entrando en un nivel de desbordamiento y por consiguiente saturación y divergencia del algoritmo.

Este algoritmo es proclive en efectos de realimentación de la salida del altavoz y el micrófono de referencia, por lo cual los mejores escenarios de cancelación se produjeron cuando el micrófono estaba dispuesto de forma direccional, y dentro de la estructura encerrada.

Finalmente se implementó un algoritmo en el cual se generaba la onda de 120 Hz digitalmente mediante una tabla de valores generada en Matlab (ver anexo 11), lo que quita el uso del micrófono de referencia y la posible realimentación y afectación del sistema por esta causa, esta disposición fue la más efectiva para este algoritmo en atenuación ya que en unos casos alcanza a atenuar hasta 4db. (Figura 4.9).

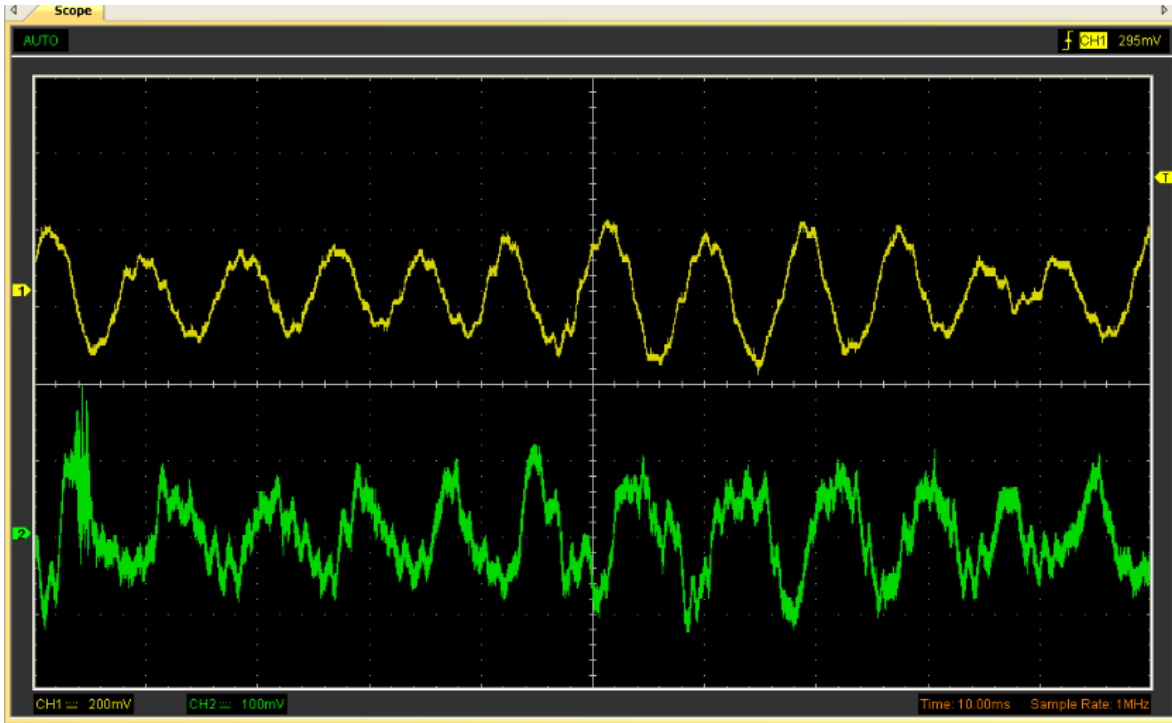


**Figura 4.10. Resultado del algoritmo FXLMS generando una onda de 120Hz. CH1 es la señal de error y CH2 la señal de control.**

#### 4.6 FXLMS-NA

La adición de ruido a los datos de entradas filtrados por el camino secundario causó una mejor estabilidad del algoritmo, aunque la velocidad de convergencia es menor, por lo que se tuvo ajustar de nuevo el factor de convergencia, para obtener valores los deseados. Este algoritmo es más estable frente a los cambios de disposición de los transductores y no llega a un estado de divergencia tan rápido como su predecesor (FxLMS).

En cuanto atenuación se observa un comportamiento similar al FXLMS.

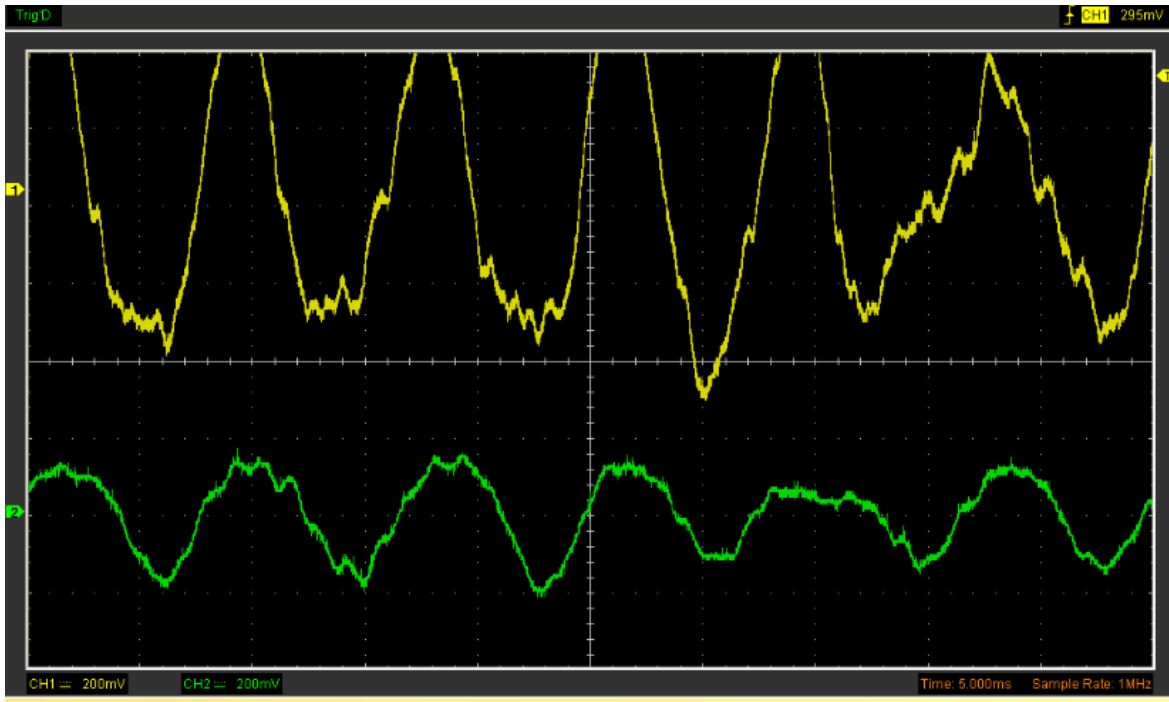


**Figura 4.11. Resultado del algoritmo FXLMS-NA. CH1 es la señal de error y CH2 la señal de control.**

Este algoritmo es más tolerante a la realimentación producida entre los micrófonos aunque su afectación es considerable cuando esta realimentación es directa.

#### 4.7 FXNLMS

En este algoritmo el factor de convergencia es variable de acuerdo a la cantidad de potencia generada por la entrada, por lo que el mismo es grande en el momento en el que el algoritmo empieza a converger y va disminuyendo cuando el mismo va llevando al mínimo global, para evitar desviación y oscilación alrededor de este punto.

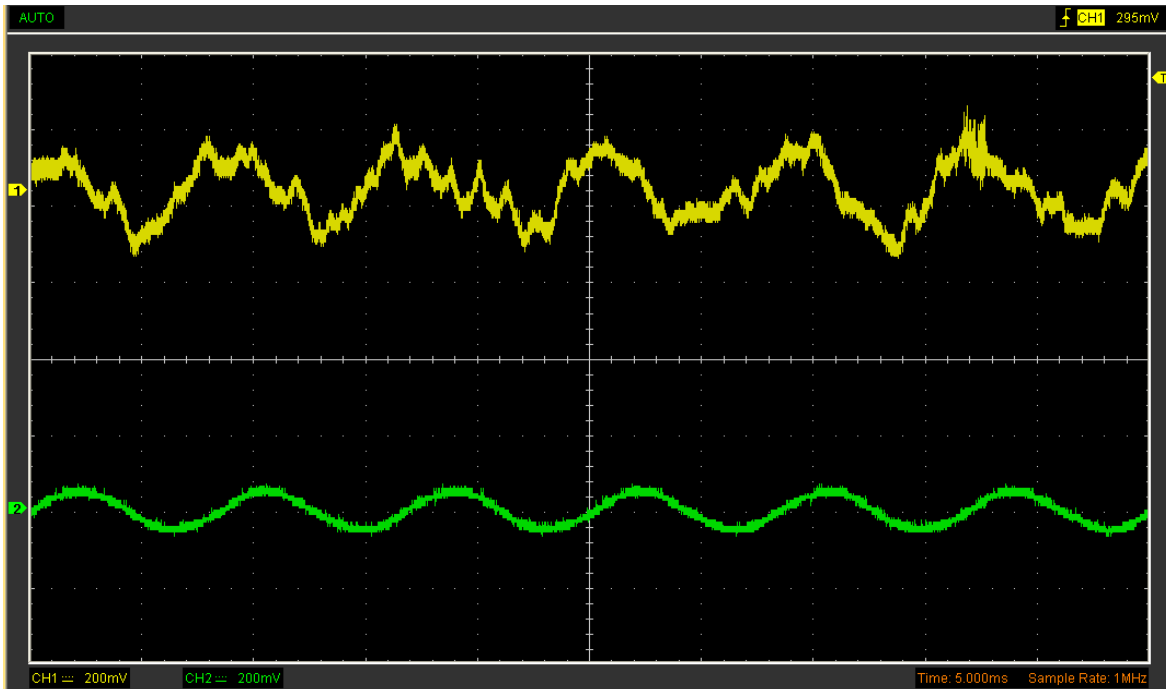


**Figura 4.12. Resultado del algoritmo FXNLMS. CH1 es la señal de error y CH2 la señal de control.**

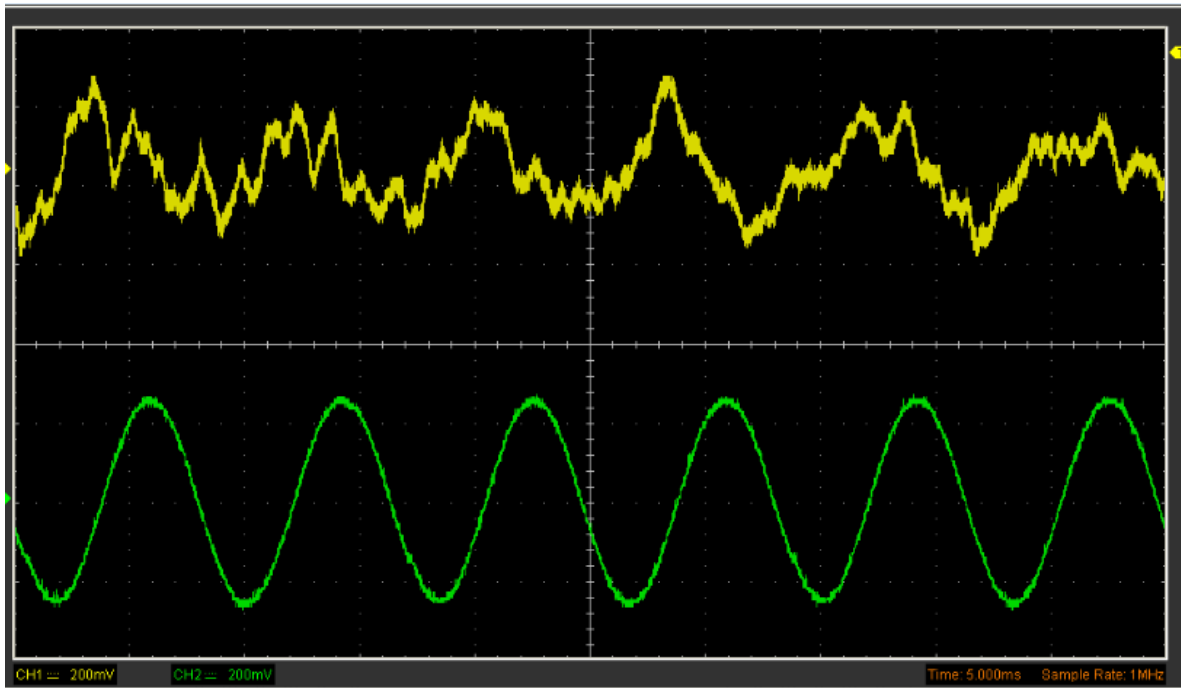
En la implementación del mismo se notó un tiempo de convergencia mucho más rápido, además de una convergencia más estable, aunque obteniendo los mismos resultados de atenuación que con el FxLMS.

#### **4.8 ALGORITMO ADAPTATIVO NOTCH**

Teniendo la caracterización del espectro de frecuencia del ruido causado por el compresor Marshal se visualizó que el ruido es producido con una gran magnitud en la frecuencia de 120Hz (una vez aislado pasivamente mediante la estructura), por lo cual, se decidió implementar un método de adaptación para atenuar esta frecuencia y evaluar los resultados producidos por el mismo. Este algoritmo fue el que consiguió una mayor atenuación acercándose a los 5 dB en recinto abierto, una atenuación muy importante.



A)

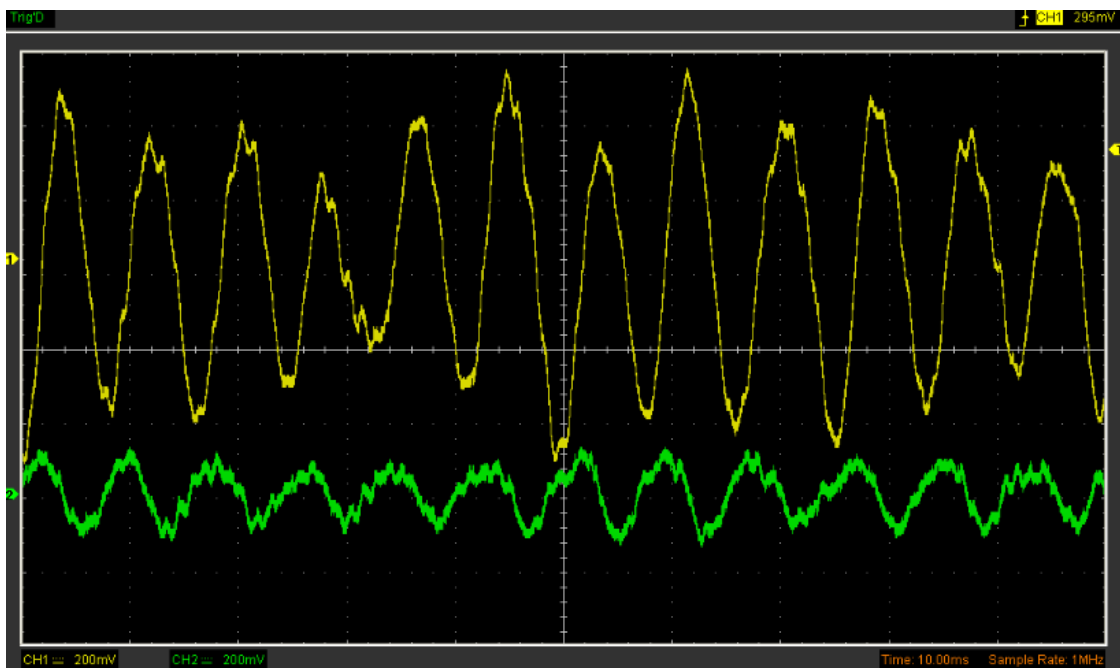


B)

**Figura 4.13. Resultados del algoritmo NOTCH. CH1 es la señal de error y CH2 la señal de control.**

## 4.9 ALGORITMO GENÉTICO

Se ha implementado una adaptación de un algoritmo genético, en el cual se escogió una población inicial aleatoria de 6 cromosomas (coeficientes de adaptación), de los cuales el que tuviera mejor rendimiento realizaba mutaciones (pequeños cambios aleatorios), de nuevo generando 6 diferentes coeficientes a partir de aquí los cuales son revisados nuevamente para escoger el que genere un error menor, realizando esto una y otra vez hasta que el algoritmo converja en una solución.



**Figura 4.14. Resultados del algoritmo genético. CH1 es la señal de error y CH2 la señal de control.**

La implementación de este algoritmo resultó en una convergencia un poco más lenta y una atenuación considerable, aunque menor que de los anteriores, aproximadamente 1.5db. La mejor atenuación se obtuvo colocando el amplificador en posición frontal a unos 50 cm de la estructura encerrada, ya que la geometría del compresor y la estructura no son similares a las del amplificador por lo que no se acoplan acústicamente.

## 5. CONCLUSIONES

El objetivo general de la presente monografía tenía como finalidad el diseño e implementación de un sistema de control activo de ruido para el compresor Marshal Indcutive Motor 931373. Se pensó en un sistema de bajo costo y simple. Para lograr tal fin, se partió con base en los objetivos parciales, los cuales enmarcan la caracterización de la señal de ruido del compresor, el desarrollo del sistema de control que atenúa la señal de ruido no deseada en un punto en el espacio y las pruebas objetivas que comprueban el funcionamiento de dicho sistema. El cumplimiento de estos objetivos representa las contribuciones aportadas por el presente trabajo y que se muestran a continuación.

Se logró obtener un desfase de ondas experimentalmente, en este caso usando el algoritmo FxLMS con un tiempo de muestreo de 16kHz. Aunque este caso no fue el más exitoso ya que el campo acústico afecta la atenuación y por lo tanto, éste no obtuvo el mayor nivel de atenuación experimentalmente.

Algunos sistemas de bajo costo, como transistores y micrófonos pueden tener gran incidencia en el comportamiento del sistema de control, si no se considera su respuesta real y las perturbaciones a las que son sometidos en los campos acústicos, para esto se debe tener en cuenta si el algoritmo que se implementa considera compensaciones a estos eventos externos.

Cuando se tiene acceso al equipo al que se desea realizar la atenuación de ruido, además de la energía que produce en el espectro de frecuencia, se pueden estudiar maneras alternativas que puedan dar un mejor rendimiento, como en este caso en donde se usó un algoritmo para atenuar la onda que estaba mayormente presente en el espectro (120Hz).

El paso de adaptación o factor de convergencia es una variable crucial a la hora de implementar la minoración del error con el algoritmo LMS. Este factor de convergencia, es la variable la cual hay que ajustar más detalladamente para obtener el mejor resultado en cada caso. En la documentación consultada aparece el cálculo de este parámetro teniendo en cuenta únicamente el número de muestras que se utilizan para la estimación, sin embargo, la

implementación de este algoritmo recursivo generó inestabilidad en el sistema, que se detectó mediante pruebas experimentales, por tal razón, se optó por la forma expuesta en el capítulo 3.1.1 realizando el cálculo de la media del aporte de energía de las entradas al cuadrado dividido la ventana de muestras tomadas para el cálculo.

Los algoritmos de banda ancha son poco efectivos en recintos abiertos, son más apropiados para ductos o estructuras donde se tenga que atenuar ondas planas con secciones onda menores, poco presentes en estructuras y motores industriales, como en este caso el compresor.

Los algoritmos adaptativos digitales son muy eficientes y responden bien a cambios sutiles y fuertes del sistema, además de un manejo eficiente abierto y de fácil manipulación ya que solo hay necesidad de cambiar código de programación en la tarjeta DSP para visualizar sus cambios.

Debido a que no hay una forma óptima para calcular la ubicación del sensor de error, se pueden tomar ubicaciones aleatorias y observar el comportamiento del sistema con base en la señal proveniente de este sensor, realizando cambios de posición e identificaciones de campo acústico por cada variación.

Se pueden utilizar algoritmos más complejos que el de pendiente negativa (LMS) de adaptación como los algoritmos genéticos usados en este caso, para obtener una reducción significativa en sistemas no lineales y en donde no se pueda realizar una identificación adecuada del campo acústico.



## BIBLIOGRAFÍA

1. Allie M.C., Bremigan C.D. and Eriksson L.J. (1998). "Hardware and Software Considerations for Active Noise Control", Proc. ICASSP 88, New York.
2. Beltrán, C.I. (1995) "Acoustic transducer nonlinearities and how they effect the performance of active noise control systems," Proc. of ACTIVE-95, Newport Beach, CA, USA.
3. Elliott S.J., Syothers I.M., Nelson P.A., McDonald A.M., Quinn D.C., and Saunders T. (1988). "The Active Control of Engine Noise Inside Cars", Proc. of Inter-Noise 88, pp. 987-990.
4. Flockton S.J. (1991). "Gradient-based Adaptive algorithms for systems with external feedback paths". IEE Proceedings-F, Vol. 138, No. 4.
5. FLORES MORALES, RAÚL. (2009). Supresión de ruido acústico en motores, pp 4-110. México D.F.
6. Fuller C.R. and Von Flotow A.H. (1995) "Active control of sound and vibration", IEEE Control Systems, pp. 9-19.
7. Lueg, Paul. (1936) "Process of silencing sound oscillations," U.S. Patent 2.043.416. Junio 19.
8. Minguez Olivares, Antonio. Ingeniería avanzada para sistemas de control de ruido acústico mediante técnicas adaptativas. <http://oa.upm.es/1184/>, Mayo 1998. [Artículo en línea]. Consultado el día 9 de abril de 2015 de la World Wide Web.
9. Pérez Vega, Constantino. (2009). Sonido y audición. Universidad de Cantabria. pp 4.
10. Oppenheim A.V. and Schafer R.W. (1989) Discrete-Time Signal Processing, Prentice-Hall, Englewood Cliffs, NJ.
11. Texas Instruments, training technical Organization. (2005) C6000 Integration Workshop. Dallas, Texas.
12. Widrow, B. Walach, E. (1996). Adaptive Inverse Control, Prentice Hall, EUA. Páginas 40 a 45.

## ANEXOS

### ANEXO 1

#### Código desarrollado con los diferentes algoritmos.

```
//ANC.c Programa para atenuación de ruido de compresor industrial
//Universidad Distrital
//Autores : Cristian Franco, Oscar Sánchez.

#include "DSK6416_AIC23.h" //librería para soporte del codec
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
Uint32 fs=DSK6416_AIC23_FREQ_24KHZ; //taza de muestreo
#define DSK6416_AIC23_INPUT_MIC 0x0015
#define DSK6416_AIC23_INPUT_LINE 0x0011
Uint16 inputsource=DSK6416_AIC23_INPUT_LINE; // entrada de la tarjeta

#define NOISELEVEL 8000 //factor para la secuencia de ruido blanco
#define BUFFER_SIZE 500 //tamaño de buffer de comunicación con interfaz

#define STATIC_BETA 1E-8;
#define WLENGTH 30 //tamaño del vector de coeficientes
#define ALFA 1/WLENGTH

#define RANDOM_NUM ((float)rand()/(RAND_MAX+1))

short noise=0;
short* sin_notch=0;
short* cos_notch=0;

//Características Generales del algoritmo
short var_beta = 1;
short var_beta_notch = 0;
float pot = 0;
float last_pot=0;
unsigned int ref_pot=0;
unsigned int error_pot=0;
unsigned int out_pot=0;
unsigned int freq;
unsigned int win_error_pot;
unsigned int win_pot;
unsigned int last_win_pot;
int change_freq=DSK6416_AIC23_FREQ_24KHZ-1;
float beta = STATIC_BETA; //factor de convergencia
float beta_b;
float beta_c;
short sineSize;
short main_counter=0;
short window_counter=0;
short ref_values[WLENGTH];
short error_values[WLENGTH];
short out_values[WLENGTH];
short restart = 0;
```

```

short ampFactor=1;
short gen_flag=1;
float* min_weigth=0;
int grad_error = 0;

float filt_error=0;

//id
float E;
short input_selector = 0;
int i = 0;

//short input_buffer[WLENGTH];
float sn[WLENGTH]; //vector de coeficientes camino secundario
float dn[WLENGTH];

short w2Length = 30;

//FXLMS
short secondary_path=0;
float xn=0;
short noise_add=0;
float w[WLENGTH]; //vector de coeficientes FIR
float c[WLENGTH];
float b[WLENGTH];
float yn_vector[WLENGTH];
float xfilt_vector[WLENGTH]; //vector de datos de entrada
float yfilt_vector[WLENGTH]; //vector de datos de salida

//notch
float notch_w1,notch_w2,wf1,wf2;
int sineindex = 0;
short table_size;
short freq_notch=480;
short last_freq=480;
float beta_notch = 3E-8;
short leaky=0;

//genético
float
gen_w[WLENGTH],gen_w2[WLENGTH],gen_w3[WLENGTH],gen_w4[WLENGTH],gen_w5[
WLENGTH],gen_w6[WLENGTH]; //buffer weights of adapt filter
float a_gen[6],amp_gen=0;
unsigned int error1=0,error2=0,error3=0,error4=0,error5=0,error6=0,sum_pot=0;
unsigned int geneindex = 0;
int prandp=0;

//Buffers

short ref_buffer[BUFFER_SIZE];
short error_buffer[BUFFER_SIZE];
short dif_buffer[BUFFER_SIZE];
short out_buffer[BUFFER_SIZE];
int buffer_index=0;
int j=0;

```

```

FILE *fptr;

//GUI
short selectIndex=0;

//variables de entrada y salida
short refsampl=0;
short errorsampl=0;
float yn=0;

typedef struct BITVAL //bits para generación de ruido blanco por algoritmo pseudo aleatorio
{
    unsigned int b0:1, b1:1, b2:1, b3:1, b4:1, b5:1, b6:1;
    unsigned int b7:1, b8:1, b9:1, b10:1, b11:1, b12:1, b13:1;
    unsigned int dweebie:2;
} bitval;

typedef union SHIFT_REG
{
    unsigned int regval;
    bitval bt;
} shift_reg;
int fb;
shift_reg sreg;
FILE *fptr; //file pointer

#pragma DATA_SECTION(ref_buffer, ".EXT_RAM") //buffer -> memoria externa de la tarjeta
#pragma DATA_SECTION(error_buffer, ".EXT_RAM") //buffer -> memoria externa de la tarjeta
#pragma DATA_SECTION(out_buffer, ".EXT_RAM") //buffer -> memoria externa de la tarjeta

void resetreg(void)
{
    sreg.regval=0xFFFF;
    fb = 1;
}

int prand(int level) //función de generación pseudo aleatoria de ruido blanco
{
    int prnseq;
    if(sreg.bt.b0)
        prnseq = -level; //scaled negative noise level
    else
        prnseq = level; //scaled positive noise level
    fb =(sreg.bt.b0)^(sreg.bt.b1); //XOR bits 0,1
    fb^=(sreg.bt.b11)^(sreg.bt.b13); //with bits 11,13 -> fb
    sreg.regval<<=1; //shift register 1 bit to left
    sreg.bt.b0=fb; //close feedback path
    return prnseq;
}

```

```

void generate_sins(){ //función de generación de ondas de una frecuencia
específica para algoritmo notch
    table_size = (short)(freq/freq_notch);
    float pi=3.14159;
    sin_notch = (short*)realloc (sin_notch,table_size*sizeof(short));
    cos_notch = (short*)realloc (cos_notch,table_size*sizeof(short));
    int i;
    for(i = 0; i < table_size; i++){
        sin_notch[i] = 1000*sin(2.0*pi*i/table_size);
    }

    for(i = 0; i < table_size; i++)
        cos_notch[i] = 1000*cos(2.0*pi*i/table_size);
}

void initializeVariables(){ //inicialización de vectores de algoritmos y de datos generales del
programa
    sreg.bt.b0 = 1;
    for (j = 0; j <= WLENGTH; j++)
    {
        w[j] = 0.0;
        c[j] = 0.0;
        b[j] = 0.0;
        sn[j] = 0.0;
        dn[j] = 0.0;
        ref_values[j]=0;
        xfilt_vector[j]=0;
        error_values[j]=0;
    }

    switch(fs){
    case 1:
        freq = 8000;
        break;
    case 2:
        freq = 16000;
        break;
    case 3:
        freq = 24000;
        break;
    case 4:
        freq = 32000;
        break;
    case 5:
        freq = 44000;
        break;
    case 6:
        freq = 48000;
        break;
    case 7:
        freq = 96000;
        break;
    }

    generate_sins();
}

```

```

srand((int)time(NULL));

for (i = 0; i < WLENGTH; i++){
    gen_w[i] = RANDOM_NUM/10;
    gen_w2[i] = RANDOM_NUM/10;
    gen_w3[i] = RANDOM_NUM/10;
    gen_w4[i] = RANDOM_NUM/10;
    gen_w5[i] = RANDOM_NUM/10;
    gen_w6[i] = RANDOM_NUM/10;
}

}

void setGenWeights(float* weigth){           //generación de datos evolutivos de
coeficientes de algoritmo genético
    for (i = 0; i < WLENGTH; i++){
        gen_w[i] = weigth[i] + al_gen[0]*amp_gen;
        gen_w2[i] = weigth[i] + al_gen[1]*amp_gen;
        gen_w3[i] = weigth[i] + al_gen[2]*amp_gen;
        gen_w4[i] = weigth[i] + al_gen[3]*amp_gen;
        gen_w5[i] = weigth[i] + al_gen[4]*amp_gen;
        gen_w6[i] = weigth[i] + al_gen[5]*amp_gen;
    }
}

unsigned int minValue(unsigned int a, unsigned int b){
    unsigned int min;
    if(a < b){
        min=a;
    }else{
        min=b;
    }
    return min;
}

//Algoritmos de atenuación
short id(short ref,short error)
{

    float E=0;
    short i;
    float adaptfir_out = 0;

    if(noise){
        prandp = prand(NOISELEVEL);
        output_left_sample(prandp);           //salida de ruido blanco
    }else{
        output_left_sample(0);
    }

    if(!input_selector){                     //control de id de camino secundario
        for (i = 0; i < WLENGTH; i++){
            adaptfir_out +=(sn[i]*error_values[i]);
            E = error - adaptfir_out; //señal de error
        }
    }
}

```

```

        for (i = 0; i < WLENGTH; i++)
            sn[i] = sn[i]+(beta*E*error_values[i]); //actualizació de
coeficientes por LMS
    }else if(input_selector){ //control de id de realimentación
        for (i = 0; i < WLENGTH; i++)
            adaptfir_out +=(dn[i]*ref_values[i]);
        E = ref - adaptfir_out; //señal de error

        for (i = 0; i < WLENGTH; i++)
            dn[i] = dn[i]+(beta*E*ref_values[i]); //actualizació de coeficientes por
LMS
    }

    return (short)E;
}

short fxlms(short ref,short error, float* wfx)
{
    short i = 0;
    float output = 0;
    float xn_filt=0;
    float leak_factor = 1;
    pot = 0;
    prandp = 0;

    if(secondary_path){
        for (i = 0; i < WLENGTH; i++){
            xn += (dn[i] * out_values[i]); //calculo de salida filtrada por camino
secundario
        }
        ref= ref-xn;
        ref_values[0] = ref;
    }

    for (i = 0; i < WLENGTH; i++){
        output += (wfx[i] * ref_values[i]); //calculo de salida con los pesos adaptativos
        xn_filt += (sn[i] * ref_values[i]);
    }

    output_left_sample((short)output); //salida a parlante
    if(noise_add){
        prandp = prand(8000);
    }
    xfilt_vector[0] = xn_filt + prandp;

    for (i = 0; i < WLENGTH; i++){
        pot += (xfilt_vector[i]*xfilt_vector[i]);
    }

    if(var_beta){
        beta = ((0.01)/(5+(WLENGTH+20)*pot)); //calculo de beta variable
    }
}

```

```

        if(leaky){
            leak_factor = 1-(beta*100);           //calculo de factor leaky
        }

        for (i = 0; i < WLENGTH; i++){
            wfx[i] = leak_factor*wfx[i] - beta*error*xfilt_vector[i]; //actualización de
coeficientes por LMS
        }

        for (i = WLENGTH-1; i > 0; i--) //actualización de vector de entradas filtradas
        {
            xfilt_vector[i]=xfilt_vector[i-1];
        }

        last_pot=pot;

return (short)output;
}

short notch(short error)
{
    if(freq_notch != last_freq){
        generate_sins();           //generación de ondas sinusoidales a cancelar
        last_freq = freq_notch;
    }
    short ref_sample_sin= sin_notch[sineindex];
    short ref_sample_cos= cos_notch[sineindex];
    if (sineindex >= table_size-1) sineindex = 0;
    sineindex++;

    refsampl= sin_notch[sineindex];
    errorsampl= cos_notch[sineindex];

    yn = 0; //init output of adapt filter

    yn = _sadd(notch_w1*ref_sample_sin, notch_w2*ref_sample_cos);

    output_left_sample((short)yn); //salida al sistema

    if(var_beta_notch){
        beta_notch = (0.001/(5+(WLENGTH+5)*error_pot)); //Cálculo de factor de
convergencia
    }

    notch_w1 = notch_w1 - beta_notch*ref_sample_sin*error; //actualización de
coeficientes de algoritmo
    notch_w2 = notch_w2 - beta_notch*ref_sample_cos*error;

return (short)yn;
}

short genetico(short ref,short error)

```



```

{
    short i,error_flag=0;
    unsigned int min_def=0;
    yn=0;

    if(last_win_pot != 0){
        if(win_pot > last_win_pot ){
            gen_flag=1;
            setGenWeighths(min_weigth); //Generar nuevos
vectores evolutivos candidatos
        }
    }

    if(gen_flag){
        if (geneindex <= WLENGTH-1){
            for (i = 0; i < WLENGTH; i++){
                yn += (gen_w[i] * ref_values[i]); //cálculo de
salida del sistema
            }
            if(geneindex >= (WLENGTH/2)){
                error1+=error*error;
            }
        }
        else if (geneindex <= 2*WLENGTH-1){
            for (i = 0; i < WLENGTH; i++){
                yn += (gen_w2[i] * ref_values[i]); //cálculo de
salida del sistema
            }
            if(geneindex >= ((3*WLENGTH)/2)){
                error2+=error*error;
            }
        }
        else if (geneindex <= 3*WLENGTH-1){
            for (i = 0; i < WLENGTH; i++){
                yn += (gen_w3[i] * ref_values[i]); //cálculo de
salida del sistema
            }
            if(geneindex >= ((5*WLENGTH)/2)){
                error3+=error*error;
            }
        }
        else if (geneindex <= 4*WLENGTH-1){
            for (i = 0; i < WLENGTH; i++){
                yn += (gen_w4[i] * ref_values[i]); //cálculo de
salida del sistema
            }
            if(geneindex >= ((7*WLENGTH)/2)){
                error4+=error*error;
            }
        }
        else if (geneindex <= 5*WLENGTH-1){
            for (i = 0; i < WLENGTH; i++){
                yn += (gen_w5[i] * ref_values[i]); //cálculo de
salida del sistema
            }
        }
    }
}
    
```

```

        if(geneindex >= ((9*WLENGTH)/2)){
            error5+=error*error;
        }
    }
    else if (geneindex <= 6*WLENGTH-1){
        for (i = 0; i < WLENGTH; i++){
            yn += (gen_w6[i] * ref_values[i]); //cálculo de
salida del sistema

        }
        if(geneindex >= ((11*WLENGTH)/2)){
            error6+=error*error;
        }
    }

    }

    output_left_sample((short)yn);

    geneindex++;
    if (geneindex > 6*WLENGTH-1){
        geneindex=0;
        error_flag=1;
    }

    if(error_flag){
        unsigned int min1,min2,min3;
        short j,sign;
        float rnd=0;

        for(j=0;j<6;j++){

            rnd= RANDOM_NUM;

            if(rnd > 0.5){
                sign = 1;
            }else{
                sign = -1;
            }
            al_gen[j] = RANDOM_NUM*sign;
        }

        min1=minValue(error1,error2);
        min2=minValue(error3,error4);
        min3=minValue(error5,error6);
        min_def= minValue(min1, minValue(min2,min3));

//vector con error cuadrático mínimo

        float prueba =
(0.1*ampFactor)/(5+(WLENGTH*error_pot));
        amp_gen = prueba;
        if(min_def == error1){
            min_weigth = &gen_w[0];
        }else if(min_def == error2){
            min_weigth = &gen_w2[0];

```

```

        }else if(min_def == error3){
            min_weigth = &gen_w3[0];
        }else if(min_def == error4){
            min_weigth = &gen_w4[0];
        }else if(min_def == error5){
            min_weigth = &gen_w5[0];
        }else if(min_def == error6){
            min_weigth = &gen_w6[0];
        }
    }

    error1=0;
    error2=0;
    error3=0;
    error4=0;
    error5=0;
    error6=0;
    gen_flag=0;
}
}
else{
    yn = fxlms(ref,error,min_weigth); //adadtación por fxlms si
no hay mínimo local
}

return yn;
}

```

```

void updateVectors(){ //Actualización de coeficientes de vectores
    for (i = WLENGTH-1; i > 0; i--)
    {
        error_values[i] = error_values[i-1];
        ref_values[i] = ref_values[i-1];
        out_values[i] = out_values[i-1];
    }

    return;
}

```

```

interrupt void c_int11() //rutina de interrupción por ADC de la tarjeta.
{
    unsigned int sum_ref_pot=0;
    unsigned int sum_error_pot=0;
    unsigned int win_error_pot=0;

    if(restart || (change_freq+1!=fs)){
        fs = change_freq+1;
        initializeVariables();
        restart = 0;
        set_freq();
    }

    if(main_counter <= WLENGTH){
        main_counter++;
    }
}

```

```

    }

    if(window_counter >= WLENGTH){
        last_win_pot = win_pot;
        window_counter = 0;
        for (i = 0; i < WLENGTH; i++){
            win_error_pot += (error_values[i]*error_values[i]);
        }
        win_pot= win_error_pot / WLENGTH;
    }
    else{
        window_counter++;
    }

    refsamplereferencia=input_left_sample(); //valor de entrada de micrófono de
    errorsamplereferencia= input_right_sample(); //valor de entrada de micrófono de error

    ref_values[0]= refsamplereferencia;
    error_values[0]= errorsamplereferencia;

    if(main_counter > WLENGTH){ //Cálculo de potencias
        for (i = 0; i < WLENGTH; i++){
            sum_ref_pot += (ref_values[i]*ref_values[i]);
            sum_error_pot += (error_values[i]*error_values[i]);
        }
        ref_pot = sqrt(sum_ref_pot/WLENGTH);
        error_pot = sqrt(sum_error_pot/WLENGTH);
        grad_error= (error_values[0] - error_values[WLENGTH-1])/WLENGTH;
    }

    switch (selectIndex){ //Selección de algoritmo por interfaz
        case 0:
            yn = id(refsamplereferencia,errorsamplereferencia);
            break;
        case 1:
            yn = fxlms(refsamplereferencia,errorsamplereferencia,&w[0]);
            break;
        case 2:
            yn = notch(errorsamplereferencia);
            break;
        case 3:
            yn = genetico(refsamplereferencia,errorsamplereferencia);
            break;
        case 4:
            yn = fulms(refsamplereferencia,errorsamplereferencia);
            break;
    }

    out_values[0]=yn;
    unsigned int sum_out_pot=0;

    if(main_counter > WLENGTH){
        for (i = 0; i < WLENGTH; i++){

```

```

        sum_out_pot += (out_values[i]*out_values[i]);
    }
    out_pot = sqrt(sum_out_pot/WLENGTH);
}

ref_buffer[buffer_index] = refsample;
error_buffer[buffer_index] = errorsample;
out_buffer[buffer_index++] = yn;
if (buffer_index >= BUFFER_SIZE) buffer_index = 0; //reinicializar buffer de datos

updateVectors();
return;
}

void main()                //main
{
    initializeVariables();
    comm_intr();           //inicializar DSK, codec, McBSP

    while(1){
        //loop infinito
    }
    free(sin_notch);
    free(cos_notch);
}

```

## ANEXO 2

### SONÓMETRO DIGITAL SOUND LEVEL METER. REFERENCIA S01.

Caracterizado con un amplio rango de medición (30 y 130 a 130 dBA/35 DBC), indicación de gráfico de barras, una ponderación de frecuencia de respuesta rápida / lenta y función de retención máxima.

También proporciona un parabrisas para filtrar la Salida señales no deseadas, pantalla iluminada, indicador de batería baja y diseño de apagado automático para ahorro de energía.

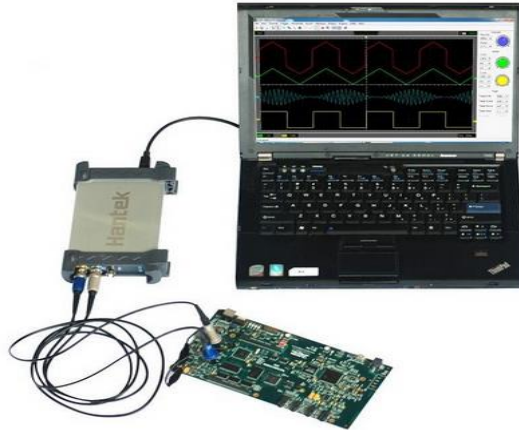


1.

<b>Especificaciones</b>	<b>Sonómetro S01</b>
Rango de medida	30 a 130 dBA/35 a 130 DBC
Precisión	+/-2.0dB
Rango de frecuencia	31.5Hz a 8.5Khz
Resolución	0.1dB
Característica de frecuencia de ponderación	A
Time Weighting	2 veces / segundo
Dimensiones	57x26x149mm

## ANEXO 3

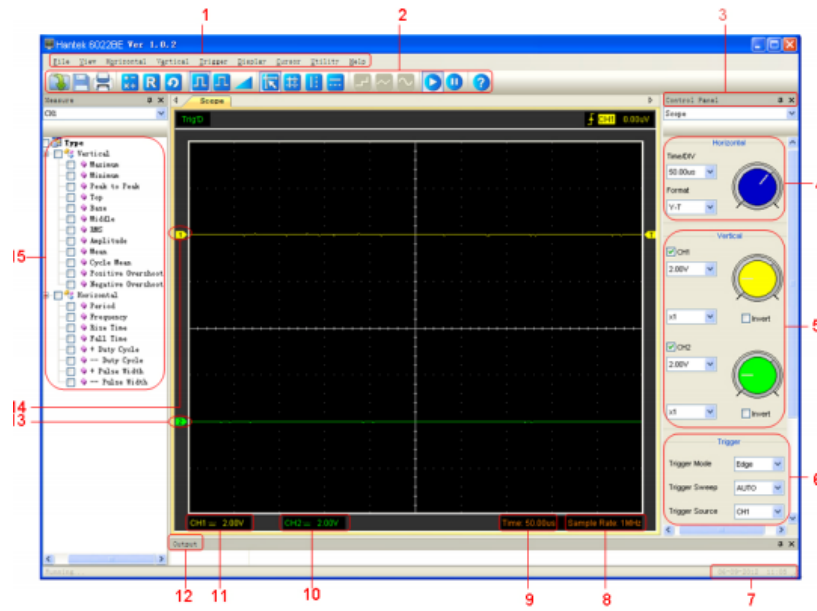
### OSCILOSCOPIO HANTEK 6022BE



El osciloscopio no requiere una fuente externa de poder, es portable y se conecta por medio de puerto USB.

Características generales.

- Tasa de muestreo en tiempo real máxima: 48MSa/s.
- Tamaño de memoria: 1Mbyte/channel
- Capacidad de la función Fast Fourier Transform (FFT);
- 20 medidas automáticas.
- Medidas de control del cursor automático.
- Almacenamiento de forma de onda.
- Selección de calibración rápida.
- Suma, resta, multiplicación y división de funciones matemáticas.
- Intensidad de forma de onda ajustable.
- Lenguaje seleccionable por el usuario.



1. Menú principal.
2. Toolbar.
3. Panel de control.
4. Panel horizontal.
5. Panel vertical.
6. Panel Trigger.
7. Ventana de tiempo.
8. Ventana de tasa de muestreo.
9. Tiempo principal y base de configuración.
10. Información del canal 2.
11. Información del canal 1.
12. Ventana de salida.
13. Los marcadores muestran los puntos de referencia en las formas de onda mostradas.  
Si no hay marca el canal no es mostrado.
14. La misma marca que el punto 13.
15. Ventana de medida.



## ANEXO 4

### FOTOGRAFÍAS DE LAS MEDICIONES DE RUIDO DEL COMPRESOR CON ESCTRUCTURA Y SIN ESCTRUCTURA



