Metodología para la implementación de filtros en un ambiente académico.

Steven Camacho
Estudiante Universidad Militar Nueva Granada. u1400943@unimilitar.edu.co

Luis Quibano
Estudiante Universidad Militar Nueva Granada. u1400855@unimilitar.edu.co

Sebastián Rivera
Estudiante Universidad Militar Nueva Granada. u1400858@unimilitar.edu.co

Diego Sana
Estudiante Universidad Militar Nueva Granada. u1400799@unimilitar.edu.co

Juan Carlos Martínez
Estudiante Universidad Militar Nueva Granada. juan.martinezq@unimilitar.edu.co

Recibido: 7 de noviembre de 2013 Aprobado: 10 de diciembre de 2013

Artículo de investigación científica y tecnológica, como producto de la Universidad Militar Nueva Granada del grupo TIGUM de la facultad de Ingeniería de Telecomunicaciones.

Resumen

Este documento presenta una metodología para la implementación de filtros digitales FIR (respuesta finita al impulso) por medio del método de ventaneo, el cual ayuda a generar filtros precisos mediante el análisis matemático de la mano con la teoría. Se utilizan herramientas de hardware y software de uso común en un ambiente académico como Matlab, Labview y la plataforma Arduino. La metodología busca facilitar la interacción de hardware, software y señales muestreadas en el proceso de implementación y análisis de resultados de un filtro digital. Se caracteriza por la capacidad de cambiar los parámetros del diseño del filtro, sin afectar la arquitectura general de la implementación. Dentro de la metodología están contempladas fases de diseño, implementación del software, implementación del hardware y validación, permitiendo realizar análisis de resultados de forma práctica. La metodología se puede tomar como una etapa previa a la implementación del filtro digital sobre un DSP (procesador digital de señal), de tal forma que permite comprobar primero todos los aspectos teóricos y prácticos del diseño sin tener que entrar en los apartes técnicos de la arquitectura del dispositivo sobre el cual se implementará finalmente; básicamente el documento es una ayuda importante en la formación en un entorno académico y es base a la hora de diseñar filtros de manera fácil y práctica.

Palabras Claves : *Filtro digital FIR, ventaneo, Arduino, filtros con labview y matlab.*

Abstract

This paper presents a methodology for the implementation of digital filters FIR (finite impulse response) by the windowing method, which helps to generate accurate mathematical analysis filters by hand with theory. Hardware and software tools commonly used are used in an academic environment as Matlab, Labview and Arduino platform. The methodology aims to facilitate the interaction of hardware, software and sampled signals in the process of implementation and analysis of results of a digital filter. It is characterized by the ability to change the filter design parameters without affecting the overall architecture of the implementation. Within the methodology are referred to the design, software implementation, hardware implementation and validation, allowing analyzes of practical results. The methodology can be taken as a prerequisite for the implementation of digital filter on a DSP (digital signal processor), so that it allows a first check of all the theoretical and practical aspects of the design without having to go into the technical issues of the device architecture in which is finally implemente. This document is basically an iportant tool in the academic formation and it is an importnat base in designing filters in an easy and practical way.

Key words: *FIR digital filter, windowing, Arduino, filters with labview and matlab.*

I. INTRODUCCIÓN

El trabajo con procesamiento digital de señales involucra diseño teórico, simulación e implementación. Se toma como referencia el enfoque matemático para el análisis y selección de requerimientos para la implementación de un sistema de filtrado digital FIR.

El objetivo es desarrollar un método útil para la implementación de un filtro digital, partiendo de los requerimientos y haciendo énfasis en el cálculo matemático.

En la figura 1 se muestra el diagrama de bloques de la metodología propuesta. Para cumplir con las etapas de forma práctica, se deben usar herramientas que faciliten la implementación y minimicen la posibilidad de ocurrencia de errores. El uso de software como Labview y la herramienta de desarrollo de hardware Arduino, permiten al estudiante preocuparse únicamente por la implementación del algoritmo del filtro, dejando de lado la rigurosidad del diseño de firmware que podría tenerse con un DSP. Luego de verificar el funcionamiento y corregir posibles errores en el diseño, se procederá a la etapa de implementación del filtro sobre el DSP si así se requiere. De cualquier forma el estudiante maneja una interacción entre señales reales, hardware y software permitiéndole ir más allá de la simple simulación o verificación mediante software de la respuesta del filtro.

II. METODOLOGÍA

FIGURA 1. CRONOGRAMA DE IMPLEMENTACIÓN

Se dice que un filtro es un sistema o una red que cambia selectivamente la forma de onda, o las características amplitud-frecuencia o fase-frecuencia de una manera deseada (Proakis y Manolakis, 2007). Los objetivos comunes del proceso de filtrado son mejorar la calidad de la señal, por ejemplo removiendo o atenuando el nivel de ruido, extrayendo información de dos o más señales previamente combinadas para hacer uso eficiente de un canal de comunicación entre otras.

El uso de filtros análogos es común, especialmente por su facilidad en el análisis frecuencial y fácil implementación con componentes discretos, sin embargo, tienen grandes limitaciones como precisión, tolerancia y estabilidad; por otro lado los filtros digitales tienen mayor estabilidad, inmunidad al ruido y reconfigurabilidad, permitiendo modificación de sus características con solo cambiar algunos parámetros del código (Universidad Nacional del Sur, 2013).

Se conocen usualmente dos tipos de filtros digitales que se eligen según las necesidades y la naturaleza del problema. Estos filtros se les conocen como FIR e IIR (respuesta infinita al pulso). En este caso se tratará el diseño de filtros FIR, ya que su implementación es de uso práctico y su respuesta en fase es lineal, lo cual lo hace útil para muchas aplicaciones. Esta metodología está dirigida a los programas académicos que impartan teoría de señales y procesamiento digital para que el estudiante pueda orientarse de forma adecuada en una práctica de laboratorio de filtros digitales. A continuación se describe la metodología propuesta (Figura 1).

III. REQUERIMIENTOS

En el entorno académico se realizan aplicaciones de filtros digitales en diferentes campos: biométricas, acústicas, sísmicas, de instrumentación y comunicaciones de audio y datos. En estas aplicaciones se usan fundamentalmente para la separación de señales que han sido mezcladas o distorsionadas (Valeriano, Rojas, y Paz, 1999).

Por otro lado, dichas aplicaciones son implementadas en su gran mayoría sobre “Procesadores Digitales de Señal” (DSP) como lo señala Carlos Alva Coras y Flor Escuadra Galindo Universidad (2011), estos dispositivos son muy populares para procesamiento de audio y permiten un entendimiento gene-

Este artículo es el resultado de un proyecto de iniciación científica desarrollado en el grupo de investigación TIGUM del programa ingeniería en telecomunicaciones de la Universidad Militar Nueva Granada.
Camacho Steven, Quibano Luis, Rivera Sebastián y Sana Diego son estudiantes del programa y participantes en el proyecto de investigación. Martínez Juan es docente del programa y tutor del proyecto.

ral del funcionamiento de los filtros implementados sobre hardware usando señales reales. Para la implementación del filtro en este caso se dispondrá de un arduino y plataformas de software como Matlab y Labview; el uso de estas herramientas ayudará a comprender de una mejor manera la parte matemática que involucra el desarrollo del filtro digital, pudiendo así comprobar los resultados que se obtienen a partir de la gráfica del comportamiento del filtro usando Matlab con la señal mezclada y filtrada usando Labview. Se desea desarrollar un sistema que permita filtrar un tono DTMF, para su implementación en el laboratorio se debe disponer de un generador de tonos DTMF o generadores de frecuencia y osciloscopio, para comprobar que el tono corresponda para ser procesado.

Se eligió la plataforma arduino debido a su sencillez en la programación para la conversión de señales análogas que se pueden tomar directamente de un generador o de un transductor de audio, así el estudiante tiene la facilidad de aprender a programar y entender cómo se digitalizan las señales, además de la capacidad de comprender si son tonos puros, mezcla de varios tonos o señales donde interactúa el ruido.

IV. CARACTERÍSTICAS DEL FILTRO FIR Y CONSIDERACIONES DE DISEÑO

La función usada inicialmente para el diseño del filtro fue el comando FIR1 () del software Matlab; esta función usa por defecto el método de ventaneo y como función ventana w(n), usa el ventaneo de Hamming.

Para poder realizar filtros digitales existen tres técnicas de diseño de filtros FIR que son de gran importancia: La técnica de ventanas, la técnica de muestreo en frecuencia, la técnica de diseños con rizado uniforme.

La técnica de ventaneo se basa en la respuesta impulso de un filtro y a esta aplicarle la ventana deseada multiplicando sus ecuaciones. La ventana hace que en el filtro real diseñado se tengan menos variaciones de transición o supresión y con esto se logre un filtrado más efectivo tal como se evidencia en el enlace http://www.mathworks.com/help/signal/ref/fir1.html.

Es posible que inicialmente no se cuente con toda la información necesaria para especificar completamente el filtro, pero cuanto más detalles se conozcan más sencillo será el proceso de diseño.

El método por ventaneo de fase lineal, está relacionado con la siguiente fórmula.

$$H[n] = Hd[n] * w\ n]$$

Donde Hd[n] corresponde a la respuesta impulso del filtro (pasa bajo, pasa alto, pasa banda, rechaza banda); cada tipo de filtro tiene su correspondiente ecuación que los describe en tabla I.

TABLA.I
ECUACIONES CARACTERÍSTICAS DE FILTROS DIGITALES USANDO MÉTODO POR

Nombre del filtro	Coeficientes del Filtro
FILTRO PASA BAJOS	$h_d(n) = \frac{\sin n\omega_c}{n\pi}$ $para\ n \neq 0$ $h_d(0) = \frac{\omega_c}{\pi}$ $para\ n = 0$
FILTRO PASA ALTOS	$h_d(n) = -\frac{\sin n\omega_c}{n\pi}$ $para\ n \neq 0$ $h_d(0) = 1 - \frac{\omega_c}{\pi}$ $para\ n = 0$
FILTRO PASA BANDA	$h_d(n) = \frac{\sin n\omega_{c2} - \sin n\omega_{c1}}{n\pi}$ $para\ n \neq 0$ $h_d(0) = \frac{\omega_{c2} - \omega_{c1}}{\pi}$ $para\ n = 0$
FILTRO RECHAZA BANDA	$h_d(n) = -\frac{\sin n\omega_{c2} - \sin n\omega_{c1}}{\pi}$ $para\ n \neq 0$ $h_d(0) = 1 - \frac{\omega_{c2} - \omega_{c1}}{\pi}$ $para\ n = 0$

Fuente: Proakis, Manolakis y Dimitris, 2007.

Uno de los efectos que se produce al crear filtros digitales es el fenómeno de Gibbs ocasionado por el truncamiento de la señal en un punto. Para disminuir el pulso creado por este fenómeno se usa la función ventana w(n).

La función ventana puede tomar cualquiera de las ecuaciones presentes en tabla II.

TABLA II.
FUNCIÓN VENTANA.

Nombre de la ventana	Funcion muestreada
Rectangular	$w(n) = 1$
Hanning	$w(n) = 0.5 + 0.5\cos\left(\frac{2\pi n}{N}\right)$
Hamming	$w(n) = 0.54 + 0.46\cos\left(\frac{2\pi n}{N}\right)$
Blackman	$w(n) = 0.42 + 0.5\cos\left(\frac{2\pi n}{N}\right) + 0.08\cos\left(\frac{4\pi n}{N}\right)$

Fuente: Mitra S, 2006

Para la realización del filtro FIR usando únicamente la función fir 1 () en Matlab hay que tener presente los siguientes parámetros:

$f_S = 10000Hz$ (Frecuencia de Muestreo)
$f_C = 900Hz$ (Frecuencia de corte)
$f_n = \frac{2*900Hz}{1000Hz} = 0.18$ (Frecuencia de corte normalizada)
$n = 12$ (Orden del filtro)

La línea de código para implementarlo en Matlab es B=fir1 (12,0.18); dando como resultado los siguientes coeficientes:

-0.0012 0.0032 0.0217 0.0648 0.1264 0.1825
0.2052 0.1825 0.1264 0.0648 0.0217 0.0032 -0.0012

La respuesta en frecuencia del filtro se muestra en la fig. 2. Al ubicarse en el punto de la frecuencia de corte normalizada se encuentra que el sistema atenúa la señal en 4.92dB, y la desfasa aproximadamente en 196°.

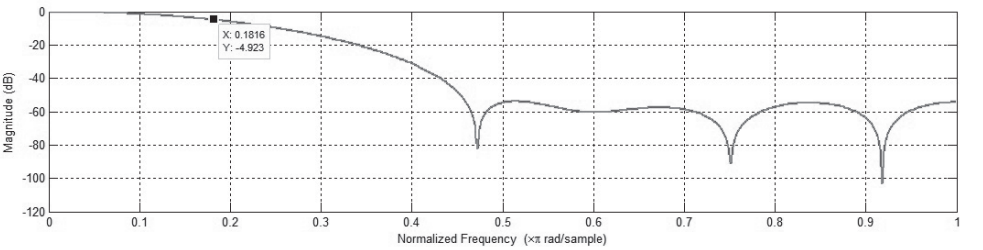


FIGURA 2. RESPUESTA EN FRECUENCIA DEL FILTRO FIR DE ORDEN 12

Una vez realizados los cálculos se comprueban los resultados usando la tabla I que describe la función del filtro y tabla II como función ventana.

Para comprobar la función fir1 que se encuentra en Matlab, se procede a realizar los cálculos matemáticos uno a uno para comprobar los coeficientes del filtro de orden 12.

$$f_n = \frac{900Hz}{1000Hz} = 0.09$$

$N = 12+1$
$H[0] = Hd[0] * w[0]$

$$Hd[0] = \frac{2\pi(0.09)}{\pi}$$
$Hd[0] = 0.18$
$$w[0] = 0.54 + 0.46\,Cos\left(2\pi\frac{(0)}{13}\right)$$
$w[0] = 1$
$H[0] = 0.18 \times 1$
$H[0] = 0.18$

$H[1] = Hd[1] * w[1]$

$$Hd[1] = \frac{\sin((1)2*\pi*(0.09))}{(1)\pi}$$
$Hd[1] = 0.1706$
$$w[1] = 0.54 + 0.46\,Cos\left(2\pi\frac{(1)}{13}\right)$$
$w[1] = 0.9473$
$H[1] = 0.1706 \times 0.9473$
$H[1] = 0.1616$

$H[2] = Hd[2] * w[2]$
$$Hd[2] = \frac{\sin((2)2*\pi*(0.09))}{(2)\pi}$$
$Hd[2] = 0.1440$
$$w[2] = 0.54 + 0.46\,Cos\left(2\pi\frac{(2)}{13}\right)$$
$w[2] = 0.8013$

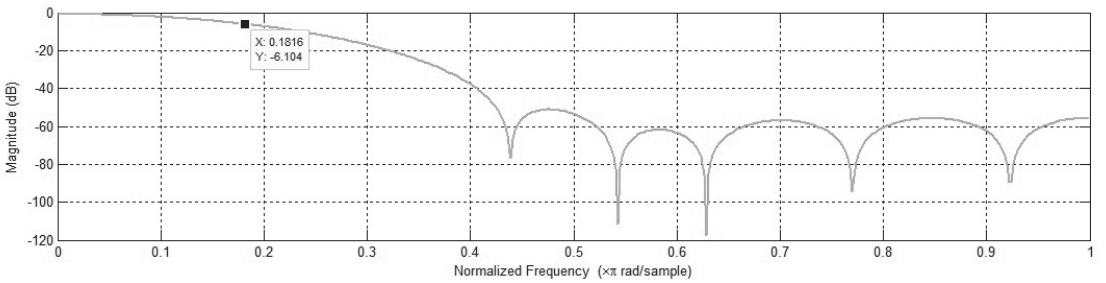


FIGURA 3. RESPUESTA EN FRECUENCIA DEL FILTRO EMPLEANDO LOS CÁLCULOS MATEMÁTICOS

TABLA.III
CÓDIGO USADO EN MATLAB PARA LA IMPLEMENTACIÓN DEL FILTRO

```
N=12 %orden del filtro
n=[-(N/2):N/2] %Longitud del Filtro
fc= 0.09; %Frecuencia normalizada
hd = 2*fc*sinc(2*n*fc); %LPF
w = 0.54+0.46*cos((2*pi*n)/13);
%Ventana Hamming
```

Si se revisa más detenidamente el comando fir1 se puede obtener los mismos coeficientes que se calcularon de forma matemática agregando el parámetro “noscale”.

Los valores de los coeficientes bn hallados forman la ecuación del filtro FIR pasa bajo de orden 12 y queda del siguiente modo:

$$y[n] = -0.0012 + 0.0032\,Z^{-1} + 0.0217Z^{-2} + 0.0648Z^{-3} + 0.1264\,Z^{-4} + 0.1825\,Z^{-5} + 0.2052\,Z^{-6} + 0.1825\,Z^{-7} + 0.1264\,Z^{-8} + 0.0648Z^{-9} + 0.0217Z^{-10} + 0.0032\,Z^{-11} - 0.0012\,Z^{-12}$$

En la fig. 4 se muestra la forma en la que se implementó el filtro digital.

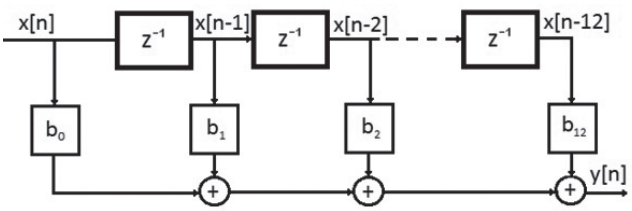
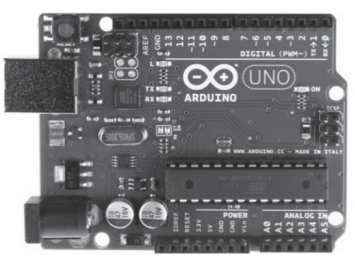


FIGURA 4. ESTRUCTURA DIRECTA DEL FILTRO IMPLEMENTADO.

V. DESARROLLO EN LA TARJETA DE ADQUISICIÓN

Arduino es una plataforma de desarrollo electrónico con microcontroladores de software y hardware abierto, se caracteriza por la facilidad en la implementación de aplicaciones ya que cuenta con un lenguaje sencillo y gran cantidad de librerías disponibles. En la Fig. 5 se muestra la tarjeta Arduino UNO, la cual cuenta con un microcontrolador ATMEGA328P, este dispositivo incluye un conversor análogo digital y comunicación serial asíncrona, lo cual lo hace ideal para construir un sistema de adquisición de señales.

FIGURA 5. TARJETA DE DESARROLLO ARDUINO UNO



La velocidad de transmisión es de 115200 bps y permite transmitir 11520 muestras por segundo, sin embargo, de acuerdo a las especificaciones del Arduino UNO, la máxima frecuencia de muestreo es:

$$fs = 10KS/s.$$

Circuito de acondicionamiento

El microcontrolador de la tarjeta Arduino UNO recibe señales análogas en un rango de 0 a 5V, por tanto, es necesario usar un circuito que permita dar un nivel dc a la señal a filtrar con el fin que no ingresar valores de voltaje negativos. El rango de amplitudes de la señal de entrada debe estar entre -2.5V y 2.5V. El circuito para este fin es un sumador con ganancia 1 y se muestra en la fig. 6.

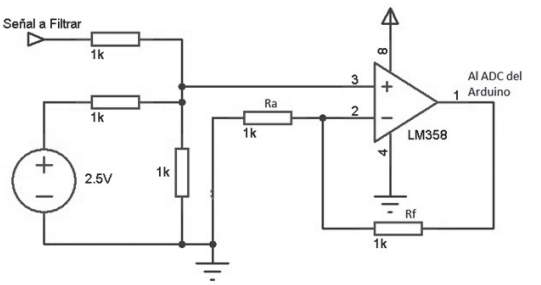


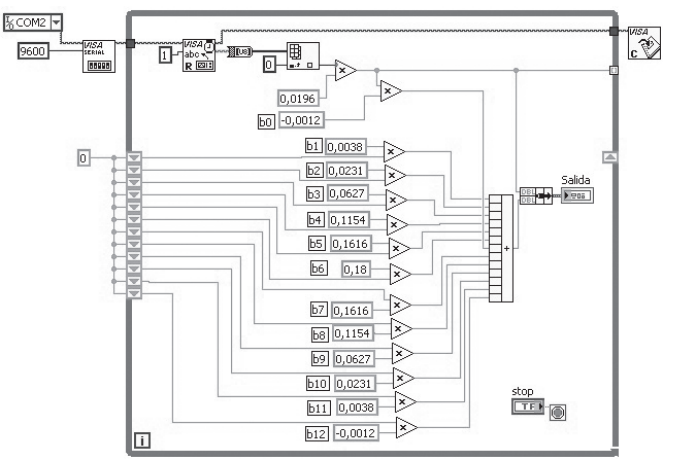
FIGURA.6 SUMADOR NO INVERSOR PARA LA ADQUISICIÓN DE LA SEÑAL.

VI. IMPLEMENTACIÓN

En la implementación del filtro se procede a realizar el programa en Labview utilizando los coeficientes obtenidos en los dos casos (Tabla.V) e introduciendo una señal con frecuencia de 900Hz para posicionarse en la frecuencia de corte de Figura a y Figura b correspondientemente.

TABLA IV

A) COEFICIENTES OBTENIDOS USANDO COMANDO FIR1 () E IMPLEMENTADO EN LABVIEW Y B) IMPLEMENTACIÓN DE LOS COEFICIENTES OBTENIDOS USANDO LOS CÁLCULOS MATEMÁTICOS E IMPLEMENTADO EN LABVIEW USANDO UNA SEÑAL SENO DE FRECUENCIA 900HZ



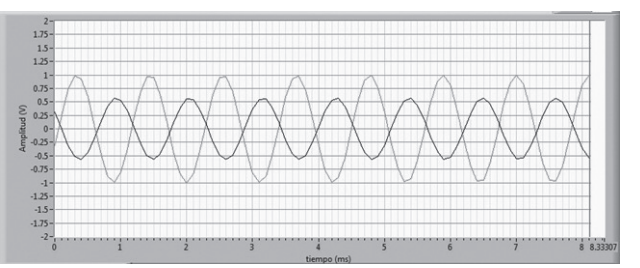
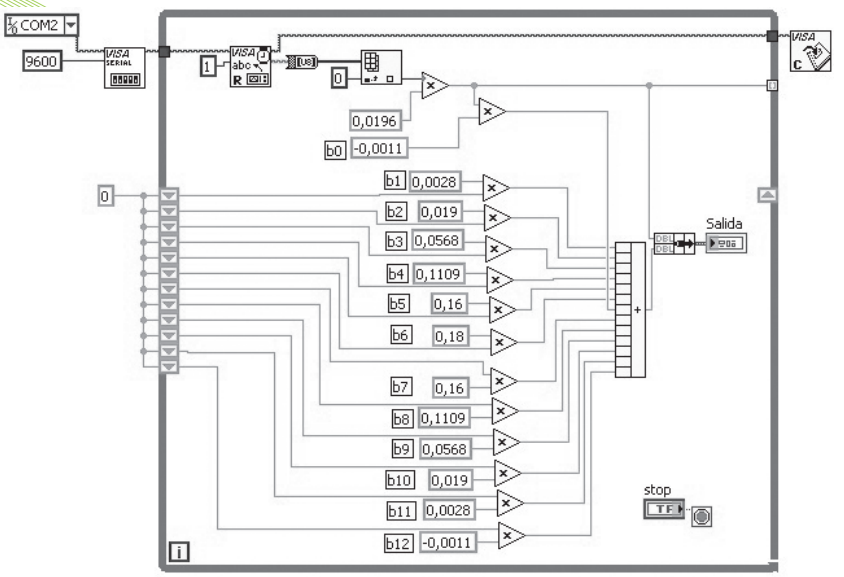
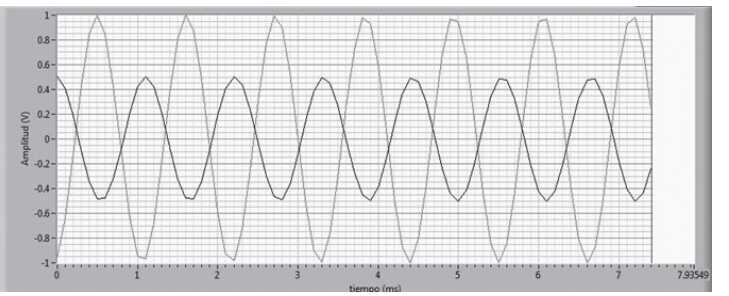


TABLA V
COMPARACIÓN DE A) Y B)





VII. VALIDACIÓN

Una vez realizada la implementación en Labview del filtro FIR se comprueba cuan atenuada se encuentra la señal cuando se usan diferentes frecuencias antes, durante y después de la frecuencia de corte registradas en Tabla.VI. Se establece un voltaje pico de entrada 1V y de esta manera se comprueba la curva de respuesta en frecuencia de fig. 2 y fig. 3

TABLA.VI
REGISTRO VALORES DE VOLTAJE DE ACUERDO A LA FRECUENCIA USADA

[Hz]	Fir1() [Vp] sin "scale"	Ganancia [dB]	Matemático [Vp]	Ganancia [dB]
100	0.99	-0.087	0.904	-0.876
600	0.775	-2.21	0.7025	-3.067
900	0.57	-4.88	0.49	-6.19
1500	0.182	-14.79	0.127	-17.92
1700	0.099	-20.08	0.07	-23.1

De acuerdo a los resultados obtenidos en la Tabla VI se puede comprobar que el proceso matemático se acerca más a la teoría de filtros que la función fir1 sin escalar porque la atenuación en el punto de corte es aproximadamente la mitad de la señal de entrada.

Una vez implementado el filtro digital usando los coeficientes obtenidos de forma matemática se procede a realizar el filtrado de una señal DTMF con frecuencia baja de 697Hz y una frecuencia alta de 1477Hz que corresponde al tono de la tecla tres (3), filtrando la señal de frecuencia baja vista en la fig. 7 y comprobando de esta manera la implementación del filtro.

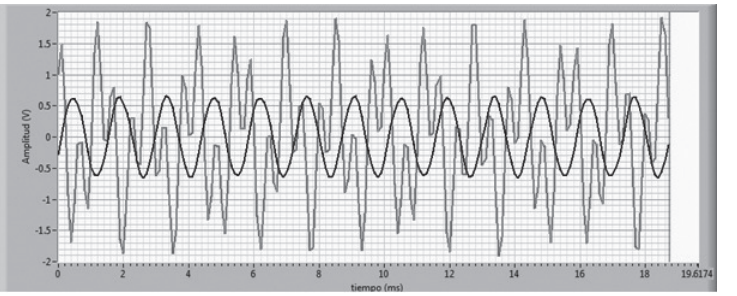


FIGURA 7. SEÑAL FILTRADA DEL TONO DTMF TECLA 3

Tal como se observa en la fig. 8, los componentes frecuenciales del tono DTMF corresponden a una frecuencia baja y una frecuencia alta, en este caso, el número tres. Una vez ha sido procesada la señal por el filtro, se observa que la componente de la frecuencia alta (1477Hz) fue atenuada, dando un voltaje pico de aproximadamente 0.127Vp corroborando una vez más los valores obtenidos en Tabla VI.

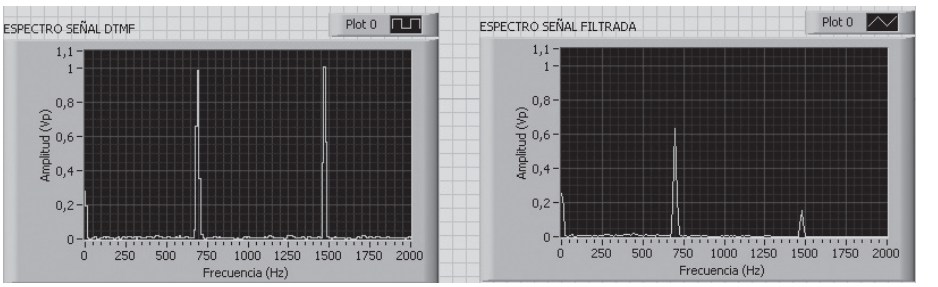


FIGURA 8. ESPECTRO DE LA SEÑAL DTMF ANTES Y DESPUÉS DE SER FILTRADA

VIII. CONCLUSIONES

Se evidencia que la matemática usada para la técnica de ventaneo se aplica de forma clara a la práctica de filtros con señales reales y se corrobora que en el punto de la frecuencia de corte la atenuación del sistema es alrededor de los 6dB de modo tal que comparado el resultado real con el resultado del comando fir1(), es importante destacar que los resultados se asemejan usando la metodología con las herramientas planteadas.

La implementación del software para el diseño del filtro digital es de gran utilidad, ya que ayuda al estudiante a analizar el comportamiento del filtro de una manera más fácil llevándose la idea básica de cómo podría ser la implementación en algún sistema más avanzado, como la implementación de un sistema de ecualización en un DSP.

Se comprueba que la metodología es adecuada no sólo para la implementación de los filtros sino también para la validación rápida de resultados, debido a las facilidades que proporcionan las herramientas utilizadas.

Luego de la validación de resultados es posible seguir con la implementación del filtro sobre un DSP si esto es lo que se requiere, con la seguridad que no existen fallos en el diseño.

Las herramientas para diseño de filtros como la función fir1 de Matlab dará unos resultados pertinentes siempre y cuando se maneje de forma adecuada los parámetros de configuración.

Para mejorar las características del sistema del detector de tonos DTMF y poder identificar de una mejor manera qué tecla ha sido oprimida se necesitará aumentar el orden del filtro y por consiguiente implementar más coeficientes.

REFERENCIAS

Alva C. y Escuadra F. Universidad (2011) Ricardo Palma-Perú "XVIII International Congress of Electronic, Electrical and Systems Engineering" En: http://www.urp.edu.pe/pdf/ingenieria/electronica/DSP-

"Diseño de Filtros Digitales", (2013.) En: http://www.ingelec.uns.edu.ar/pds2803/Materiales/Cap07/07-Cap07.pdf

Mitra S. (2006) "Procesamiento Digital de señales". Mc Graw –Hill. Cap. 10.

Pearson, Prentice Hall (2007). "Diseño de Filtros Digitales", Agosto, 2013. En:
Proakis, John G; Manolakis, Dimitris G (2007). "Tratamiento digital de señal". Madrid

Valeriano C., Rojas A., Paz J. (1999). Universidad Nacional de Ingeniería Lima – Perú. En: http://fiee.uni.edu.pe/sites/default/files/ethan_frome.pdf.

"Window-based finite impulse response filter design – MATLAB 2_Filtros_Digitales_para_Proceso_de_Audio_en_tiempo_real.pdf.En: http://www.mathworks.com/help/signal/ref/fir1.html