

IMPLEMENTACIÓN DEL ALGORITMO LMS EN LA PLATAFORMA DSPIC PARA LA ELIMINACIÓN DE RUIDO ACÚSTICO SINUSOIDAL

LMS ALGORITHM IMPLEMENTATION ON THE PLATFORM DSPIC FOR THE REMOVAL OF ACOUSTIC NOISE SINUSOIDAL



AUTOR

MARCO FIDEL FLÓREZ FRANCO
Maestría en Ingeniería (c)-UPB
Especialista en Telecomunicaciones-UPB
Ingeniero Electricista -UIS
Unidades Tecnológicas de Santander
Docente Investigador- Grupo CEAC
Ingeniería Electrónica
Marcofidel_f@yahoo.com
COLOMBIA

INSTITUCIÓN

Unidades Tecnológicas de Santander
UTS
Institución tecnológica redefinida
Calle de los estudiante 9-82 barrio
Real de minas
Bucaramanga
uts.uts@edu.co
COLOMBIA

Recepción: Junio 10 de 2009

Aceptación: Septiembre 4 de 2009

Temática: Procesamiento de Señales, Electrónica y Modelado de Sistemas Físicos.

Artículo Tipo: Artículo de Investigación Científica y Tecnológica

RESUMEN

En este artículo se presentan los resultados de implementar el algoritmo de filtrado adaptativo LMS (Least-Mean-Square) para la eliminación de ruido acústico sinusoidal, en una plataforma de audio con base un controlador digital de señales (DSC), DSPIC33FJ256GP710. El objetivo general de este trabajo es comprobar la capacidad del DSC en el filtrado adaptativo señales de audio en tiempo real, el cual puede extrapolarse a otro tipo de señales en el mismo rango de frecuencia. El algoritmo fue implementado en lenguaje C empleando el compilador C30, con una estructura tipo FIR y dos tipos de datos, punto flotante y punto fijo, la evaluación de los resultados fue realizada utilizando MATLAB, para lo cual se desarrollo un script que permite la reproducción de señales de audio con la adición de ruido sinusoidal, evaluando las señales antes y después del filtrado adaptativo. Los resultados fueron satisfactorios usando el algoritmo desarrollado en punto fijo, el cual aprovecha la arquitectura del DSC para la ejecución de aritmética fraccional; dejando como conclusión que el dispositivo cumple con las expectativas para el tratamiento en tiempo real de este tipo de señales.

PALABRAS CLAVES

DSPIC
Ruido Acústico
Filtrado adaptativo
Algoritmo LMS

ABSTRACT

This article presents the results of implementation of the adaptive filtering algorithm LMS (least-mean square) for the elimination of the acoustic noise is removal, on a platform based on the audio in the digital controller (DSC), DSPIC3FJ256GP710. The purpose of the general of this work is to test the performance of DSC to a be adaptive audio filters with the signal audio in the real time, which can be extrapolated to the signal on the same frequency range. The algorithm was worked inc using the c30 compiler, with a structure with a FIR (Finite Impulse Response) and two types of data, floating point and fixed point, the evaluation of the results was performed using a Matlab, and is developing a scrip that will allow to playback the audio signals with the addition of the sinusoidal noise, evaluating the signal before and after adapting the filter. The results were satisfactory, using the algorithm in point fixed, which was uses the DSC and the architecture for the implementing arithmetic fractional. This leads to the conclusion that the device did the correct function with the expectations for the treatment in the real time with its type of signal.

KEYWORDS

DSPIC
Acoustic Noise
Adaptive filters
LMS algorithm

INTRODUCCION

En los sistemas de adquisición de señales es muy común que se presente ruido de tipo sinusoidal debido a que los sistemas de alimentación eléctrica utilizan este tipo de señales, generando ruido relacionados con sus armónicas; este ruido altera la forma de las señales y se convierte en un problema de gran relevancia si se trata de señales biomédicas, cuyas frecuencias son cercanas a estos valores [1]. Otro problema similar se presenta en las señales de audio que están próximas a maquinas rotativas que producen un ruido periódico acotado en frecuencia generando interferencia en las comunicaciones [2].

Esta investigación evalúa la plataforma DSPIC como controlador digital de señales, con el fin de eliminar el ruido acústico en tiempo real, mejorando la calidad

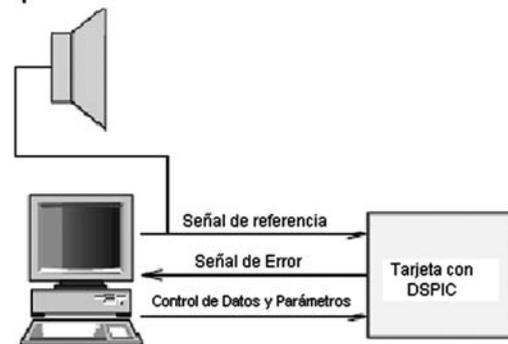
de las señales , y está inmerso dentro del proyecto de investigación titulado, "Implementación de filtros adaptativos bajo la tecnología DSPIC para el Procesamiento de señales ECG." el cual actualmente se adelanta en el grupo de conversión de energía, automática y control (CEAC), de la Unidades Tecnológicas De Santander, este proyecto tiene como finalidad el desarrollo de equipos para la adquisición, tratamiento y visualización de señales ECG.

El documento cuenta con cuatro secciones: En la primera de ellas realiza una explicación de los equipos y método aplicado, en la segunda se realiza una recopilación teórica a los conceptos relacionados con el filtrado digital clásico y adaptativo, en la tercera sección se detalla el programa implementado y en la última sección mostrará los resultados y su respectivo análisis.

1. PLATAFORMA DE DESARROLLO.

Para la implementación del algoritmo LMS, se utilizo una tarjeta de audio con base en el DSPIC3fj256gp710 y un CODEC de audio SI3000 de 16 bits, un micrófono para PC, una computadora portátil marca TOSHIBA, Las pruebas se realizaron en el laboratorio de máquinas eléctricas de la Unidades Tecnológicas de Santander a una temperatura ambiente promedio de 29 grados Celsius, interconectando la tarjeta de sonido del Laptop con la tarjeta de audio con el DSP como se muestra en la figura 1, aplicando una frecuencia de muestreo de 10000 muestras/segundo y almacenando el resultado en formato WAV (Windows Audio/Video). Con duración de 60 segundos.

Figura 1. Esquema del Experimento [3]

Fuente primaria de Ruido

A continuación se describe las características técnicas del equipo utilizado. El sensor utilizado es un micrófono para computadora, marca BENWIN con referencia MP008 con las siguientes especificaciones:

- Patrón de direcciones Omnidireccional.
- Impedancia de salida: 1.4 KΩ.
- Frecuencia de respuesta: 50 a 16000 Hz.
- Sensibilidad: -58dB ± 2dB
- Voltaje de operación: 1V
- Consumo ordinario: 350µA máx.
- Proporción S/N: 40 dB .
- Largo del cable: 1.5 Metros.

Para la adquisición de las muestras de audio se usó una computadora portátil TOSHIBA SATELLITE M35XS349, con las siguientes características:

- DVD_RW / CD-RW.
- Capacidad de Disco 80 GB.
- Procesador Pentium M.
- Microsoft Windows XP.
- Memoria RAM 512 MB.
- Velocidad 1.7 GHz.

1.1 Dspic33FJ256GP710

Para desarrollo de este artículo, se utilizó el controlador digital de señales dspic33fj256gp710 [4] de 100 pines perteneciente a la familia dsPIC33FJXXXGPX06/X08/X10, de MICROCHIP que presenta un módulo especializado para la comunicación con el CODEC de audio llamado Interfaz del Conversor de Datos (DCI), este módulo está en la capacidad de recibir datos directamente del SI3000 y puede funcionar en modo maestro generando las señales de reloj y de control; o esclavo recibiendo los datos de audio por interrupción. Otra ventaja de este DSPIC es su memoria de programa de 256 KB y la memoria de datos con 32 KB [4], adicionalmente posee un módulo DMA que permite el acceso directo a la memoria liberando al DSP de la tarea de recibir los datos provenientes del conversor, aumentando la velocidad en el desarrollo de los procesos. Los controladores DSPIC tienen una arquitectura tipo Harvard, diseñada con un motor de búsqueda DSP capaz de realizar operaciones matemáticas y de búsqueda en memoria en un solo ciclo de máquina. Las operaciones DSP se realizan con acumuladores de 40 bits que emplean la aritmética fraccional para trabajar los datos de 16 Bits, en esta aritmética toma los datos binarios en complemento a dos con magnitudes -1 y 1 permite la aritmética entera disminuyendo la complejidad de las operaciones, aumentando la velocidad de proceso, pero introduce errores de cuantización y desbordamiento. Para esta investigación se implementaron los programas en punto flotante y punto fijo para determinar el rendimiento del procesador y la calidad de los resultados.

En el mercado existen varios fabricantes de procesadores digitales de señales como Texas Instruments, Motorola, Analog devices, Lucent technologies, con capacidades

de cómputo superiores al DSPIC, pero sus plataformas de desarrollo es compleja y costosa comparativamente habiendo con la plataforma simple y accesible presentada por MICROCHIP, que adicionalmente permite usar libremente su software de desarrollo, siendo una gran ventaja para fabricar equipos de bajo costo.

2. FILTRADO DIGITAL

En esencia un filtro adaptativo realiza la misma operación de filtrado que un filtro clásico; la diferencia radica en el algoritmo de adaptación que permite modificar los coeficientes de acorde con los cambios en la señal de entrada. A continuación se describen las características fundamentales de los tipos de filtrado el clásico y adaptativo.

2.1 FILTRADO CLÁSICO

Un filtro digital lleva a cabo una modificación del espectro de la señal de entrada de acorde con su función de transferencia (pasa bajas, pasa altas, etc.) y el orden de los coeficientes implementados. Para tal efecto los dos filtros más utilizados en la cancelación activa de ruido son: el filtro FIR (Finite Impulse Response) y el filtro IIR (Infinite Impulse Response).

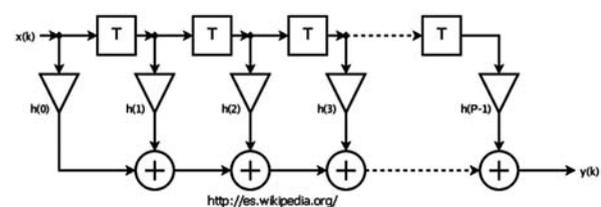
2.1.1 Filtros FIR

Estos filtros se caracterizan por tener una respuesta impulsional $h(n)$ de longitud M finita y con frecuencia se supone que son causales [5]. Para obtener la salida del filtro se utilizan las muestras actuales y las anteriores. Su expresión matemática es la siguiente:

$$y(n) = \sum_{k=0}^M c(k)x(n-k) \quad (1)$$

Y su estructura descrita con ecuaciones en diferencias constantes se muestra en la figura 2 que se muestra a continuación.

FIGURA 2. Estructura básica de un filtro FIR



Los filtros FIR tienen la gran ventaja de que pueden diseñarse para ser de fase lineal, lo cual hace que presenten ciertas propiedades en la simetría de los coeficientes. Este tipo de filtros tiene especial interés en aplicaciones de audio, además son siempre estables.

Como defecto tienen la desventaja de necesitar un orden mayor respecto a los filtros IIR para cumplir las mismas características. Esto se traduce en un mayor gasto computacional [6].

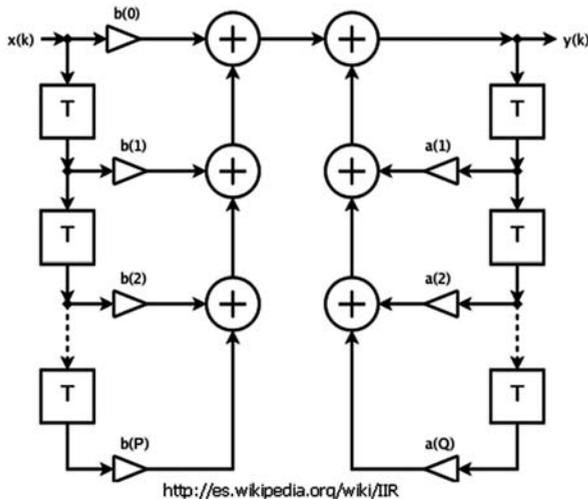
2.1.2 Filtros IIR

Un filtro IIR es un caso particular de sistema de respuesta impulsional infinita, para lo tanto para implementar un filtro IIR teóricamente se necesita una $h(n)$ con infinitos valores [6], en la práctica el número de coeficientes es limitado. La expresión matemática de un filtro IIR es la siguiente.

$$y(n) = \sum_{i=0}^M b_i x(n-k) - \sum_{k=1}^N a_k y(n-k) \quad (2)$$

Su estructura descrita con ecuaciones en diferencias constantes se muestra en la figura 3.

FIGURA 3. Estructura básica de un filtro IIR



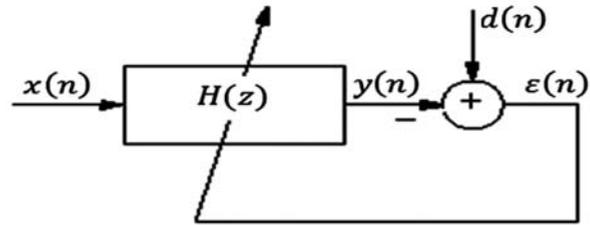
Las principales diferencias respecto a los filtros FIR es que los IIR pueden cumplir las mismas exigencias que los anteriores pero con menos orden de filtro. Esto es importante a la hora de implementar el filtro, pues presenta una menor carga computacional. Este tipo de filtros puede ser inestable, aún cuando se diseñen para ser estables. En principio no pueden diseñarse para tener fase lineal pero se pueden aplicar algunas técnicas como el filtrado bidireccional para lograrlo [7].

2.2 FILTRADO ADAPTATIVO

La idea del filtrado adaptativo nace de la aleatoriedad de las señales de ruido y por lo tanto se necesita un sistema cuyos coeficientes se adapten a los cambios en la señal de entrada. [8] Para ello se pretende la respuesta óptima

actualizando los coeficientes, buscando alcanzar el mínimo de una función de costo. En la figura 4 se ilustra la estructura de un filtro adaptativo, Donde $x(n)$ es la señal de referencia que se toma para actualizar para correlacionar los coeficientes con la señal de entrada $d(n)$ y $e(n)$ es la salida de error del filtro. El orden del filtro dependería del número de retardos de la anchura de la ventana en la serie temporal

FIGURA 4. Estructura de un filtro Adaptativo



2.2.1 Algoritmo FIR LMS [8]

El algoritmo (LMS), presentado por Widrow y Hoff en 1959 es un algoritmo de adaptación, que utiliza un método basado según la dirección opuesta del gradiente de la superficie del error. LMS utiliza las estimaciones de la pendiente del vector de los datos disponibles. LMS incorpora un procedimiento iterativo que hace que las sucesivas correcciones que se introduzcan en el vector de peso, sean en la dirección del gradiente negativo, que finalmente conduce al mínimo error cuadrático medio [9]. En comparación con otros algoritmos LMS algoritmo es relativamente sencillo, no requiere la función de correlación de cálculo, ni requiere invertir una matriz [10]. La señal de error obedece a la ecuación 3:

$$e(n) = d(n) - y(n) \quad (3)$$

Combinando la señal del error con la salida de un filtro FIR se Obtiene:

$$e(n) = d(n) - \sum_{k=0}^M c(k)x(n-k) \quad (4)$$

Bajo estas condiciones el error cuadrático instantáneo Será:

$$e(n)^2 = [d(n) - \sum_{k=0}^M c(k)x(n-k)]^2 \quad (5)$$

El eje principal del algoritmo es que los coeficientes $c(k)$ se actualicen de forma iterativa por el método de la máxima Pendiente los nuevos coeficientes se hallan con base en la ecuación 6 [11], esta idea se ilustra de manera grafica en la figura 5.

$$c(k)_{n+1} = c(k)_n - \mu \nabla_{c(k)} e(n)^2 \quad (6)$$

Es decir, los coeficientes en la iteración siguiente se calculan a partir de los valores de la iteración actual, menos el gradiente del error cuadrático instantáneo, $e(n)^2$ multiplicado por una constante μ . El gradiente o la derivada del error cuadrático instantáneo, con respecto a cada uno de los coeficientes, tiene la siguiente expresión [9]:

$$\nabla_{c(k)} e(n)^2 = 2e(n)\nabla_{c(k)} e(n) = 2e(n) \frac{\partial e(n)}{\partial c(k)} \quad (7)$$

$$\frac{\partial e(n)}{\partial c(k)} = \frac{\partial}{\partial c(k)} \left[d(n) - \sum_{k=0}^M c(k)x(n-k) \right] = -x(n-k) \quad (8)$$

La expresión final de actualización de los coeficientes según el algoritmo LMS:

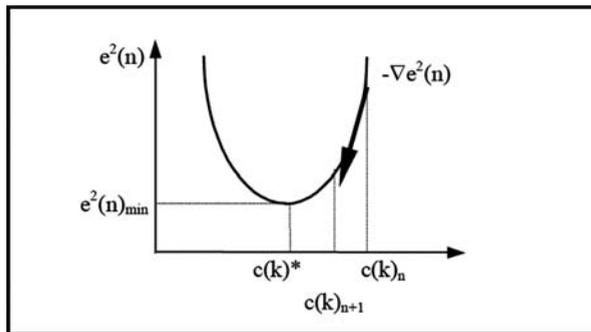
$$c(k)_{n+1} = c(k)_n + 2\mu e(n)x(n-k) \quad (9)$$

El parámetro μ es el paso de adaptación, y gobierna la velocidad de convergencia del algoritmo y el desajuste final del error cuadrático con respecto a su valor mínimo [9]. De forma que, cuando se escoge un factor grande se llega rápidamente a las inmediaciones del error cuadrático mínimo, oscilando alrededor del mismo una determinada magnitud. Sin embargo, si se elige un paso de adaptación pequeño la velocidad de convergencia hacia el valor mínimo del error cuadrático es lenta aunque el desajuste final es pequeño [8]. Los límites en los que se mueve el paso de adaptación son:

$$0 < \mu < \frac{1}{(M+1)\sigma^2} \quad (10)$$

Donde $M+1$ es el número de coeficientes del filtro FIR y σ^2 es la potencia de la señal de entrada $x(n)$. Si el valor del paso de adaptación está fuera de estos límites el algoritmo diverge y no es capaz de encontrar los valores de los coeficientes que hacen mínimo el error cuadrático instantáneo [8].

FIGURA 5. Gradiente de la superficie del error cuadrático instantáneo [2].



2.2.2 Algoritmo IIR LMS.

la ecuación 2 muestra la estructura de un filtro IIR, al asociarla con la señal del error se obtiene:

$$e(n) = d(n) - \sum_{k=0}^M c(k)x(n-k) - \sum_{k=1}^N b(k)y(n-k) \quad (11)$$

Bajo estas condiciones el error cuadrático instantáneo Será:

$$e(n)^2 = [d(n) - \sum_{k=0}^M c(k)x(n-k) - \sum_{k=1}^N b(k)y(n-k)]^2 \quad (12)$$

Actualizando los coeficientes de forma iterativa por el método de la máxima pendiente descrito en la sección anterior se obtiene se obtiene:

$$c(k)_{n+1} = c(k)_n + 2\mu_c x(n-k)e(n) \quad (13)$$

$$b(k)_{n+1} = b(k)_n + 2\mu_b x(n-k)e(n) \quad (14)$$

Donde μ_c , μ_b son el paso de adaptación para los coeficientes $c(k)$ y $b(k)$.

La implementación de filtro adaptativo IIR LMS, presenta mayores inconvenientes, ya que pueden ser inestables cuando se realizan las iteraciones para buscar los coeficientes óptimos, porque los polos del sistema salen de la circunferencia unitaria en el plano Z, adicionalmente la forma de la superficie del error presenta muchos mínimos, creando una convergencia incorrecta en un mínimo local [9], es por este motivo que se decidió implementar un filtro FIR para la realización del presente trabajo.

2.3 TRABAJOS ANTERIORES RELACIONADOS CON EL FILTRADO ADAPTATIVO DE SEÑALES.

Desde que se creó el algoritmo LMS en los años sesenta ha sido utilizado en diferentes aplicaciones para la eliminación de ruido, hoy en día sigue vigente la implementación de nuevos algoritmos que lo utilizan como base. Durante la última década se ha evidenciado la presentación de trabajos de investigación en publicaciones de carácter científico entre las cuales se pueden destacar las siguientes:

En año de 2005 se publicó el artículo titulado, "comparación de algoritmos feed-forward adaptativos Para el control activo de ruido en un conducto", realizado por el laboratorio sistemas de control avanzado de la de Universidad politécnica de Cataluña, el artículo describe una aplicación del control activo del ruido variando el núcleo del controlador entre el algoritmo de mínimos cuadrados recursivos (RLS) y algoritmo LMS, mostrando que se obtienen mejores resultados con el algoritmo RLS pero a un elevado costo

computacional. En julio del 2005 se presentó la tesis titulada “**Variantes del algoritmo LMS. Aplicación a un sistema cancelador de ecos**”, de la escuela Universitaria Politécnica De Teruel, el propósito de de esta tesis es usar las diferentes variantes del algoritmo LMS para cancelar el eco producido en los sistemas de telecomunicaciones, mostrando como resultados que para este tipo de aplicación el algoritmo LMS clásico presentó la mejor relación entre la atenuación del eco y el costo computacional. En el año 2007 la revista Chilena de ingeniería publicó el artículo “**Proposición Y Simulación De Un Algoritmo Adaptativo Para Sistemas De Antenas Inteligentes**”, en este trabajo se compara los algoritmos LMS y RLS, con un algoritmo híbrido desarrollado por los autores que combina la sencillez y velocidad de convergencia de LMS con la robustez del RLS, probando los resultados usando la plataforma MATLAB. En el año 2008 fue presentado el trabajo de investigación, “**Algoritmo de Mínimos Cuadrados con Error Codificado para Filtrado Adaptivo**”, en la revista Información Tecnológica, en el se presenta una modificación al algoritmo con el fin de reducir su complejidad, la mejora presentada se basa en cambiar la codificación del algoritmo, logrando reducir el número de operaciones en punto flotante, aumentando la velocidad de convergencia del mismo.

Los trabajos que se encontraron a la fecha son soportados por plataformas de cómputo robustas, como MATLAB o procesadores digitales de señales nativos de punto flotante de Texas Instrument, lo que hace más complicado y costoso su implementación en circuitos embebidos, aquí radica la importancia de este tipo de trabajos, que permiten la implementación de equipos de bajo costo y complejidad que puedan ser aplicados en la solución de problemas prácticos.

3. IMPLEMENTACIÓN DEL ALGORITMO.

Para evaluar el filtrado adaptativo en la plataforma DSPIC se implementaron dos algoritmos, uno en MATLAB que permite analizar las componentes espectrales de una señal de audio y un programa en lenguaje con el compilador C30 para el DSPIC, estos algoritmos se describen a continuación:

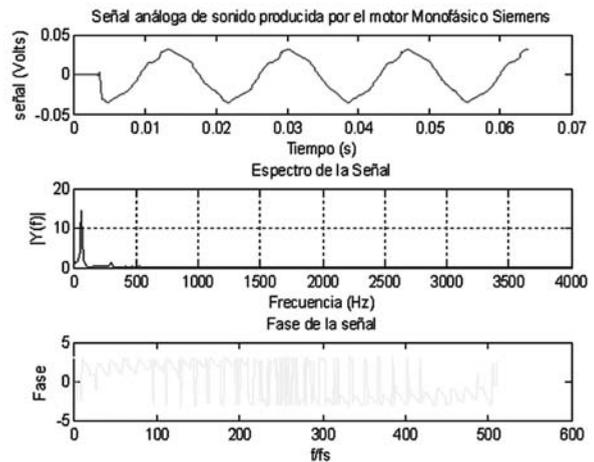
3.1 ALGORITMO EN MATLAB

En esta investigación se ha desarrollado un programa que permite tomar una señal de audio y evaluar las diferentes componentes espectrales presentes en la señal y realizar un espectrograma que permite ver la evolución de las componentes espectrales en el tiempo.

El algoritmo diseñado para la evaluación tiene cinco elementos principales que se detallan a continuación: La primera parte lee una señal de audio ubicada

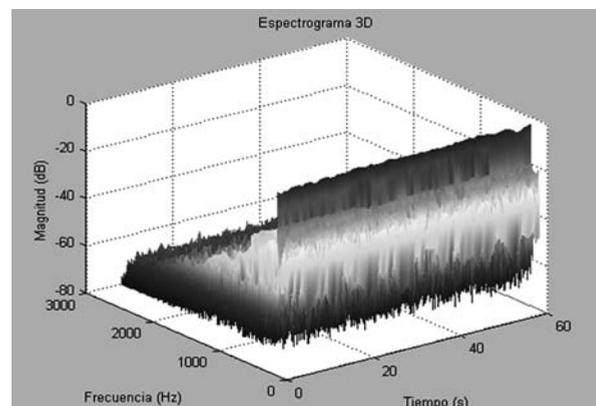
en memoria y gráfica la señal análoga de sonido, su componente espectral y la fase en la figura 6 el resultado de dicho código. La segunda parte del programa realiza la conversión del espectro a decibelios, en la tercera sección realiza un espectrograma utilizando la función de MATLAB. Specgram() [12], en el se puede ver la evolución de las frecuencias en el tiempo, esta función muestra el espectrograma en colores.

FIGURA 6. Salida del Algoritmo en Matlab



Para la cuarta sección del programa se utilizó una función que permite la visualización del espectrograma en escala de grises donde el negro indica mayor intensidad del sonido y el blanco la zona de menor intensidad. También se realizó un análisis en tres dimensiones con el fin de analizar con más detalle las señales, Esto se logra con la función surf(x,y,z) [13], la cual produce graficas de superficie, para este caso el eje x se ubico el tiempo, en el eje y el vector de frecuencias y en el eje z la amplitud del espectro en decibios. En la figura 7 se observa el resultado de dicho código.

FIGURA 7. Salida del Algoritmo 3D en Matlab



3.2 ALGORITMO ADAPTATIVO CON EL DSPIC33FJ256GP710

Para implementar los filtros adaptativos en la plataforma DSPIC, se diseñó un algoritmo que fue realizado en lenguaje ansi-c, empleando el compilador MPLAB C30 versión estudiantil de Microchip este algoritmo cuenta con los siguientes procesos:

En la primera parte del programa se incluyen las librerías matemática "math.h" y de procesos DSP "dsp.h" proporcionadas por el compilador C30, estas librerías facilitan el trabajo de la implementación de los filtros. En esta primera parte se definen las variables que son necesarias para el desarrollo del programa. Se diseñaron funciones con el fin de inicializar el CODEC SI3000, activando el reset del dispositivo y cargando todos sus registros de configuración. Otra función del código se encarga de configurar el modulo DCI que tiene el DSPIC, haciéndolo compatible con la configuración del SI3000, ya que sin este paso la comunicación entre los dos no sería posible. Para este programa el modulo DCI funciona como esclavo del códec y siempre está a la espera de datos.

La función principal main (void) configura el reloj del sistema para que opere a 40 MIPS, los puertos de entrada y salida, finalmente llama las funciones que configuran el modulo DCI y el SI3000. Cuando el numero de muestras es el adecuado se llama a la función de filtrado adaptativo esta función fue implementada en dos versiones, con datos tipo flotante y fraccionales, con 200 muestras de audio y variando la longitud del filtro M entre 2 y 8. La tabla 1 muestra los tiempos de ejecución en milisegundos.

TABLA 1. Tiempo de ejecución del algoritmo

| M | Punto flotante | Punto Fijo |
|---|----------------|------------|
| 2 | 5.19 | 0.95 |
| 3 | 7.08 | 1.27 |
| 4 | 8.97 | 1.59 |
| 5 | 10.86 | 1.91 |
| 6 | 12.75 | 2.23 |
| 7 | 14.64 | 2.55 |
| 8 | 16.53 | 2.87 |

4. EXPERIMENTOS EN TIEMPO REAL

Para evaluar el desempeño de la plataforma DSPIC33FJ256GP710 en el filtrado adaptativo de señales de audio, se diseñó el siguiente experimento: Se creó en MATLAB una señal de audio compuesta por la suma de la frecuencias tonales de 500,1000,1500,2300y 3000 Hz, se realizó el análisis espectral de la señal sin ruido aplicando el algoritmo desarrollado, a esta señal se le agregó de manera separada ruido sinusoidal con frecuencias de 300,800,1200,2000 y 3500 Hz, La señal con ruido fue reproducida en la tarjeta de audio de La PC y filtrada de manera adaptativa en la plataforma DSPIC y nuevamente reintroducida a la computadora donde se realizó nuevamente el análisis espectral con el fin de evaluar la efectividad del filtro, El esquema del experimento se muestra en la figura 1. En las figuras 8 y 9 se puede observar la señal de audio con ruido de 800 y 2000 Hz, En la figuras 10 y 11 las señales filtradas por el DSPIC donde se observa una mejor atenuación para la frecuencia de ruido de 2000 Hz. Otra prueba fue realizada reproduciendo una canción en formato WAV, a la que se le adicione ruido sinusoidal de diferentes frecuencias entre 300 y 4000 Hz, evaluando el resultado del filtrado adaptativo en las figuras 12 y 13 ilustran la canción con ruido de 1800 Hz y las figuras 14 y 15, muestran los resultados después de ser filtrada por el DSPIC. Para esta prueba se tomo un vector con 200 muestras y se varió el tamaño del filtro desde dos a doce coeficientes no pudiendo superar los tres coeficientes al trabajar con datos en punto flotante, ya que el tiempo de procesamiento se volvía inferior a la tasa de llegada de los datos, de diez mil muestras por segundo, caso contrario al usar la característica especial del DSPIC de procesar datos en punto fijo como si fueran fraccionales, donde se pudo alcanzar más de diez coeficientes. Al reproducir en tiempo real la señal filtrada el tono introducido no era perceptible y no se presentaba una alteración sustancial en el timbre del sonido.

FIGURA 8. Suma de señales con ruido de 800 Hz

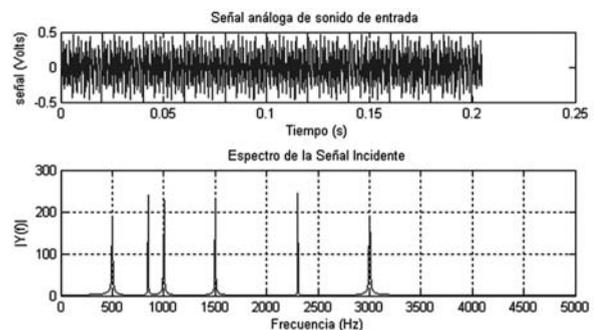


FIGURA 9. Suma de señales con ruido de 2000 Hz

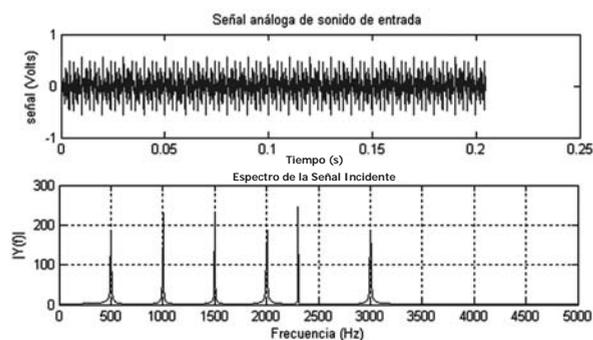


FIGURA 10. Señal de filtrada con ruido 800 Hz

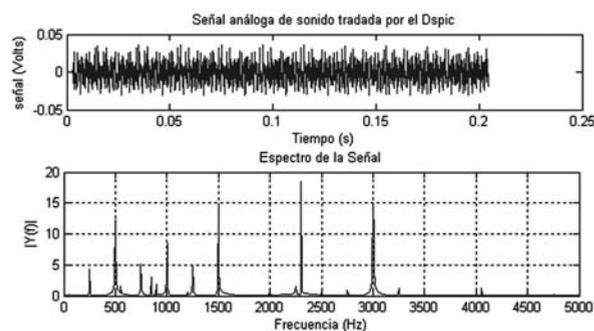


FIGURA 11. Señal de filtrada con ruido 2000 Hz

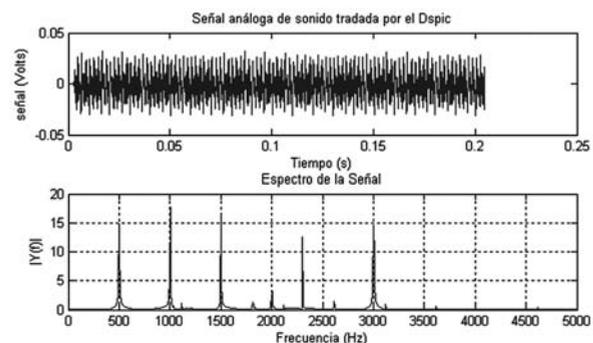


FIGURA 12. Señal de voz con ruido de 1800 Hz

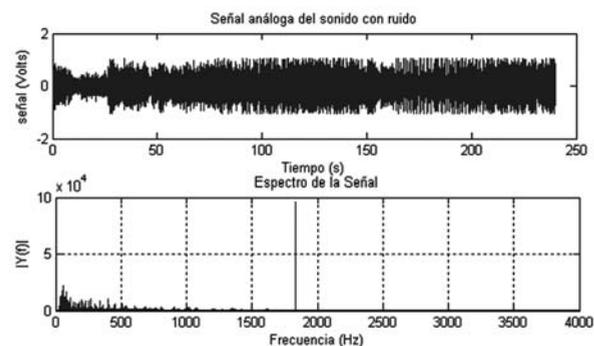


FIGURA 13. Espectro en DB de la señal de voz con ruido

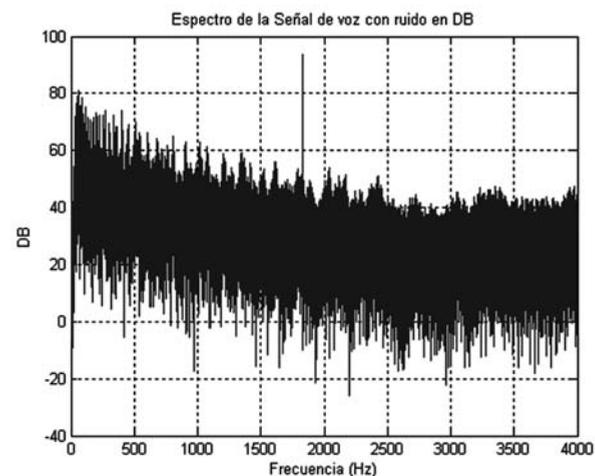


FIGURA 14. Señal de voz filtrada adaptativamente

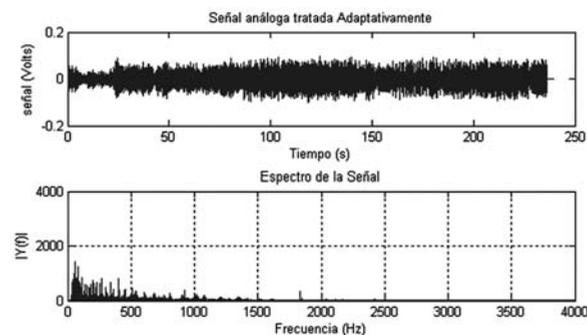
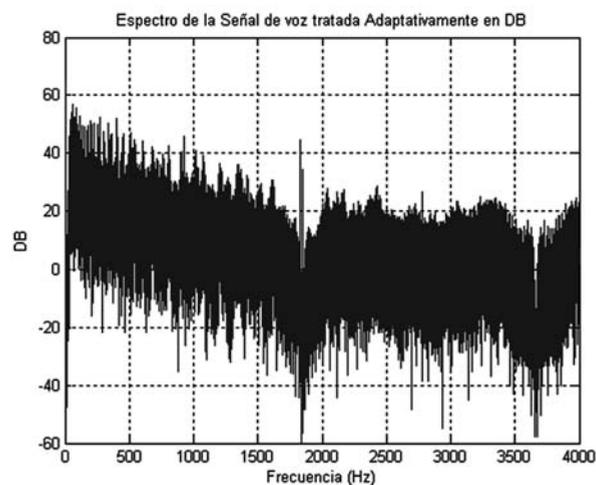


FIGURA 15. Espectro en DB de la señal filtrada adaptativamente.



5. CONCLUSIONES.

- Se lograron Mejores resultados para la eliminación del ruido acústico sinusoidal en el rango de los 1000 y 3000 Hz sin distorsiones en la señal de salida.
- Para determinar el factor de adaptación μ Se debe realizar varios ensayos ya que su magnitud esta directamente relacionado con la amplitud de la señal de referencia.
- Para el filtrado de señales sinusoidales es suficiente con una longitud del filtro de 2.
- La plataforma DSPIC mostro un bajo desempeño para el procesamiento de datos tipo flotante.
- La frecuencia de muestreo juega un papel de vital importancia para lo convergencia del algoritmo, se deben realizar pruebas de convergencia para determinar cuál es la mejor frecuencia de muestreo antes de implementar el algoritmo en tiempo real.
- El algoritmo implementado filtra las señales entiendo simulado con una longitud de filtro mayor de tres con excelentes prestaciones , pero en tiempo real no se pudo superar la barrera de los cuatro pesos con datos tipo float , ya que el procesador se retrasaba de manera considerable en el procesamiento y se producen efectos de audio no deseados como el eco.

6. REFERENCIAS

- [1] Sörnmo Leif, Laguna Pablo, Bioelectrical Signal Processing in Cardiac and Neurological Applications , ACADEMIC PRESS, (año 2005).
- [2] Farhang-Boroujeny B., "Adaptive Filters: Theory and Applications Willey & Sons, (año 1999)
- [3] Vicente Luis, Masgrau Enrique, Sebastián J. Miguel, Communication Technologies Group, Aragon Institute of Engineering Research, University of Zaragoza María de Luna, 1, E50018, Zaragoza, Spain
- [4] Microchip Technology Datasheet., [Online document]. Disponible en www.microchip.com/downloads/en/DeviceDoc/70286A.pdf
- [5] Oppenheim, Alan - Schafer, Ronald Tratamiento De Señales En Tiempo Discreto, Prentice Hall , (año 2004)
- [6] Angulo, José María Microcontroladores Avanzados Dspic, Thomson Education , (año 2006)
- [7] Proakis, Jhon G Tratamiento Digital De Señales, Pearson Education , (año 2006).
- [8] Widrow Bernard, Stearns Samuel D. Adaptive Signal Processing, Prentice Hall, (año 1985).
- [9] Mínguez Olivares, Antonio, Tesis doctoral Ingeniería avanzada para sistemas de control de ruido acústico aéreo mediante técnicas adaptativas, Universidad Politécnica de Madrid , (año 1998)
- [10] Shetty, Kiran Kumar A Novel Algorithm For Uplink Interference Suppression Using Smart Antennas In Mobile Communications, Florida state University, (año 2004)
- [11] Sen M. Kuo – Dennis R. Morgan. Active Noise Control System. Ed. New York, John Willey & Sons, (año 1996.)
- [12] Etter, Delores M. Solución de problemas de ingeniería con Matlab, Prentice Hall, México , (año 2005)
- [13] Báez López, David. MATLAB con aplicaciones a la ingeniería, física y finanzas, alfaomega (año 2006)