

Filtros Digitales con implementación DSP para Aplicaciones en Procesamiento de Audio en Tiempo Real

Carlos Alva Coras y Flor Escuadra Galindo
Universidad Ricardo Palma – Perú
 Asesor: Dr. Ing. Oscar Penny Cabrera

Resumen— Este proyecto consiste en la aplicación de la teoría de procesamiento digital de señales para el área de procesamiento en tiempo real. Utilizando los recursos actualmente usado en la industria electrónica los DSP será clave para la implementación del proyecto y verificación de los resultados con las simulaciones de forma analítica. La múltiples herramientas de medida como los analizadores de espectro tienen además otros formatos de mostrar sus resultados como veremos el espectrograma será de utilidad para observar el ancho de banda del filtro FIR.

Además las múltiples opciones de programación que el mismo fabricante Texas Instruments brinda a sus usuarios para el desarrollo de aplicaciones en dispositivos procesador. Por último la grandes prestaciones de procesador de punto flotante como una herramienta poderosa evitar las desventajas de los procesadores punto fijo. Ahora bien el procesador implica un gasto extra de energía eléctrica pero brindando una alta performance para el procesamiento de audio.

Índices— PDS, Filtro Digitales, Procesamiento de Audio, Texas Instruments, DSP.

I. NOMENCLATURA

PDS : Procesador Digital de Señales.
 DSK : Kit de Iniciación a los DSP.
 FIR : Respuesta Finita al Impulso.
 ADC : Conversión Analógica a digital.
 VLIW : Palabra de Instrucción Muy Amplia.
 IIR : Respuesta Infinita al Impulso.
 DSP : Procesador Digital de señales.

II. INTRODUCCIÓN

El estudio de sistemas discretos es ahora muy amplio en aplicaciones diversas, además las tecnologías desarrolladas que están a disposición en el mercado existe variedad de productos para todo usuario según sus necesidades y las características dos o mas productos que cumple una misma función hará que uno se diferencie de otro según el

Este trabajo fue apoyado por la Facultad de Ingeniería de la Universidad Ricardo Palma.

Carlos Alva y Flor Escuadra, carrera Profesional de Ingeniería Electrónica de la Universidad Ricardo Palma, Distrito Surco, Lima - 33 PERÚ (e-mail: 200621140@mail.urp.edu.pe, 200616923@mail.urp.edu.pe).

Asesor Dr. Ing. Oscar Penny Cabrera de la Universidad Ricardo Palma, Distrito Surco, Lima-33 PERÚ (e-mail: opennyc@hotmail.com).

precio y prestaciones de este, referido a la Ingeniería Electrónica se vio el avance en los sistemas digitales realizando circuitos más eficaces que uno analógico.

Una de las empresas que brindan productos electrónicos de altas prestaciones es Texas Instruments siendo la gama de productos de este, las reconocidas familias de Procesadores digitales de señales DSP (acrónimo en inglés Digital Signal Processing Processor) TMS320, son los dispositivos electrónicos que son parte de muchos artefactos de uso cotidiano e industrial. Por ese motivo y su soporte al estudiante ha sido razón por el cual se tomo este estudio aplicado de la familia C6000 de Texas Instruments para ser más preciso se utilizo TMS320C6713 DSK teniendo como procesador DSP TMS320C6713 de punto flotante.

A continuación se mostrará el desarrollo del protocolo de investigación siendo como objetivo principal la implementación de filtros digitales recursivos por ser de gran importancia la linealidad de la fase para el procesamiento de audio en tiempo real.

III. PREPARACIÓN DEL TRABAJO TÉCNICO

Pues bien partimos de la ecuación de convolución, como el formato de filtro FIR.

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

Siendo:

M el número de coeficientes del filtro, también llamado pesos.

Y $x(n-i)$ es la señal de entrada en tiempos discretos retrasado por i . Desglosamos la ecuación para mostrar así como se aplicara los algoritmos en lenguaje de programación. Para esto se define ahora que los sistemas son causales en este estudio.

Sistemas causales

$$y(0) = h(0)x(0) \quad (2)$$

$$y(1) = h(0)x(1) + h(1)x(0) \quad (3)$$

$$y(2) = h(0)x(2) + h(1)x(1) + h(2)x(0) \quad (4)$$

Esto también se puede expresar gráficamente y con la estructura clásica del filtro FIR.

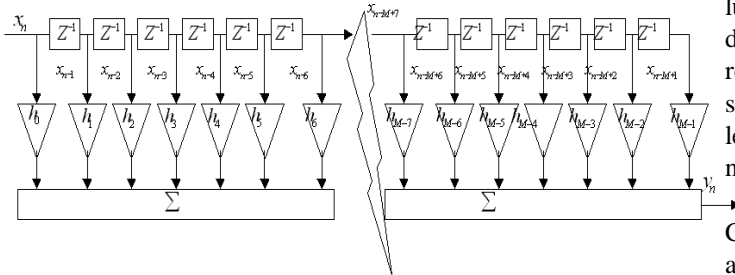


Fig. 1. Estructura de un Filtro FIR.

A. Uso de funciones ventanas:

Siendo la base la rectangular que se expresa en (5). Luego la ventana Hamming la cual se compara con la Hanning en la sección de simulación para luego pasar a la Implementación. Ecuaciones (6) y (7) son la ventana Hamming y Von Han respectivamente

$$Wr(n) = 1 \tag{5}$$

$$Whm(n) = 0.54 + (1 - 0.54) * \cos \left[\left(n - \frac{M-1}{2} \right) * \frac{\pi}{\frac{M-1}{2}} \right] \tag{6}$$

$$Whn(n) = 0.5 + (1 - 0.5) * \cos \left[\left(n - \frac{M-1}{2} \right) * \frac{\pi}{\frac{M-1}{2}} \right] \tag{7}$$

A1. Implementación en un DSP

“Pues se observa que el algoritmo estándar a utilizar será una operación MPY (Multiply) y ADD (suma) como instrucción iterativa a usar para el filtraje digital. Ahora bien el uso de lenguajes de programación es amplio haciendo que el usuario decide que es lo conveniente para su aplicación. Veamos el lenguaje ensamblador para observar las operaciones básicas a utilizar”. [1]

```

MVKL .S2 pt1, A5 ;A5 contiene el primer octeto y
MVKH .S2 pt1, A5 ;el segundo octeto de pt1
MVKL .S2 pt2, A6 ;A6 contiene el primer octeto y
MVKH .S2 pt2, A6 ;el Segundo octeto de pt2
MVKL .S2 pt3, A7 ;A7 contiene el primer octeto y
MVKH .S2 pt3, A7 ;el Segundo octeto de pt3
MVKL .S2 cuenta, B1 ;cuenta--> B1, B1 registro condicional
ZERO .L A4 ;Limpiar A4 (acumulador)
Lazo LDH .D *A5++, A0 ;A0 se carga el dato con la dirección A5
LDH .D *A6++, A1 ;A1 se carga el dato con la dirección A6
MPY .M A0, A1, A3 ;A3 tiene el producto de A0 y A1
ADD .L A4, A3, A4 ;por último se suman A4 y A3 y guarda en A4
SUB .S B0, 1, B0 ;decrementa cuenta
[B1] B .S lazo ;instrucción Branch condicional
STH .D A4, *A7 ;Se guarda resultado en la dirección que tiene A7
    
```

Fig. 2. Líneas de programación en assembler.

Para llegar al objetivo se tiene que tener a los registros A5 y A6 como punteros hacia los coeficientes donde se encuentran los pesos y datos muestreados respectivamente y A7 como registro puntero hacia el espacio de memoria donde se

guardara los resultados. Debe tenerse en cuenta que A5 apunta primero contiene h(0) y luego de un post-incremento apunta luego h(1) así sucesivamente caso contrario ocurre en la dirección que apunta A6. Primero A6 apunta a la muestra mas reciente y luego apuntara a la muestra anterior y así sucesivamente Es ahí donde viene la flexibilidad de pasar del lenguaje ensamblador al C o a unos lenguajes de más alto nivel.

Como se puede observar en las ecuaciones (1), (2), (3) el algoritmo consistiría en multiplicar los pesos de menor índice con las muestras más recientes como se observo realizando en ensamblador.

Como ejemplo de esto sea el arreglo X [M] con M elementos que cuenta como primer elemento la última muestra y el último elemento será la más antigua muestra. Pues así se cumpliría la ecuación de convolución del siguiente algoritmo. Ecuación 8. Tener en cuenta que en la ecuación 8, h[k] y x[k] son arreglos definidos en el programa C lo que realizaba A5 y A6 que contenían los espacios de memoria apuntados. Además el valor M sería el contenido en B1 (valor cuenta) y por ultimo A4 sería yn ya que es el acumulador. En este caso A7 estaría redundante ya que al final del algoritmo el resultado estaría en A4 en caso del ensamblador.

```

For (k=0, k < M ; k++) //Lazo con M iteraciones
yn+ = h[k]x[k]; // yn contiene la multiplicación del peso k con la muestra k.
For (i=M-1 ; i > 0; i--) //Lazo de delay
x[i] = x[i-1] //x[i] contiene una muestra anterior.
    
```

Fig. 3. Líneas de programación en lenguaje C.

Con su esquema tenemos:

$$y(n) = \sum_{k=0}^{M-1} h[k].x[k] \tag{8}$$

Pues ahora consistiría en encontrar la forma más adecuada con que se transfieren los datos en si dependería de la arquitectura del procesador y las características internas que brinda al usuario. Dicho esto, la frecuencia de muestreo estará limitada por el mismo periférico ADC que estará en el sistema procesador.

Para este caso mostraremos el uso de los procesadores de punto flotante que puede trabajar con datos con formatos, punto flotante simple (32 bits) y puntos flotantes de doble precisión (64 bits) los cuales nos brinda un amplio rango de valores con precisión. Estos formatos vienen siendo definidos por la IEEE Ver tabla I. [2]

TABLA I
RANGOS Y REPRESENTACIONES DE LOS TIPOS DE DATOS EN EL COMPILADOR C/C++

Type	Size	Representation	Range	
			Minimum	Maximum
char, signed char	8 bits	ASCII	-128	127
unsigned char	8 bits	ASCII	0	255
short	16 bits	2s complement	-32 768	32 767
unsigned short	16 bits	Binary	0	65 535
int, signed int	32 bits	2s complement	-2 147 483 648	2 147 483 647
unsigned int	32 bits	Binary	0	4 294 967 295
long, signed long	40 bits	2s complement	-540 755 813 888	540 755 813 887
unsigned long	40 bits	Binary	0	1 099 511 627 775
long long, signed long long	64 bits	2s complement	-9 223 372 036 854 775 808	9 223 372 036 854 775 807
unsigned long long	64 bits	Binary	0	18 446 744 073 709 551 615
enum	32 bits	2s complement	-2 147 483 648	2 147 483 647
float	32 bits	IEEE 32-bit	1.175 494e-38 ⁽¹⁾	3.40 282 346e+38
double	64 bits	IEEE 64-bit	2.22 507 385e-308 ⁽¹⁾	1.79 769 313e+308
long double	64 bits	IEEE 64-bit	2.22 507 385e-308 ⁽¹⁾	1.79 769 313e+308
pointers, references, pointer to data members	32 bits	Binary	0	0xFFFFFFFF

⁽¹⁾ Figures are minimum precision.

“Respecto al método de captura, el más indicado ante cualquier sistema DSP es el de interrupción del procesador ya se permitiría a optimizar más provecho al procesador de sus periféricos, etc. Por último, las frecuencias de muestreo son de 8, 16, 32, 44, 48, 96Khz que son características del periférico TLV320AIC23 códec que está integrada en la placa C6713 DSK de Texas Instruments. Dicho periférico brindaría 2 entradas micrófono o entradas de línea y 2 salidas, audífonos y la otra de Línea de salida. Con respecto a las líneas de entrada – salida cuenta con ganancia fija y cuenta con los conectores 3.5mm tipo estéreo”. [3]

“Siendo la voz humana estar sus principales componentes hasta una frecuencia de 4Khz. Esto implica tener en cuenta un filtro aliasing a la entrada del sistema para evitar frecuencias altas. Haciendo que la frecuencia de corte a 4Khz garantiza así que no habría inconveniente al muestrear al doble de la frecuencia de Nyquist. Pero esta aplicación es para uso de telefonía; entre otros existen el procesamiento profesional de audio donde existen frecuencia de muestreo superior que 8Khz sino también 16, 32, 44.1, 48 y 96Khz. Las cuales el de 44.1Khz es el estándar para los formatos aun usados en audios de calidad CD-ROM. Agregando también las frecuencias de muestreos múltiplos de 48Khz son usados en audio de alta definición DVD-A”. [4]

Los resultados serán puestos en prueba gracias a muchas ventajas de los depuradores de Texas Instruments. El Code Composer Studio es una herramienta grande para esto.

“Los C6x usan la arquitectura VLIW, permitiendo así que 8 instrucciones sean capturadas simultáneamente desde la memoria del chip a través de sus bus de 256 bitios de datos. La arquitectura original VLIW fue modificado por Texas Instruments permitiendo la ejecución de paquetes en la fase de Fetch”. [5]

B. Resultados

Con la implementación hecha se realiza la simulación teórica a través de Mathcad y la parte física se observa la salida de la

línea del DSP TMS320C6713 DSK, con los siguientes parámetros: El orden del Filtro es de 21. La frecuencia de muestreo es 48Khz. La frecuencia central (f0) es 10Khz y el ancho de banda de 2Khz.

En este caso se compara las ventanas para observar cual tiene mayor selectividad. Estas son: Rectangular (Color de línea negra), Von Han (Color Azul), Hamming (Color rojo) y Kaiser de parámetro 2. También sus respectivas graficas de fase. En la figura 4 y 5, se observa que ninguna tiene una selectividad adecuada según los parámetros de diseño pero por el contrario se obtiene buenas atenuaciones en los lóbulos laterales.

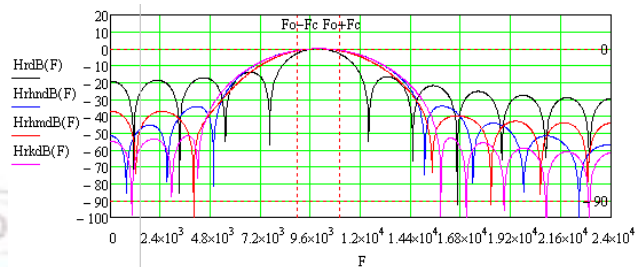


Fig. 4. Comparación de las selectividades de las ventanas

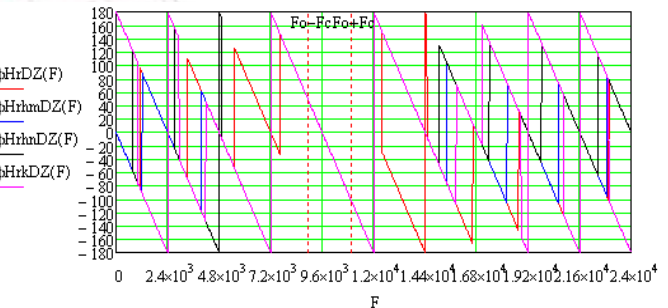
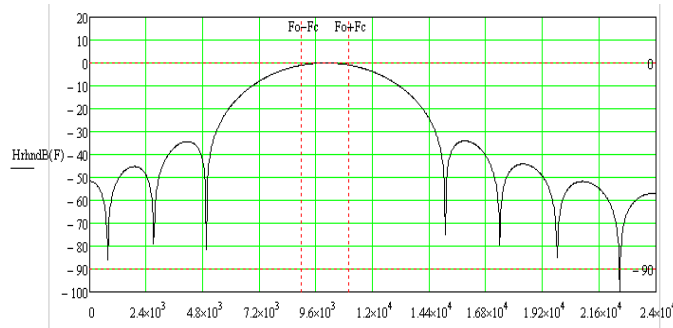
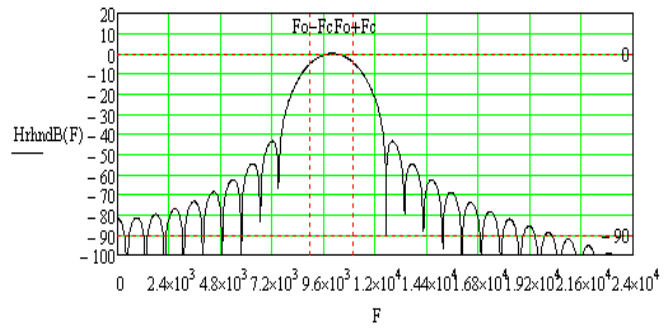


Fig. 5.- Graficas de Fases. Linealidad para todos los casos

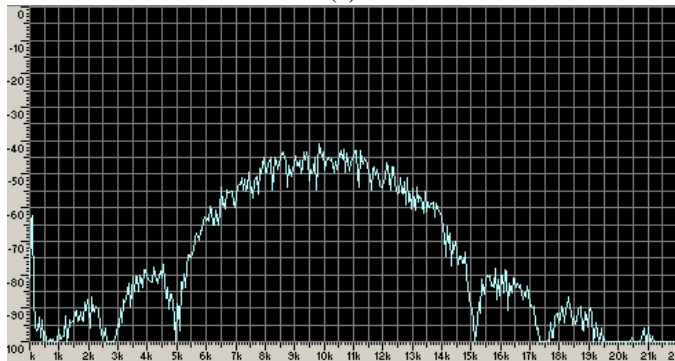
Ahora bien centrándonos en la ventana Von Han. Hacemos una comparación lo cual prueba que existe una mínima diferencia. Observar que en la implementación física tiene una atenuación de -40dB en el lóbulo principal por el motivo que el generador de ruido pseudoaleatorio tiene ese nivel de potencia. Es por eso si se observa el nivel de los lóbulos secundarios, es decir el nivel entre el lóbulo principal y el secundario es casi idéntico que el de la simulación. Estas muestras de los gráficos de la respuesta del DSP son gracias al paquete de software Goldwave por su económica herramienta virtual para la visualización de la señal en tiempo real. Figura 3 a, b y c.



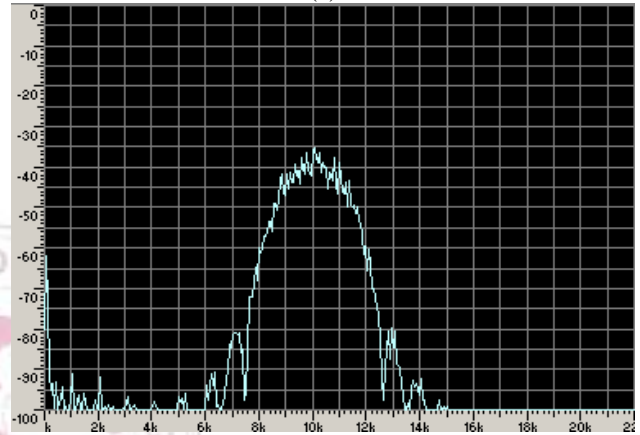
(a)



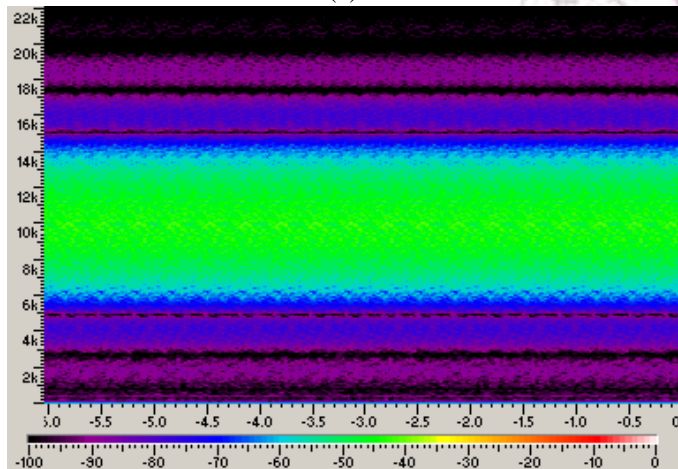
(a)



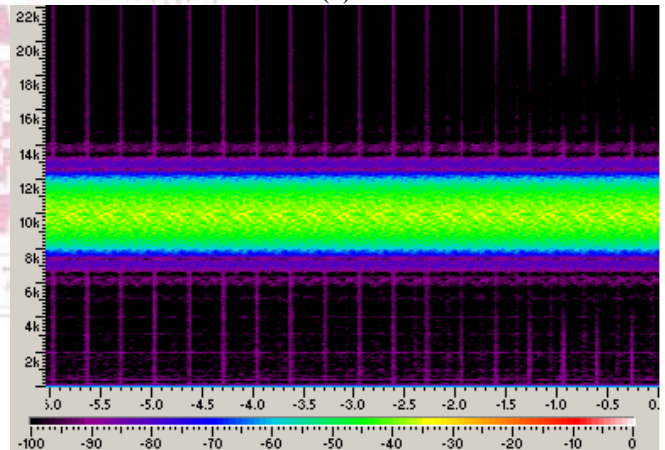
(b)



(b)



(c)



(c)

Figura 5.- Comparaciones de la simulación e implementación física. (a) Ventana en simulación Mathcad. (b) Muestra de Analizador de Espectro de Goldwave. (c) Espectrograma del filtro vía Goldwave

Figura 5.- Comparaciones de la simulación e implementación física. (a) Ventana en simulación Mathcad. (b) Muestra de Analizador de Espectro de Goldwave. (c) Espectrograma del filtro vía Goldwave

A este punto se observa que los parámetros de diseño no se satisfacen es mas el ancho de banda del filtro es muy angosta. Es por eso se ve obligado de incrementar el orden del filtro a 53 pesos para así mejorar la selectividad. Tener en cuenta que sigue siendo una implementación en punto flotante simple.

Como se observa las tres figuras siguientes muestran que se llega a mejorar su selectividad usando la misma ventana y frecuencia de reloj del procesador a 225Mhz. Figura 6 a, b y c.

C. Conclusiones

- Las aplicaciones de los filtros hoy en día es más amplio en múltiples áreas de la Ingeniería. Mostrar el uso en procesamiento de audio es uno rama de las tantas en cual se implementa los dispositivos de telefonía y audio profesional.
- La tasa de muestro se limitara por los periféricos externos a usar en un sistema procesador. Pues por otra parte se desarrolla ADC con frecuencia de muestro en

orden de los Mbps el cual permite el procesamiento a otros tipos de señales aparte del audio. Ya que la permite un mayor ancho de banda.

- La TMS320C6000 es una familia que por excelencia dedica su arquitectura a tratamiento de señales con alta performance y Texas Instruments brinda herramienta donde facilita el usuario en el manejo de estas. Es cierto que cada familia tiene una aplicación especial respecto a otra variando en su arquitectura y tecnología de consumo de energía. Esto brinda opciones al usuarios según el producto que desarrolla y que necesidades es relevante.
- La linealidad de la fase ante este tipo de procesamiento es necesario en el audio caso contrario se produciría distorsión de este. Ante ese motivo no se realizo el filtraje IIR ya que no se adecua a los requerimientos pero paso siguiente esta el filtraje adaptivo con el algoritmo LMS como base clave a la actualización de los pesos o coeficientes del filtro.

IV. AGRADECIMIENTOS

Los autores reconocen con agradecimiento las contribuciones de la Universidad Ricardo Palma, cuyo rector es PhD. Iván Rodríguez, a la Facultad de Ingeniería, cuyo Decano es el Ing. Rafael Taype, al laboratorio de Electrónica, a todo el Staff de Ingenieros, a nuestro Asesor, profesor y Director de Escuela Dr. Ing. Oscar Penny.

V. REFERENCIAS

- [1] (C6000 Teaching Materials. SSQC012. Easy to Use tutorials and Applications. By Dr Naim Dahnoun 2004. Texas Instruments). Chapter 1
- [2] SPRU187Q TMS320C6000 Optimizing Compiler v 7.0 - User's Guide - February 2010. Texas Instruments), Pag.139.
- [3] Communication System Design Using DSP Algorithms- with Laboratory Experiments for the TMS320C6713™ DSK- Steven A. Tretter- University of Maryland –College Park MD, Pag. 20.
- [4] Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416 DSK- SECOND EDITION. Rulph Chassaing - Worcester Polytechnic Institute and Donald Reay- Heriot-Watt University. A JOHN WILEY & SONS, INC., PUBLICATION
- [5] Real Time, Digital signal Processing, Based on the TMS320C6000. Nasser Kehtarnavaz, Pag. 31.

