

**LABORATORIO DE TRATAMIENTO DIGITAL DE SEÑALES**

**FRANCISCO JAVIER OSORIO MARTÍNEZ  
JHONNY MARK VELA SALCEDO**

**Trabajo de grado presentado como requisito para  
optar el título de Ingeniero Electrónico**

Director  
**ING. ENRIQUE VANEGAS**

**CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLIVAR**

**FACULTAD DE INGENIERIA ELECTRONICA**

**CARTAGENA DE INDIAS, D. T. Y C.**

**2003**

## **ARTICULO 105**

La Corporación Universitaria Tecnológica de Bolívar se reserva el derecho de propiedad intelectual de todos los trabajos de grados aprobados y no pueden ser explotados comercialmente sin autorización.

## CONTENIDO

Introducción	3
1. Generalidades Del Tratamiento Digital De Señales	5
2. Conceptos De Tratamiento Digital de Señales	7
2.1 Muestreo Y Cuantización	7
2.2 Transformada de Fourier. DFT. FFT.	13
2.3 Filtros Digitales	13
2.3.1 FIR	13
2.3.2 IIR	14
2.4 Filtros Adaptativos	15
2.4.1 Identificación	16
2.4.2 Modelización Inversa	17
2.4.3 Predicción	18
2.4.4 Eliminación De Interferencia	18
2.5 Estructura De Un Sistema De Tratamiento Digital De Señales	20
3. EL PROYECTO	21
3.1 Generalidades del Laboratorio de Tratamiento Digital de Señales	21
3.2 Practicas de Laboratorio	22
3.3 Módulo De Trabajo	23
3.4 Herramienta Software	23
4. Criterios utilizados en la selección del procesador.	24

4.1 Aritmética De Coma Flotante.	24
4.2 Coste	25
4.3 Popularidad En El Medio Universitario	26
5. Recomendaciones para el uso de la tarjeta TMS320C6711 DSK	26
6. Conclusiones	32
Referencias Bibliográficas	36
Anexos	
Lista de figuras	38
Glosario	39

## INTRODUCCION.

El Laboratorio De Tratamiento Digital De Señales se basa en una herramienta de prácticas fácil de usar que permita al estudiante y/o usuario que pretenda iniciarse en esta materia la implementación de algoritmos básicos de procesamiento de señales sobre un procesador específico, el TMS320C31, y comparar los resultados obtenidos en un sistema real con los que en teoría habrían de lograrse. Esta herramienta de prácticas a su vez la conforma un módulo de prácticas y un software.

El Módulo integrado en una sola caja, contiene la tarjeta DSP Starter Kit TMS320C3X de Texas Instruments de procesado digital, de la cual se han hecho accesibles desde el exterior de la caja la entrada y salida de la señal tanto para osciloscopio utilizando conectores BNC, como para interface con otros dispositivos de sonido o video utilizando conectores RCA. Todo el conjunto es alimentado por una fuente interna debidamente estabilizada. El módulo evita la necesidad de montar cables en cada una de las prácticas diseñadas prácticas, evitando así confusiones que podrían causar desperfectos y ahorrando un tiempo considerable. Se ha Implementado una herramienta *software* (la cual integra aplicaciones de filtros FIR, IIR, FFT y generación de señales) que permita al usuario interactuar con la tarjeta DSP Starter Kit TMS320C3X, poner a prueba sus conocimientos

sobre tratamiento digital de señales, implementar sus prácticas, evaluar de manera sencilla los resultados de estas, realizar diseños propios; reforzando de esta manera lo aprendido en el estudio teórico de esta materia.

La interacción del usuario con el procesador DSP se realiza desde un ordenador y desde éste es posible cambiar parámetros de diseño en tiempo real sin interrumpir al DSP. Se brinda también la posibilidad de un trabajo compartido entre el ordenador y el procesador DSP para tareas que involucren una mayor complejidad. Además se hace entrega de un CD AUXILIAR donde se encuentran debidamente ordenados todos los soportes teóricos, algoritmos, el instalador de la herramienta software y en general todo lo que se podría necesitar durante el desarrollo del laboratorio. Esperamos pues se contribuya de esta manera al desarrollo de esta rama de la electrónica en la institución.

## **1. Generalidades Del Tratamiento Digital De Señales**

El procesado digital de señales es una rama de la ciencia y la ingeniería que se ha desarrollado rápidamente durante los últimos treinta años. Este rápido desarrollo es el resultado de los avances tecnológicos tanto en los ordenadores digitales como en la fabricación de circuitos integrados. Los ordenadores digitales y el Hardware asociado hace tres décadas eran relativamente grandes y caros y como consecuencia su uso se limitaba a aplicaciones de propósito general en tiempo no real, tanto científicas como no comerciales. El rápido desarrollo de la tecnología de circuitos integrados, empezando con la integración a media escala (MSI) y continuando con la integración a gran escala (LSI) y ahora la integración a muy gran escala (VLSI) de circuitos electrónicos integrados ha estimulado el desarrollo de ordenadores digitales más potentes, pequeños rápidos y baratos han hecho posible construir sistemas digitales altamente sofisticados capaces de realizar funciones y tareas del procesado de señal que convencionalmente se realizaban analógicamente se realicen hoy mediante hardware digital, más barato y a menudo más fiable. Esto no quiere decir que el Tratamiento de señal digital sea la solución apropiada a todos los problemas de procesado de señal así para muchas señales de gran ancho de banda se requiere procesado en tiempo real. Para tales señales el procesado quizás óptico son las únicas soluciones válidas. Sin embargo cuando los circuitos digitales están disponibles y tienen la suficiente

velocidad suficiente, son normalmente preferibles. Los circuitos digitales no solo dan lugar a sistemas para el procesamiento de señales baratos y fiables sino que también tienen otras ventajas. En particular, el procesamiento digital de señal permite operaciones programables. Por medio de software se pueden modificar fácilmente las funciones de tratamiento de señal para que sean realizadas por el hardware. Por tanto el hardware digital y el software asociado proporcionan un mayor grado de flexibilidad en el diseño de sistemas. Además normalmente se consigue mayor precisión con el hardware digital y el software en comparación con los circuitos analógicos y los sistemas de procesamiento de señales analógicas. Por todas estas razones se ha producido un crecimiento explosivo de la teoría del tratamiento digital de señales y sus aplicaciones en las últimas tres décadas.

## **2. Conceptos De Tratamiento Digital De Señales**

En el CD auxiliar del laboratorio, en al carpeta anexos teóricos se ha dispuesto el documento teoría de señales y sistemas que contiene un resumen de los conceptos, pilares del Tratamiento digital de señales.

### **2.1 Muestreo Y Cuantización**

Cuando una señal es pasada de su forma análoga a la forma digital, esta es representada usando bits. Esta representación de un número finito de bits es llamada *Cuantización*. El número de bits usados para cada muestra determina la cantidad de cuantización hecha. Una muestra representada con 8 bits, tiene mayor precisión que una representada con 4 bits. La precisión requerida depende de la aplicación.

El proceso de pasar una señal análoga a digital, requiere de dos pasos que son mostrados en la figura 2.1.



Figura 1 Conversión análoga/digital

El primer paso es el muestreo y el segundo la cuantización. Del teorema de Nyquist sabemos que una señal muestreada a más de 2 veces su frecuencia más alta, puede ser recuperada exactamente. El paso de la cuantización es redondeando y este es irreversible. Cuando cuantizamos una señal, la estamos distorsionando y no puede ser recuperada a su señal original. Una forma de cuantización es el ruido aditivo y es mostrado en la figura 2.2

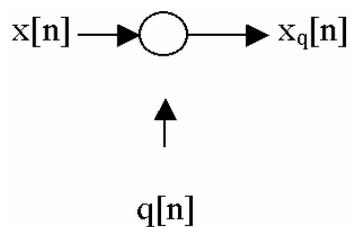


Figura 2. Modelo de cuantización

En este modelo,  $x[n]$  es la señal muestreada,  $q[n]$  es el ruido de cuantización, y  $x_q[n]$  es la señal cuantizada.

Estudiándolo desde el punto matemático, asumimos que cada número es representado con  $n$  bits, y que la magnitud entre los niveles de cuantización es  $\Delta$  (ver figura 2.3). Con  $n$  bits, hay  $2^n$  niveles de cuantización. El complemento a dos binario es usado generalmente para representar estos niveles, entonces con  $n$  bits y cuantización uniforme, los niveles van desde  $(-2^{n-1})\Delta$  hasta  $(2^{n-1}-1)\Delta$ . ( $2^n$  total). Por ejemplo, si  $n = 2$ , entonces hay cuatro niveles de cuantización, y usando complemento a dos, los niveles van desde  $-2\Delta$  hasta  $1\Delta$ .

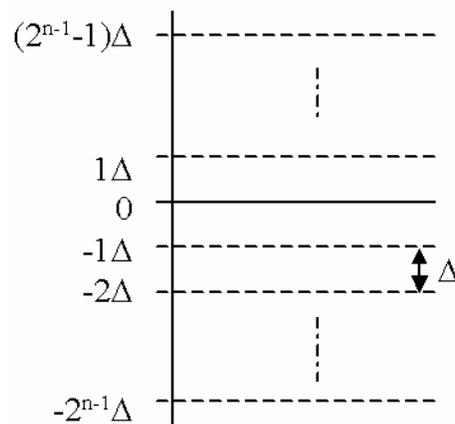


Figura 3.  $n$  niveles de cuantización.

Si  $\Delta$  es la magnitud entre los niveles de cuantización, la diferencia máxima entre  $x[n]$  y  $x_q[n]$  entre los niveles de cuantización es  $\Delta/2$ . En otras palabras,  $q[n]$  esta comprendido entre:

$$-\frac{\Delta}{2} < q[n] < \frac{\Delta}{2}.$$

$x_q[n]$  solo puede tomar valores entre  $(-2^{n-1})\Delta$  y  $(2^{n-1}-1)\Delta$ , y esta también limitada. Es recomendable para  $q[n]$  que este siempre limitado por  $\Delta/2$ , no solo entre los niveles de cuantización. Y puede ser hecho, limitando la señal entre:

$$\left(-2^{n-1}\right)\Delta - \frac{\Delta}{2} < x[n] < \left(2^{n-1} - 1\right)\Delta + \frac{\Delta}{2}.$$

El factor matemático puede ser usado para determinar cuantos bits son necesarios, dependiendo de la calidad de la señal deseada. La calidad de la señal es generalmente medida como la relación de ruido en dB. En este caso, la cuantización es el ruido, entonces la relación señal ruido esta definida como:

$$SNR = 10 \log_{10} \frac{\text{Power in Signal}}{\text{Power in Noise}}.$$

Si tenemos una señal seno,  $\text{Acos}(\omega n)$ , la potencia promedio es  $A^2/2$ . Para el ejemplo de la figura 2.3, la máxima potencia de la señal es:

$$P_x = \frac{\left( (2^{n-1} - 1)\Delta + \frac{\Delta}{2} \right)^2}{2} = \frac{\left( 2^{n-1}\Delta - \Delta + \frac{\Delta}{2} \right)^2}{2} \approx \frac{(2^{n-1}\Delta)^2}{2} = 2^{2n-3}\Delta^2.$$

La cuantización es ruido, y puede ser expresada como una ecuación. Es generalmente modelada como un proceso estocástico, si asumimos que la cuantización es independiente de la señal, blanca, y uniformemente distribuida de  $-\Delta/2$  a  $\Delta/2$ , entonces la potencia promedio en la cuantización es calculada de su varianza.

$$\begin{aligned} \text{Variance of } q[n] &= \sigma_q^2 = E\{q^2[n]\} - E^2\{q[n]\} \\ &= \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \alpha^2 f_q(\alpha) d\alpha = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \alpha^2 \frac{1}{\Delta} d\alpha = \left[ \frac{\alpha^3}{3} \frac{1}{\Delta} \right]_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} = \frac{\left(\frac{\Delta}{2}\right)^3 - \left(-\frac{\Delta}{2}\right)^3}{3} \frac{1}{\Delta} = \frac{\Delta^2}{12}. \end{aligned}$$

La relación señal ruido después de la cuantización es:

$$SNR = 10 \log_{10} \frac{2^{2n-3}\Delta^2}{\frac{\Delta^2}{12}} = 10 \log_{10} \frac{2^{2n-3}}{\frac{1}{12}} = 10 \log_{10} (12 * 2^{2n-3})$$

El TMS320C3X DSK cuantiza usando 14 bits y puede tener un máximo SNR de 86.05dB. Esto no significa que todas las señales tendrán un SNR de 86.05dB, este

el SNR de una onda seno usando la máxima amplitud de entrada. Si entramos una señal pequeña, algunos de los 14 bits serán siempre cero y el SNR será más pequeño.

La ecuación para SNR puede ser aproximadamente:

$$SNR = 10 \log_{10} \left( 12 * 2^{2n-3} \right) \approx 6.02n + 1.78.$$

Esto significa que hay aproximadamente 6dB de ganancia en SNR por cada bit adicional usado. Esto nos da una forma fácil de relacionar cuantos bits necesitamos para una calidad de señal deseada. Por ejemplo si fue determinado que el SNR fuera menor de 45dB, en base a esto seleccionamos 8 bits para el sistema. Otro ejemplo es el CD. Tomando en cuenta el ruido ambiental en el hogar, y los máximos niveles de sonido, 90-100dB. Un CD usa 16 bits y tiene un SNR de aproximadamente 98.1dB. En el DVD-Audio, la señal puede ser representada de 16 a 24 bits. 24 bits significan un SNR de 144 dB. 144 dB es bastante, no hay un circuito análogo que tenga este ruido tan bajo. El ruido térmico en una resistencia es más que esto, luego este SNR no es implementable a menos que el sistema corra cerca del cero absoluto.

Para más información remitirse al archivo: *teoría de sistemas y señales.doc* incluida en le CD del laboratorio en la carpeta anexos teóricos.

## **2.2 Transformada de Fourier. DFT. FFT.**

Ver archivo teoría de *sistemas y señales.doc* incluido en el CD del laboratorio.

Ver capítulos del 8 al 12 del libro: *The Scientist and Engineer's Guide to Digital Signal Processing*. Incluido en el CD del laboratorio ubicado en la carpeta *dsp book*.

Ver tema 6 del curso de DSP incluido en el CD del laboratorio y ubicado en la carpeta *curso dsp*.

## **2.3 Filtros Digitales**

### **2.3.1 FIR**

Ver archivo teoría de *sistemas y señales.doc* incluido en el CD del laboratorio.

Ver Capítulos 8 y 9 del curso de DSP, incluido en el CD del laboratorio, ubicado en la carpeta *curso dsp*.

Ver capítulos del 14 al 21, del libro *The Scientist and Engineer's Guide to Digital Signal Processing*. Incluido en el CD del laboratorio ubicado en la carpeta *dsp book*.

En un filtro en tiempo real, la convolución es hecha utilizando ecuaciones diferenciales. Si  $x[n]$  es la entrada y  $y[n]$  la salida, entonces la ecuación diferencial del filtro FIR es:

$$y[n] = h_0x[n] + h_1x[n-1] + \dots + h_{N-1}x[n-N+1]$$

### 2.3.2 IIR

Ver archivo teoría de *sistemas y señales.doc* incluido en el CD del laboratorio.

Ver Capítulos 8 y 9 del curso de DSP, incluido en el CD del laboratorio, ubicado en la carpeta curso dsp.

Ver capítulos del 14 al 21, del libro *The Scientist and Engineer's Guide to Digital Signal Processing*. Incluido en el CD del laboratorio ubicado en la carpeta *dsp book*.

La ecuación diferencial de filtro IIR es:

$$y[n] = b_0x[n] + b_1x[n-1] + \dots + b_{p-1}x[n-P+1] - a_1y[n-1] - \dots - a_{M-1}y[n-M+1]$$

## 2.4 Filtros Adaptativos

Un filtro adaptativo es un filtro cuyos coeficientes son ajustados de forma adaptativa en función de objetivos o condiciones variables en el tiempo que se traducen en una señal de error.

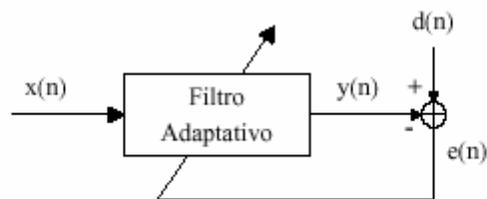


Figura 4 Diagrama Filtro Adaptativo

Una aplicación típica consiste en eliminar o modelar ciertas componentes (posiblemente indeseables), representadas por  $x(n)$  y proyectada sobre una señal  $d(n)$ , de acuerdo con algún criterio, incidiendo en una señal de error  $e(n)$ .

Un criterio típico y práctico para adaptar los coeficientes de un filtro y optimizar su desempeño, es minimizar el valor medio cuadrático de la señal de error  $e(n)$ .

Para entradas estacionarias y usando el criterio de minimización de error medio cuadrático, un filtro óptimo y único se diseñaría por filtro de WIENER; y para entradas no estacionarias los filtros de KALMAN son más adecuados y eficientes.

Los filtros adaptativos pueden ser de dos tipos FIR o IIR, sin embargo es casi siempre preferible utilizar la estructura FIR, ya que estas son más estables.

Los filtros adaptativos son usados en diversas aplicaciones y contextos, todo dependiendo como se quiera la respuesta deseada del sistema adaptativo. Hay esencialmente cuatro tipos de aplicaciones: Identificación, modelización inversa, predicción y eliminación de interferencia.

### 2.4.1 Identificación

En este tipo de aplicaciones el filtro adaptativo es usado para formar un modelo lineal que representa una mejor aproximación de acuerdo con algún criterio a un sistema desconocido.

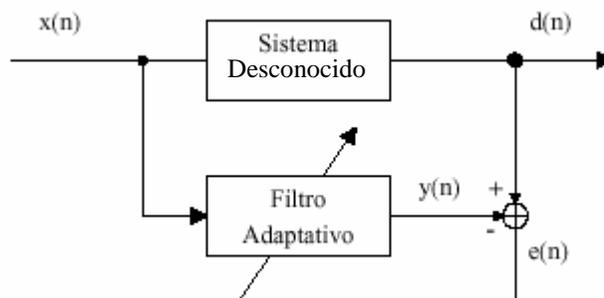


Figura 5. Identificación

## 2.4.2 Modelización Inversa

En estas aplicaciones el filtro adaptativo tiene por objetivo formar un modelo inverso que represente la mejor aproximación de acuerdo con algún criterio a un sistema desconocido.

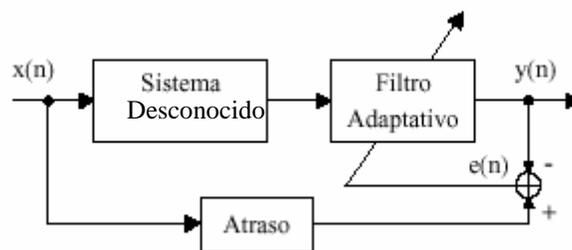


Figura 6. Modelización Inversa

Ejemplos de aplicación:  
Decorrelación predictiva  
Igualación adaptativa

### 2.4.3 Predicción

En estas aplicaciones el filtro tiene por objetivo formar una mejor predicción para la evaluación de una señal con base en un conocimiento de su pasado.

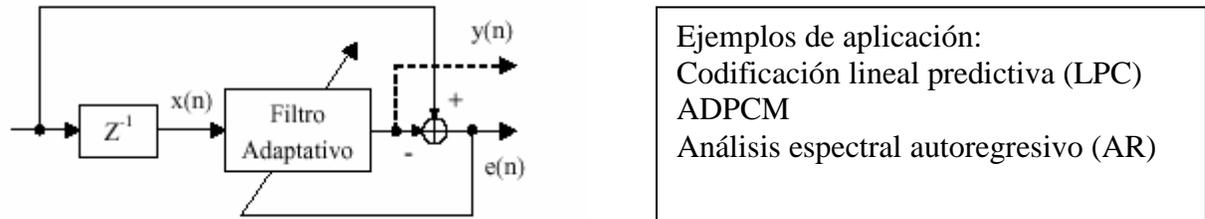


Figura 7. Predicción

### 2.4.4 Eliminación De Interferencia

En estas aplicaciones el filtro tiene por objetivo eliminar una interferencia desconocida contenida en  $d(n)$ , más conocida como señal de referencia  $x(n)$ . El objetivo es simplemente sustraer el ruido de la señal principal de forma adaptativa, de modo de mejorar la salida y la relación señal-ruido.

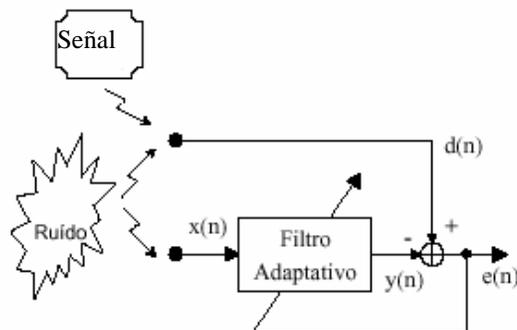


Figura 8. Eliminación de interferencia.

Dado que este último caso es el que servirá de base para el trabajo de laboratorio, procederemos a hacer algunas conclusiones pertinentes a partir del análisis de la señal y operaciones implicadas. Considerando que una señal  $d(n) = s(n) + x'(n)$  contiene una parte de información útil a preservar y una componente de ruido a eliminar  $x'(n)$ , se toman como hipótesis las siguientes:

$$E[s(n)x'(n-k)] = 0, \quad \forall k \qquad E[s(n)x(n-k)] = 0, \quad \forall k \qquad E[x(n)x'(n-k)] = p(-k)$$

O sea que  $s(n)$  es correlacionado con cualquiera de las representaciones de ruido y a su vez tiene una correlación cruzada, dada por  $p(k)$  para una distancia  $K$ .

La salida de un filtro adaptativo  $y(n)$  en una señal de error  $e(n)$ , es dada por:

$$y(n) = \sum_{k=0}^{N-1} h_x(k)x(n-k)$$

$$e(n) = d(n) - y(n) = s(n) + x'(n) - y(n)$$

Que coloca en evidencia que la señal  $s(n)$  fue parte de la señal de error. Por eso dadas las hipótesis anteriores, minimizar el error medio cuadrático de  $e(n)$ , equivale a minimizar el error medio cuadrático de ruido indeseado dado por:  $e(n) - s(n) = x'(n) - y(n)$ , lo que significa que  $s(n)$  no es afectado por el proceso de filtrado adaptativo.

Como caso limite de operación, en el filtro adaptativo tenemos dos situaciones, la primera en que  $y(n) = x'(n)$  y la segunda en que  $E[d(n)x(n-k)] = 0, \forall k$

En el primer caso de operaron el filtrado es efectivo, eliminando todo el ruido. En el segundo caso la señal de referencia es completamente descorrelacionada con la señal primaria, haciendo que el filtro adaptativo se auto-desligue, produciendo  $y(n) = 0$ , lo que afectaría la señal principal. Ver clases teóricas de la 7 a la 12 en formato *pdf* ubicadas en el CD del laboratorio, carpeta *anexos teóricos*, subcarpeta *uniporto* y subcarpeta *prácticas y teoría en PDF*.

## **2.5 Estructura De Un Sistema De Tratamiento Digital De Señales**

El procesado digital de señales proporciona un método alternativo para procesar una señal analógica. Para realizar el procesado digitalmente se necesita una interfaz entre la señal analógica y el procesador digital. Esta interfaz se denomina Conversor Analógico-Digital (A/D). La salida del conversor analógico digital es una señal adecuada como entrada al procesador digital. El Procesador Digital De Señales puede ser un gran ordenador digital programable o un pequeño microprocesador diseñado para la manipulación matemática de señales representadas digitalmente, es decir es quien realiza las operaciones deseadas sobre la señal de entrada. Las máquinas programables proporcionan la flexibilidad de cambiar las operaciones de procesado de señales mediante un cambio del software. En consecuencia, los procesadores de señales programables son de uso muy frecuente. En aplicaciones donde la salida digital del procesador digital de

señales se ha de entregar en forma analógica, como en comunicaciones digitales, debemos proporcionar otra interfaz desde el dominio digital al analógico. Tal interfaz se denomina Conversor Digital-Analógico. No obstante existen otras aplicaciones prácticas que requieren análisis de señales en la que la información deseada se encuentra en formato digital y no se requiere ningún conversor D/A. Por ejemplo, en el procesado digital de señales de radar, la información extraída de la señal de radar, como la posición de la nave y su velocidad, se puede imprimir directamente sobre papel. En este caso, no hay necesidad de conversor D/A.

### **3. EL PROYECTO**

#### **3.1 Generalidades del Laboratorio de Tratamiento Digital de Señales**

Actualmente la Corporación Universitaria Tecnológica De Bolívar no ha definido la infraestructura de un laboratorio para la asignatura tratamiento digital de señales, la cual es de alto grado de importancia en el currículo académico impartido en otras universidades de otros países. El proyecto busca suplir esta necesidad creándose la infraestructura modelo de lo que sería un laboratorio de DSP para la universidad, *en su etapa número uno*, la cual está conformada por:

### **3.2 Prácticas de Laboratorio**

Siete prácticas de laboratorio iniciales donde se cubren conceptos fundamentales del Tratamiento Digital de Señales haciendo énfasis en la implementación en tiempo real de procesos de filtrado, transformada de Fourier, generación de señales y sonido. En esta etapa inicial del laboratorio el estudiante no tiene que desarrollar algoritmos en las prácticas, pues demanda conocimientos de programación en ensamblador algo avanzado o en su defecto lenguaje c, además de un manejo muy versátil de la teoría del TDS (ecuaciones en diferencia, FFT, manejo de notación en coma flotante, etc.), pero precisamente se busca motivar al alumnado a introducirse en este campo y profundizar en el tema, puesto que La idea es que el estudiante aprenda a implementar algoritmos y continúe desarrollando y evaluando conceptos después de finalizadas las prácticas. Para ayudar a este propósito se expone una clara explicación y amplia teoría sobre el procesador utilizado y fundamentos para su programación. Recalcándose que no es el fin de esta etapa inicial. De esta parte del proyecto se elaboraron los siguientes documentos:

**Guías de prácticas para estudiantes**

**Guías de prácticas para el docente**

Que hacen parte del producto final entregado del proyecto.

### 3.3 Módulo De Trabajo

Propuesto y construido, que permite la realización de dichas prácticas y el cual es un modelo para futuros módulos que se deseen implementar pero con otro procesador y su respectiva tarjeta madre (básicamente como se expone en la sugerencia de la utilización del TMS320C6711 DSK). La descripción y funcionamiento se describen en el **Manual De Operación Y Mantenimiento Del Modulo De Prácticas Y Del Software C3XDSK 1.0** que hace parte del producto final entregado del proyecto.

### 3.4 Herramienta Software

Permite interactuar con el módulo, permitiendo ensamblar, cargar programas a la tarjeta, implementar filtros, FFT, etc. Muy sencilla de manejar.

Aunque las prácticas se desarrollaron para estudiar conceptos fundamentales y confrontar los conocimientos adquiridos, el estudiante podrá aprender por si mismo a diseñar sus propios procesos. De tal forma que se aborde la continuación del proyecto desarrollando más y mejores prácticas y hasta aplicaciones útiles.

Además se añaden otras herramientas software ya creadas para aquel estudiante que desee experimentar y afianzar conocimientos. La descripción y

funcionamiento se describen en el **Manual De Operación Y Mantenimiento Del Modulo De Prácticas Y Del Software C3XDSK 1.0.**

Se han tomado como modelos los laboratorios del FIU (universidad internacional de Florida) y Universidad de Puerto Rico Mayagüez (UPRM) y la universidad de Porto, Portugal<sup>1</sup>.

#### **4. CRITERIOS UTILIZADOS EN LA SELECCIÓN DEL PROCESADOR.**

Básicamente se escogió el procesador TMS320C31 por las siguientes razones:

##### **4.1 Aritmética De Coma Flotante.**

Una de las características fundamentales de los procesadores digitales programables es el tipo de aritmética utilizada por el procesador. La mayor parte de los DSP usan aritmética de coma fija, donde los números se representan como enteros o como fracciones entre -1.0 y +1.0. En la aritmética de coma flotante, los valores se representan por una mantisa y un exponente como  $\text{mantisa} \times 2^{\text{exp}}$ . La

---

<sup>1</sup> Ver archivos FIU.pdf y UPRM.pdf incluidos y el contenido de la subcarpeta carpeta uniporto en el CD auxiliar del laboratorio, en la carpeta anexos teóricos.

mantisa generalmente es una fracción con rango entre -1.0 y +1.0, mientras el exponente es un entero que representa en binario el número de lugares a partir de la coma que se debe desplazar a izquierda o derecha para obtener el valor representado. La aritmética en coma flotante es mucho más flexible que la de coma fija. En coma flotante, los diseñadores de sistemas tienen acceso a un rango dinámico más amplio (la distancia entre mayor y el menor valor que puede representar). En coma flotante el programador no necesita conocer en muchos casos ni el rango dinámico ni la precisión, mientras que, en coma fija, los programadores han de ser cuidadosos asegurándose de que sus señales no excedan el rango dinámico. Las características técnicas del procesador las encontramos en el libro *Manual De Mantenimiento Del Modulo Y Del Software*, en el archivo *1.pdf* ubicado en el CD del laboratorio en la carpeta anexos teóricos y en el archivo *guía funcional del TMS320C3X DSK.doc* ubicado en la misma carpeta.

## **4.2 Coste**

Los DSP de coma flotante son generalmente más fáciles de programar que sus correspondientes de coma fija, pero normalmente más caros. El incremento del coste se debe a la compleja circuitería necesaria para realizar los procesos en coma flotante. Se escogió el TMS320C31 pues era el procesador de coma flotante

y 32 bits, de menor coste de Texas Instruments al momento de efectuar la adquisición<sup>2</sup>, pero con todas las ventajas que implicaban tales características. Además el relativo bajo coste de este DSP lo hace más accesible para quienes tienen propósitos académicos.

### **4.3 Popularidad En El Medio Universitario**

Esta tarjeta y procesador se utiliza en las universidades como Universidad De Puerto Rico Mayagüez, Universidad Internacional De Florida, Universidad de Porto, Portugal, Escuela Universitaria Politécnica De Valladolid España, Universidad de Patras entre otras donde hace tiempos se experimenta en tiempo real en esta asignatura, contando estas universidades con grupos de desarrollo de aplicaciones en el área del tratamiento digital de señales.

## **5. Recomendaciones Para El Uso De La Nueva Tarjeta C6711 DSK**

Después de analizar varias opciones para el cambio de tarjeta teniendo en cuenta diversos aspectos como son: la velocidad de procesamiento, espacio de memoria, puertos, posibilidad de expansión, y características de lenguaje de programación

---

<sup>2</sup> Debido a la decisión de Texas Instruments de discontinuar la fabricación de la tarjeta madre TMS320C3X, DSK se optó por sugerir el C6711 DSK: Board + DSP TMS320C6711, más costoso pero con características mejoradas (además de poseer aritmética de coma flotante y palabra de datos de 32 bits).

que no diferenciaran mucho de las tarjetas ya existentes, por tal motivo se eligió la TMS320C6711, la cual es la mas parecida a la que actualmente se está trabajando, con la única diferencia que esta presenta mayores ventajas y soporte a la hora de implementar las práctica, como por ejemplo mayor espacio de memoria de programa, mas velocidad de procesamiento, y otras herramientas adicionales que vienen incluidas con el paquete, como por ejemplo, el compilador de C, muy útil a la hora de hacer programas en lenguaje C para después ensamblarlos en el DSK, todo esto a un precio de U.S\$395 con la opción de que esta incluida en el plan universitario para una mayor economía. A continuación están descritas todas las características de la nueva tarjeta para una mejor visión de la va a ser la nueva adquisición.

- Ciclo útil: 10nS
- Memoria: 32Kb de caché de datos, 32Kb de caché de programa, 512Kb caché L2.
- Interfase de memoria externa: 32 bits
- Puerto host: 16 bits
- Timers: (2) 32 bits

A continuación hay una imagen de la tarjeta y seguido todos los accesorios que vienen con ella a la hora de la adquisición.

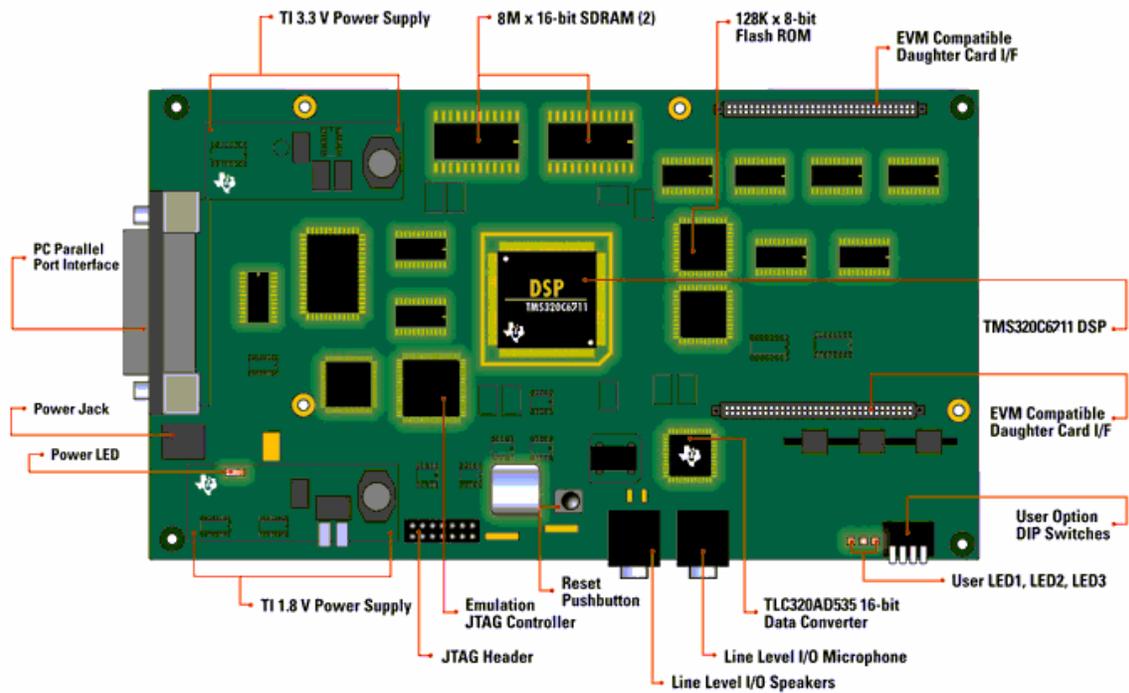


Figura 9. C6711 DSK.

- Tarjeta DSK para TMS320C6711 de fácil conexión al PC a través del puerto paralelo: 150MHz
- 16MB de SDRAM externa y 128KB de Flash externa, provee almacenamiento adicional de programa y datos.
- Conversores de datos de 16 bits TLC320AD535.
- Dispositivo de manejo de potencia, TPS56100

- Controlado JTAG, provee fácil simulación y debbuging.
- Interfase para tarjeta de expansión, dispuesta para el desarrollo de un sistema que necesite extensión.
- Fuente de poder universal para DSK
- Software: el DSK viene con todas las herramientas para ensamblar y ejecutar, así como un compilador de C para realizar nuestros programas más fácilmente en ese lenguaje.

Debido a que el tamaño de la tarjeta no se diferencia mucho con la actual, no es necesario hacer muchas modificaciones al modulo, este puede seguir con las especificaciones hasta ahora planteadas, así como la fuente de poder que puede ser utilizada la misma, además de no olvidar los conectores para los dos tipos de cables, BNC, para osciloscopio y generador de señales y los RCA para entrada y salida de audio.

Es recomendable sin embargo al necesitar información adicional, buscar en la pagina WEB de Texas Instruments, cuyo vinculo directo es el siguiente:

<http://www.ti.com/sc/docs/general/dsp/programs/index.htm>

Y el del programa universitario:

[http://www.dspvillage.com/docs/catalog/general/general.jhtml?templateId=5121&path=templatedata/cm/vilorphan/data/univ\\_pricelist](http://www.dspvillage.com/docs/catalog/general/general.jhtml?templateId=5121&path=templatedata/cm/vilorphan/data/univ_pricelist)

O en otro caso la persona que pueden contactar: **Mr. Plinio Naranjo, e-mail: [plintec@multi.net.co](mailto:plintec@multi.net.co)**

En el CD auxiliar del laboratorio hemos hecho disponible un proyecto para experiencias de laboratorio desarrollado por el departamento de física de la Universidad de Patras, Grecia. Esta ubicado en la carpeta *proyecciones del laboratorio*, se encuentra un paper en formato pdf (DSP2002 2003.pdf) donde se bosquejan las características del proyecto, otro documento pdf (DSPLab.pdf) donde se encuentran las guías, y finalmente un archivo comprimido donde están todos los códigos utilizados escritos en lenguaje c (listos para compilar). Estas experiencias utilizan el TMS320C6211, EL CUAL ES LA VERSIÓN EN PUNTO FIJO DEL TMS320C6711, ambos procesadores utilizan la misma tarjeta madre y

los algoritmos suministrados por la universidad son compatibles con el TMS320C6711.

Esperamos que la nueva tarjeta sea de total agrado y que las nuevas mejoras puedan ser utilizadas para la implantación de prácticas más avanzadas y que se puedan desarrollar sin muchas complicaciones por la nueva generación de estudiantes que aspire continuar nuestra labor.

## CONCLUSIONES

1. El tratamiento digital de señales moderno es fundamentalmente basado en software, gracias al rápido desarrollo y fabricación de procesadores digitales de señal, que permiten la implementación de sistemas y procesos en tiempo real y con elevada precisión, haciendo uso de herramientas de programación sofisticadas igualmente diseñadas para tal fin.
2. Los procesadores digitales de señal, acaparan la industria tanto así que los procesadores de propósito general (muy comerciales hoy día), ya poseen características propias de los DSP. No obstante se complementa entre sí, pero los DSPs siempre serán preferidos para aplicaciones donde se manejen variables análogas, que requieran amplios procesos y cálculos matemáticos, además del reducido uso de hardware que nos podría evitar.
3. A la hora de implementar procesos en tiempo real se requiere un conocimiento muy amplio de la teoría del tratamiento de señales en el dominio digital. Se debe conocer a profundidad el manejo correcto de

frecuencias, así como las especificaciones del DSP utilizado, cantidad de memoria, saber que rango de datos necesitamos para saber si el tamaño de la palabra de datos del procesador es el indicado.

4. Los procesadores digitales de señal son muy útiles a la hora de generar diversas formas de onda, ya que permiten el desarrollo de una señal con solo implementar una tabla con las coordenadas de los puntos, dicha tabla puede ser generada por medio de una función o una ecuación y luego los puntos son aproximados a la onda por medio de la unión de los mismos, es importante también tener un buen manejo de las expresiones recursivas.
5. Cualquier implementación en hardware debe tener en cuenta una precisión de conteo, puesto que la señal, como el sistema, son representados en forma binaria y tienen precisión finita, se debe tener en cuenta esto para evitar desbordamientos y minimizar los efectos no deseados en la cuantización. En la práctica numero 2, se pudieron apreciar dichos efectos.
6. La transformada rápida de Fourier es uno de los algoritmos más usados en DSP y puede ser usada en una gran variedad de tareas, prestándose por su eficiencia y rapidez a la hora de implementarlo a nivel de software. En la práctica de transformada rápida de Fourier es posible apreciar las diferencias de tiempo y precisión cuando se tienen transformadas de 1025,

512 y 256 puntos, además de tener la posibilidad de ver en un analizador de espectros las componentes de frecuencia de una señal cualquiera.

7. El ecualizador de sonido de parámetros modificables en tiempo real desarrollado y que se encuentra en la herramienta software DSKC3X 1.0, es quizás el ejemplo más representativo de lo ventajoso que es trabajar con DSP's. el efecto de los filtros que componen el ecualizador sobre la señal, sería mucho más rápido y eficiente con procesadores de mayor desempeño y con PC's auxiliares de gran capacidad de memoria RAM.
8. En la práctica de filtros FIR e IIR se confrontaron los efectos de simulación que aunque recalcan el poder de trabajo de MATLAB, todavía nos afianza aun más la superioridad para propósitos académicos de un procesador digital de señales, pues los sistemas actuales requieren tratamientos en tiempo real y personal que entienda y comprenda el comportamiento de señales en el dominio digital.
9. Sistemas de seguridad modernos como el reconocimiento de voz, de huellas digitales, entre otros están utilizando el procesamiento digital de señales como una herramienta poderosa a la hora de descomponer señales y analizar sus características. En la práctica de tratamiento de sonido, no solo vimos el reconocimiento de voz, sino también que es posible la síntesis

de sonido por medio de procesos digitales de manejo de frecuencias, y que además se puede precisar de que tipo de hablante se trata por la tonalidad de la voz y sus componentes de frecuencia.

## REFERENCIAS BIBLIOGRÁFICAS

Chassaing Rulph. Digital Signal Processing Laboratory Experiments Using C and the TMS320C31 DSK (Topics in Digital Signal Processing). Hardcover.1999.

Proakis John, Manolakis Dimitris. Tratamiento Digital De Señales Principios Algoritmos Y Aplicaciones. Editorial Prentice may.

Phillips, C.L. and Parr, J.M. (1995). Signals systems and transforms. Prentice Hall. New jersey.

Oppenheim, A.V. And Willsky, A.S. Señales Y Sistemas. Segunda Edición En Español. Editorial Prentice May Hispanoamericana S.A., Mexico.

Smith Steven. The scientist and Engineer's Guide to Digital Signal processing. California Technical Publishing. 1997.

Texas Instruments. TMS320C3X DSP Starter Kit User's Guide.

Paper: Armando Barreto. FIU (Florida International University). Low Cost personal DSP Training Station Based on the TI C3x DSK. 2000.

Paper: NAIS. National Spice. DSP development. Pags 7 may 1999.

Theory and application of digital signal processing. Rabiner gold Prentice Hall, 1975.

Introduction To Signal Processing. Sophocles Orfanidis, Prentice Hall.

Toolbox Signal Processing Matlab 5.0.

## Anexo A

### LISTA DE FIGURAS

Figura 1. Conversión analógica digital	8
Figura 2. Modelo de cuantización	8
Figura 3. $n$ niveles de cuantización.	9
Figura 4 Diagrama Filtro Adaptativo	15
Figura 5. Identificación.	
Figura 6. Modelización Inversa.	17
Figura 7. Predicción	18
Figura 8. Eliminación de Interferencia	18
Figura 9. Tarjeta C6711 DSK	27

## Anexo B

### GLOSARIO

**D.S.P.:** de las siglas en ingles Digital Signal Processing, procesamiento digital de señales.

**FFT:** de las siglas en ingles Fast Fourier Transform. Transformada rápida de Fourier.

**Interface:** canal de comunicación entre dos dispositivos. Puede ser lógica o físico.

**FIR:** de las siglas en ingles Finite Impulse Response. Respuesta finita al impulso.

**IIR:** de las siglas en ingles Infinite Impulse Response. Respuesta infinita al impulso.

**Señal Digital:** señal que solo puede representarse en dos estados (1 ó 0).

**Conversión análoga-digital:** es el proceso para llevar una señal del dominio análogo al dominio digital

**Conversión digital-análoga:** es el proceso de llevar una señal digital a su estado análogo.

**LABORATORIO DE TRATAMIENTO DIGITAL DE SEÑALES**

**GUÍA DE PRÁCTICAS DEL ESTUDIANTE**

**FRANCISCO JAVIER OSORIO MARTÍNEZ  
JHONNY MARK VELA SALCEDO**

**CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLIVAR**

**FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA**

**CARTAGENA DE INDIAS, D. T. H. Y C.**

**2003**

## CONTENIDO

INTRODUCCION.....	44
REGLAMENTO DEL LABORATORIO.....	49
PRACTICAS	
1. Introducción al manejo del módulo de prácticas.....	53
y Generación de onda seno	
2. Efectos de cuantización de señal.....	61
3. Transformada rápida de fourier.....	67
4. Implementación de filtros FIR.....	72
5. Implementación de filtros IIR.....	78
6. Implementación de filtros adaptativos.....	84
7. Tratamiento de sonido.....	90

## INTRODUCCION

En el transcurso de la carrera no hemos dado cuenta que el estudiante siempre se interesa en implementar y aplicar los conocimientos adquiridos en la materia en ejemplos de aplicación de la vida real y en el caso del DSP desarrollar sistemas en tiempo real, para poner en practica la teoría. El propósito de estas practicas será el de implementar en hardware la teoría que anteriormente solo se simulaba y desarrollar en el estudiante la habilidad de diseño en sistemas de procesamiento de señales.

De todas las cosas que el estudiante hace en el diseño, la única cosa que no se había hecho antes era la implementación de DSP en hardware. Parte de la preparación de implementar sistemas en tiempo real consiste en aprender a utilizar el hardware y las herramientas de software, para facilitar esto tenemos dos prácticas de introducción y seis más de experimentos para que el estudiante se vaya familiarizando con dichas herramientas y aprenda a utilizar los algoritmos básicos.

Las primeras prácticas de introducción enseñan a utilizar el assembler, el debugger y como correr programas en el DSK. En las siguientes experiencias el estudiante aplica algoritmos de procesamiento de señales tales como filtros FIR,

filtros IIR, filtros adaptativo y FFT. También veremos experimentos de precisión, cuantización y generación de señales, así como tratamiento de sonido.

Cada una de las prácticas estará concatenada con la materia de acuerdo al programa de la asignatura,. La siguiente es la forma como consideramos más apropiada, para el seguimiento de la materia con las prácticas:

**Generalidades: Señales y sistemas discretos.**

*Práctica 1*

**Representación digital de señales:**

**Representación espectral de las señales.**

*Práctica 2*

**Algoritmos rápidos para el cálculo de espectros.**

*Práctica 3*

**Sistemas discretos: Implementación software de sistemas filtros digitales lineales. Aplicaciones a audio y video**

*Práctica 4*

*Práctica 5*

## **Filtrado no lineal: Algoritmos básicos**

### *Práctica 6*

## **Señales y sistemas multidimensionales tratamiento de imágenes**

### **Ejemplos de aplicación: Síntesis y reconocimiento de voz**

### *Práctica 7*

Hay tres consideraciones importantes en el de experimentos. La primera es hacer que el estudiante aprenda haciendo, esto se logra haciendo que el estudiante aprenda las habilidades necesarias haciendo todo con sus propias manos. Segundo, el énfasis de este laboratorio es en el procesamiento de señales, no en la programación, si el estudiante esta interesado necesitaría conocimientos básicos de programación en C o en lenguaje ensamblador, por otro lado el estudiante tendrá acceso a los algoritmos de los experimentos escritos en lenguaje ensamblador, Todos los programas estarán en una carpeta en la unidad C desde la cual trabajaremos y en la que también estarán ubicadas las herramientas a utilizar. La tercera consideración es que debido a que la materia solo es de un semestre y que el énfasis no es en programación, muchas subrutinas hechas para la simulación en MATLAB serán provistas en las practicas para que el estudiante las utilice ya sea transcribiéndolas o copiándolas desde la carpeta de trabajo donde también estarán ubicadas. Las subrutinas necesarias

para acceder al conversor A/D y D/A y para la FFT también serán provistas por nosotros.

La descripción de cada experimento será hecha de manera concisa de forma que la guía va llevando al estudiante a través de toda la práctica de forma que tengan una idea clara de que es lo que se va a hacer en el proyecto. Esto será un procedimiento gradual, empezando por experimentos simples hasta llegar a unos mas complejos, en los dos primeros las descripciones serán muy específicas, dando los comandos necesarios y las instrucciones a utilizar mientras el estudiante se acostumbra a ellas, luego simplemente se dará la instrucción y el estudiante seguirá los pasos de acuerdo a lo que ha aprendido. También se recomienda leer teoría de otras fuentes y hacer comparaciones, además de leer el manual de operación y mantenimiento del TMS320C3X DSK y sus herramientas antes de empezar el laboratorio.

El equipo que se suministrará en el laboratorio será:

- PC's
- Módulo DSK
- Osciloscopio
- Generador de señales
- Cables y conectores

El software disponible será:

- MATLAB
- Assembler para TMS320C3X bajo DOS
- Debugger para TMS320C3X bajo DOS
- Software C3XDSK 1.0 bajo Windows
- Aplicaciones Anexas

## REGLAMENTO DE LOS LABORATORIOS

1. Se entenderá por **PRÁCTICA DE LABORATORIO** no sólo la ejecución, sino resultado y conclusiones a que diere lugar cada una de ellas, en consecuencia y para la realización de informes y evaluación se tendrán en cuenta estos factores.
2. En el primer día de clase el profesor procederá a distribuir a los estudiantes debidamente matriculados en subgrupos, cada uno de los cuales tendrá un código o número distintivo. El número de subgrupos permitirá al estudiante conocer previamente que experimento o práctica debe realizar en la sección de laboratorio semana por semana.
3. Los laboratorios inician actividades a la hora señalada en el horario, efectuando el quiz semanal.
4. Una vez presentado el quiz cada subgrupo recibirá del auxiliar del laboratorio el equipo necesario para efectuar La correspondiente práctica. Los estudiantes deben verificar confrontando los elementos y la lista. Cualquier observación debe hacerse inmediatamente al profesor, puesto

que una vez iniciada la práctica todo daño o pérdida corre por cuenta del subgrupo.

5. Verificado el equipo se anotan claramente los nombres de los estudiantes, grupo, subgrupo, hora y fecha de la realización del experimento. Una vez firmada la hoja del equipo ésta será recogida por el auxiliar.
6. El material o equipo de laboratorio se devolverá durante los 30 minutos finales del tiempo estipulado para cada práctica semanal, una vez realizada la parte experimental exigida.
7. Se aclara que el equipo será recibido por el auxiliar mediante inventario físico con el fin de establecer faltantes. Se realizará además un chequeo para determinar si hubo daño en los aparatos utilizados.
8. En caso de daño por negligencia comprobada, o en caso de pérdida de algún elemento se deberá reponer o pagar además de afectarse la calificación de la respectiva práctica.
9. Con ayuda del manual de laboratorio el estudiante deberá prepararse, de tal que manera pueda ejecutar el experimento por si solo.

10. La evaluación de esta preparación se llevará a cabo por medio de quices semanales y el control del trabajo de cada estudiante durante la práctica mediante preguntas pertinentes sobre los principios físicos, el montaje o instrumentación utilizada. Si se considera que el estudiante está desarrollando la práctica incorrectamente el profesor podrá exigir la repetición de la misma.
11. Durante la práctica el profesor y el monitor del Laboratorio supervisarán el desempeño de cada estudiante. A juicio del profesor se prestará ayuda, si el grado de dificultad en algún aspecto de la práctica lo requiere.
12. Se realizará un informe por cada subgrupo, el cual se consignará en una libreta cuadriculada tamaño carta y será elaborado en su totalidad a tinta. Las gráficas se elaborarán en papel milimetrado recortado al tamaño de la hoja. Cada informe deberá llevar el nombre de los estudiantes que asistieron y el día de realización de la práctica.
13. El tiempo estipulado para la realización de la práctica es de máximo una hora y 45 minutos, al final del cual se suspende la actividad y se procede a la elaboración del informe.
14. Durante la elaboración del informe no está permitido el intercambio de

información entre subgrupos.

15. Para la calificación del estudiante se tendrán en cuenta los siguientes aspectos:

- Quiz semanal
- Desarrollo de la práctica
- Informe

16. En el desarrollo de la práctica, además del conocimiento y preparación de la misma, se evaluarán otros factores tales como: trabajo en grupo, destreza y habilidad para resolver problemas y orden en el laboratorio.

17. El contenido de cada informe debe ser como mínimo el siguiente:

- Título de la práctica
- Integrantes del subgrupo
- Datos tabulados y características de los instrumentos utilizados
- Cálculos tipo incluyendo el cálculo de errores.
- Resultados tabulados y/o gráficas.
- Análisis e interpretación de gráficas y/o resultados.
- Conclusiones.

# 1. INTRODUCCION AL MANEJO DEL MODULO DE PRACTICAS Y GENERACION DE SEÑALES

## OBJETIVOS

Objetivo General:

El propósito de es aprender a usar las herramientas del software DSK y ejecutar un programa de ejemplo en el DSK.

Objetivos Específicos:

- Aprender a usar el assembler/linker **dsk3a**.
- Aprender a usar el debugger, **dsk3d**, para cargar y ejecutar programas y a la vez ver valores de registros de memoria
- Aplicar la herramienta generador del programa C3XDSK 1.0 y analizar las señales producidas, mediante el osciloscopio.

## **TEMAS DE CONSULTA**

DSK Hardware Interface, DSK Communications Kernel, TLC32040 AIC Initialization, Host Software.

## **BIBLIOGRAFÍA**

TMS320C3x DSP Starter Kit.pdf. Manual de operación y mantenimiento

## **PRELIMINARES**

El procedimiento a realizar en esta práctica será el de ensamblar un archivo .asm utilizando las herramientas dsk3a.exe, y el c3xdsk 1.0, generando un archivo .dsk que se ejecutará para corroborar su funcionamiento y el del modulo.

## **PREGUNTAS DE REPASO**

1. Explique brevemente en que consiste el muestreo y la cuantización.
2. Explique el teorema del muestreo.

3. ¿Que es la frecuencia de Nyquist?
4. ¿Porque se produce el aliasing?
5. En general, ¿cuales son las condiciones ideales para una buena cuantización?

## **EQUIPO**

- Osciloscopio.
- (1) Cable BNC-BNC.
- PC con todo el software y archivos previamente instalados.
- Parlante del módulo.
- Cable RCA-JACK.

## **PROCEDIMIENTO**

1. Conecte el módulo del DSK a la alimentación y al puerto paralelo a través del cable db25.
2. Encienda el módulo.

3. Conecte el osciloscopio a la salida del módulo, a través del cable BNC-BNC.
4. Conecte el cable RCA del parlante a la salida RCA del módulo.
5. Inserte el CD-ROM destinado para el laboratorio, copie la carpeta DSP a la unidad C del computador.
6. Ejecute el archivo C3XDSK.EXE. Este archivo extraerá el programa C3XDSK 1.0 a la unidad C, en una carpeta con el mismo nombre.
7. Abra la ventana de comandos de DOS y vaya al directorio donde se encuentran los archivos a ser utilizados, en este caso trabajaremos en el directorio >>C:\DSP\ allí han sido localizados todos los programas que vamos a utilizar durante el curso. Para esta experiencia cerciorese que el archivo **minuet.txt** se encuentre en la carpeta DSP.
8. Ejecute desde la ventana de DOS la siguiente instrucción: **C:\DSP\dsk3a demo1d.asm** con esto ya habremos ensamblado el archivo, creándose uno ejecutable con el nombre de **demo1d.dsk**.

9. Ejecute la siguiente instrucción: **>>C:\DSP\dsk3d reset** después de esto se debe ver al siguiente ventana:

DISASSEMBLY		C31 DSP STARTERS KIT	
809c03	50700080 start	LDIU	00080h, DP
809c04	08349c2c	LDI	@09c2cH, SP
809c05	07608000	LDF	0.000000e+00, R0
809c06	c610c1c0	LDI	*AR0, R0    LDI *AR
809c07	c610c1c0	LDI	*AR0, R0    LDI *AR
809c08	08600100	LDI	256, R0
809c09	09a09c00	LSH	@09c00H, R0
809c0a	61809c0e	BRD	jump
809c0b	07618000	LDF	0.000000e+00, R1
809c0c	07628000	LDF	0.000000e+00, R2
809c0d	07630000	LDF	1.000000e+00, R3
809c0e	07640000	jump	LDF 1.000000e+00, R4
809c0f	087b0003	loop	LDI 3, RC
809c10	64809c1a	RPTB	block
809c11	02640001	ADDI	1, R4

COMMAND		MEMORY	
Texas Instruments 1994		809800	00000007 ffffffff 00809802 00809827
load testa		809804	0080982c 00809839 0080983c 0080983f
		809808	00809843 00809842 00809868 0080989a
		80980c	008098a9 10800000 0f350000 0f300000
		809810	0f200000 0f320000 0f280000 0f290000
		809814	1a770004 6a050006 628098a9 50700080

figura 1.1

En la subventana inferior izquierda (ventana de comandos) ejecute la siguiente instrucción: **load demo1d.dsk** esto hace que el programa se cargue en la memoria del DSP, ahora presione F5 para ejecutarlo.

10. Observe la salida en el osciloscopio y escúchela en el parlante. Para cancelar la ejecución del programa presione cualquier tecla, luego de esto ejecute la instrucción **reset** y después **exit** para salir del programa.

11. Copie el archivo demo1d.asm y minuet.txt en la carpeta C3XDSK. Ejecute el archivo c3xdsk.exe ubicado en la misma carpeta. En la barra de menús haga click en la opción *CMP*. Elija demo1d.asm y haga doble click sobre el. Inmediatamente se ensamblará (debe aparecer una ventana temporal durante el ensamble, que luego se cierra).
12. Vaya al menú opciones del software y seleccione modo conectado.
13. En el menú archivo elija cargar archivo demo1d.dsk para cargarlo al módulo y ejecutarlo. Corrobore su efecto en el osciloscopio y escúchelo en el parlante.
14. Para finalizar el algoritmo, en el menú *comandos* seleccione la opción *reiniciar*. y haga clic en *salir*.

## GENERACIÓN DE SEÑALES

1. Vaya al menú *opciones* del software y seleccione *modo conectado*.

2. Haga clic sobre la opción generador. Aparecerá un cuadro con posibilidades para generar formas de onda triangulares, cuadradas y senoidales y de establecer su amplitud y frecuencia.
  
3. Llene la siguiente tabla de datos, donde la frecuencia ajustada es la que se ha de colocar en el recuadro frecuencia de la herramienta *generador*, ajuste la amplitud en su máximo valor, pulsando el botón seno y después procesar por cada frecuencia ajustada (el voltaje de offset manténgalo en 0 V).

Frecuencia							
Ajustada	100Hz	500Hz	1.0KHz	3.0KHz	6.5KHz	7KHz	9KHz
Medida							
Error							

F. ajustada	100Hz	500Hz	1.0KHz	3.0KHz	6.5KHz	9KHz
Voltaje pico						

Tabla 1.1

4. Para salir del programa, En el menú *comandos* seleccione la opción *reiniciar*. y haga clic en *salir*

## PREGUNTAS SOBRE EL DESARROLLO DEL LABORATORIO

1. ¿De acuerdo a lo observado al ejecutar el archivo **demo1d.dsk** de la primera parte de la experiencia, ¿Cómo cree que se pudo haber generado esa secuencia de sonido?
2. ¿De acuerdo al cuadro de dialogo mostrado por al aplicación al generar la señal, porque cree usted que el rango máximo para generación no puede ser mayor de 9999 Hz.?
3. ¿Según las medidas a partir de que valor de frecuencia la señal se vuelve realmente despreciable? ¿Por qué ?
4. Con la aplicación genera una onda triangular y una cuadrada y experimente con varias frecuencias entre 5000 Hz y 9999 Hz. ¿Qué efecto puede apreciar? ¿Porque cree usted que sucede esto?

## CONCLUSIONES Y OBSERVACIONES

## 2. EFECTOS DE CUANTIZACION DE SEÑAL

### OBJETIVOS

Objetivo General:

El propósito de esta práctica es observar los efectos de cuantización y redondeo en una señal.

Objetivos Específicos:

- Conocer la relación entre los bits que son usados para representar una señal y los que no.
- Conocer en que afecta la implementación en el desarrollo del sistema.
- Conocer en que afecta la precisión en el desarrollo del sistema.

## **TEMAS DE CONSULTA**

Teorema de muestreo y cuantización, conversión análoga-digital y digital-análoga, efecto aliasing.

## **BIBLIOGRAFIA**

Proakis John, Manolakis Dimitris. Tratamiento Digital De Señales Principios Algoritmos Y Aplicaciones. Editorial Prentice may.

## **PRELIMINARES**

El procedimiento a realizar en esta práctica será el de introducir una señal seno por la entrada análoga del AIC y luego llevarla a la salida por medio de un programa, en dicha salida se mirará la señal resultante por medio del osciloscopio después de haberse hecho la cuantización y reconstrucción de la señal y se comparará con la señal de entrada. Al final se vera un ejemplo ilustrativo con una señal de audio.

## PREGUNTAS DE REPASO

1. Explique brevemente en que consiste el muestreo y la cuantización.
2. Explique el teorema del muestreo.
3. ¿Que es la frecuencia de Nyquist?
4. ¿Porque se produce el aliasing?
5. En general, ¿cuales son las condiciones ideales para una buena cuantización?

## EQUIPO

- Osciloscopio.
- Generador de señales.
- (3) Cables BNC-BNC.
- (1) T-BNC.
- PC con todo el software y archivos previamente instalados.
- Parlante del módulo.
- CD de audio.
- Cable RCA-JACK.

## PROCEDIMIENTO

1. Conecte el modulo del DSK a la alimentación y al PC a través del puerto paralelo.
2. Conecte un generador de señales y una de las entradas del osciloscopio a la entrada análoga del módulo y la otra entrada del osciloscopio a la salida del módulo. Utilice cables BNC-BNC para las conexiones y para conectar el osciloscopio y generador a la entrada del módulo a la vez, utilice una T-BNC.
3. Encienda el módulo.
4. Copie el archivo **loopaic.asm** ubicado en la carpeta DSP, a la carpeta C3xdsk y haga clic en la opción CMP (compilar), escoja el archivo y haga doble-click sobre el, inmediatamente se compilará. Generando el archivo **loopaic.dsk**.
5. En el menú *Archivo* seleccione *Cargar Archivo DSK* y ábralo. Quedará cargado en la tarjeta.

6. Introduzca una señal seno por medio del generado de señales a través de la entrada análoga, con una amplitud no mayor a 1V-pico, establézcale una frecuencia de aproximadamente 1.0KHz y conecte una de las entradas del osciloscopio a esta señal para observarla, la otra entrada conéctela a la salida análoga del módulo para poder comparar las dos señales simultáneamente. Si utiliza un osciloscopio análogo colóquelo en modo Both y Alt., en el recuadro de configuración vertical.
  
7. Ahora varíe la frecuencia y observe que cambio puede tener en la salida. Tome datos de atenuación de la señal de salida con respecto a la entrada, cambio de fase y amplitud.

Medición	1	2	3	4	5	6	7
Frecuencia	1 KHz						
Amplitud	1Vp						
Fase							
Atenuación							

Tabla 2.1

8. Detenga la ejecución del programa y desconecte todos los cables BNC del modulo provenientes del generador de señales y el osciloscopio.

9. Ahora conecte la entrada RCA del módulo a la unidad de CD por medio del cable RCA-JACK y conecte el parlante a la salida RCA y podrá escuchar el sonido.

10. Para salir del programa, En el menú *comandos* seleccione la opción *reiniciar.* y haga clic en *salir*

## **PREGUNTAS SOBRE EL DESARROLLO DEL LABORATORIO**

1. ¿Que atenuación proporciona el programa *loopaic.dsk* a la señal de entrada?
2. ¿Cual es la amplitud máxima que se puede obtener a la salida del DSK antes que se recorte la señal?
3. ¿A que frecuencia se pierde significativamente la señal en casi 70% de la amplitud promedio?
4. ¿Que produce las limitaciones de frecuencia en el DSK?

## **CONCLUSIONES Y OBSERVACIONES**

### **3. TRANSFORMADA RAPIDA DE FOURIER**

#### **OBJETIVOS**

Objetivo General:

El propósito de esta práctica es aplicar la transformada rápida de fourier en tiempo real.

Objetivos Específicos:

- Implementar la transformada rápida de Fourier en tiempo real.
- Entender el algoritmo de FFT.

#### **TEMAS DE CONSULTA**

Transformada de Fourier. Transformada discreta de Fourier . FFT

#### **BIBLIOGRAFIA**

Proakis John, Manolakis Dimitris. Tratamiento Digital De Señales Principios Algoritmos Y Aplicaciones. Editorial Prentice Hall. Curso DSP incluido en el CD.

## PRELIMINARES

Se realizará una simulación en MATLAB, un sencillo algoritmo de ejemplo en el cual aplicamos la Transformada Rápida de Fourier (FFT) a una señal exponencial modulada en amplitud. Como puede ver la transformada se hace ejecutando la instrucción **fft ()** y los demás son solo pasos para graficar. Luego se analizará el efecto de una FFT de 256 en la reconstrucción de una señal cuadrada.

## PREGUNTAS DE REPASO

1. ¿Explique en que consiste la DFT?
2. ¿Para que se usa la FFT?

## EQUIPO

- Osciloscopio.
- Generador de señales.
- (3) Cables BNC-BNC.
- (1) T-BNC.
- PC con todo el software y archivos previamente instalados.
- Parlante del módulo.
- CD de audio.
- Cable RCA-JACK.

## PROCEDIMIENTO

### SIMULACION

1. Ubique el archivo `fftej3.m` ubicado en la carpeta DSP en la carpeta de trabajo de MATLAB (para la versión 5.0 o superior, la carpeta se llama `work`, ubicado en al carpeta `MATLABR11`).
2. En la consola de comandos de MATLAB, digite: `fftej3(128,0.2)` y presione ENTER.
3. Observe las graficas y saque conclusiones.
4. Cierre la ventana de MATLAB.

### IMPLEMENTACION EN TIEMPO REAL

1. Conecte un generador de señales y una de las entradas del osciloscopio a la entrada análoga del módulo y la otra entrada del osciloscopio a la salida del módulo. Utilice cables BNC-BNC para las conexiones y para conectar el osciloscopio y generador a la entrada del módulo a la vez, utilice una T-BNC.

2. Encienda el modulo.
3. Ejecute el software C3XDSK 1.0, vaya al menú *opciones* y seleccione la opción *modo conectado*.
4. Ajuste el generador de señales de tal forma que genere una onda cuadrada con una amplitud no mayor de 1V-pico y frecuencia 1 KHz.
6. Haga clic sobre el menú *FFT* para ejecutar la transformada (dicha transformada está hecha para 256 puntos).
7. Determine la relación entre la amplitud, frecuencia y forma de la señal de entrada y la de salida.
8. Varíe lentamente la frecuencia en la entrada y observe el cambio a la salida.
9. Para salir del programa, En el menú *comandos* seleccione la opción *reiniciar*. y haga clic en *salir*

## **PREGUNTAS SOBRE EL DESARROLLO DEL LABORATORIO**

1. Explique a que se refieren los parámetros 128 y 0.2 ingresados para la ejecución del algoritmo ffej3.
2. ¿Qué representa las graficas obtenidas en MATLAB?
3. ¿Según la experiencia de los items 4 y 5 que tratamiento realiza la FFT sobre dicha señal?
4. ¿Qué diferencia significativa encontró en la reconstrucción en la onda cuadrada?

## **CONCLUSIONES Y OBSERVACIONES**

## **4. IMPLEMENTACION DE FILTROS FIR**

### **OBJETIVOS**

Objetivo General:

El objetivo de este proyecto es implementar un filtro FIR en tiempo real.

Objetivos Específicos:

- Simular un filtro FIR con la ayuda de MATLAB.
- Implementar un filtro FIR en tiempo real utilizando el DSP y la herramienta FIR del software C3XDSK 1.0
- Modificar algunos parámetros del filtro con la herramienta y analizar los cambios en la señal de prueba.

### **TEMAS DE CONSULTA**

Filtros digitales. Filtros FIR. Características de los filtros FIR.

## **BIBLIOGRAFIA**

Proakis John, Manolakis Dimitris. Tratamiento Digital De Señales Principios Algoritmos Y Aplicaciones. Editorial Prentice Hall. Curso DSP incluido en el CD.

## **PRELIMINARES**

Aplicaremos los conceptos estudiados de un filtro FIR para hacer primero una simulación en MATLAB y luego implementarlo en tiempo real a través del módulo del DSK.

## **PREGUNTAS DE REPASO**

1. ¿Qué es un filtro digital?
2. Enumere los pasos necesarios para el diseño de un filtro digital
3. Enumere tres características de los filtros FIR.
4. Diga dos ventajas de los filtros FIR con respecto a los dos filtros.

## **EQUIPO**

- Osciloscopio.
- (3) Cable BNC-BNC.
- (1) conector en T tipo BNC
- PC con todo el software y archivos previamente instalados.
- Parlante del módulo.
- CD de audio
- Cable RCA-JACK.

## **PROCEDIMIENTO**

1. Abra el programa MATLAB y ejecute el archivo FIR.m.
2. Para el tipo de filtro a diseñar ingrese lp (pasabajos), para el numero de puntos 256. en frecuencia de corte normalizada 0.1 y finalmente ingrese hamming, para el tipo de ventana. El programa debe arrojar la respuesta en frecuencia (diagrama de bode) del filtro. Mantenga las graficas abiertas.

3. Conecte el modulo del DSK a la alimentación y al PC a través del puerto paralelo.
4. Conecte un generador de señales y una de las entradas del osciloscopio a la entrada análoga del módulo y la otra entrada del osciloscopio a la salida del módulo. Utilice cables BNC-BNC para las conexiones y para conectar el osciloscopio y generador a la entrada del módulo a la vez, utilice el conector en T tipo BNC.
5. Con el generador de señales alimente el modulo con una onda seno de 1 Vp.
6. Abra el programa C3XDSK 1.0. Vaya al menú *opciones* y seleccione la opción *modo conectado*, luego haga click sobre el menú *IIR*, después de esto se abrirá una ventana donde especificará el tipo de filtro y sus características. Especifique un filtro de orden 15, pasabajo tipo butterworth (el método que usa la aplicación para implementar el filtro es el de transformación bilineal) y haga clic en aceptar.
7. En la aplicación *FIR típicos*, elija filtro pasabajos, frecuencia de corte 1 KHz también, ventana Hamming y coeficiente 256. deje los demás valores por defecto. seleccione *aceptar* y observe la salida a través del osciloscopio.

8. En el generador Varíe la frecuencia alrededor de 1KHz. Tome valores de frecuencia para atenuaciones de 50%, 70%, 90% y 95%. Llene la tabla.

Atenuación	50%	70%	90%	95%
Frecuencia				

Tabla 4.1

9. Vaya nuevamente a la aplicación FIR típicos y ajuste la frecuencia de corte a 9000 Hz, haga click en aceptar y varíe la frecuencia de la onda seno. Mida el valor de frecuencia para el cual la señal se atenúa en un 70% y anótelo.
10. Vaya al menú *Comandos* y seleccione *Reiniciar*.
11. Conecte la unidad de CD a la entrada del módulo a través del cable RCA – JACK y el parlante a la salida RCA. Ejecute un CD de audio.
12. Vaya el menú *FIR* y seleccione *ecualizador FIR*, con esta aplicación podrá suprimir, atenuar o acentuar componentes de frecuencia, efecto que podrá apreciar de manera auditiva.

## **PREGUNTAS SOBRE EL DESARROLLO DEL LABORATORIO**

1. ¿Qué significado tiene la grafica producto de la ejecución del archivo FIR.m en MATLAB? ¿Para que se utilizan?
2. ¿por qué se utiliza una frecuencia normalizada de 0.1? ¿Cómo se obtiene?
3. De acuerdo a la grafica magnitud contra frecuencia arrojado por MATLAB. ¿Cuál es el valor de la frecuencia de corte?
4. ¿Coincide esta frecuencia de corte con la obtenida en tiempo real con el DSP? ¿Cuál es el valor?
5. ¿Coincide la frecuencia de corte ajustada (9000 Hz) en el numeral 9 del procedimiento con la obtenida? Explique el resultado.

## **CONCLUSIONES Y OBSERVACIONES**

## 5. IMPLEMENTACION DE FILTROS IIR

### OBJETIVOS

Objetivo General:

El objetivo de este proyecto es implementar un filtro IIR en tiempo real.

Objetivos Específicos:

- Simular un filtro IIR con la ayuda de MATLAB.
- Implementar un filtro IIR en tiempo real utilizando el DSP y la herramienta IIR del software C3XDSK 1.0
- Modificar algunos parámetros del filtro con la herramienta y analizar los cambios en una señal de prueba.

## **TEMAS DE CONSULTA**

Filtros digitales. Filtros IIR. Características de los filtros IIR.

## **BIBLIOGRAFÍA**

Proakis John, Manolakis Dimitris. Tratamiento Digital De Señales Principios Algoritmos Y Aplicaciones. Editorial Prentice Hall. Curso DSP incluido en el CD.

## **PRELIMINARES**

Aplicaremos los conceptos estudiados de un filtro IIR para hacer primero una simulación en MATLAB y luego implementarlo en tiempo real a través del módulo del DSK.

## **PREGUNTAS DE REPASO**

1. Enumere tres características de los filtros IIR.
2. diga dos ventajas de los filtros IIR con respecto a los otros filtros.
3. mencione dos aplicaciones de los filtros

## EQUIPO

- Osciloscopio.
- (3) Cable BNC-BNC.
- (1) conector en T tipo BNC
- PC con todo el software y archivos previamente instalados.
- Parlante del módulo.
- CD de audio
- Cable RCA-JACK.

## PROCEDIMIENTO

1. Abra el programa MATLAB y ejecute el archivo IIR.m.
2. Para el orden del filtro ingrese 15 y en el tipo de filtro ingrese *butt*. Debe arrojar una grafica de polos y su respuesta en frecuencia (diagrama de bode).
3. Entre el tipo de conversión lp (pasabajos). Y en frecuencia de corte ingrese 1000 Hz.

4. Cuando le pida el método ingrese bil (bilineal) y en frecuencia de muestreo ingrese 19531 (ya que la frecuencia máxima de muestreo comprobada del DSK es de 19531Hz).
5. Conecte el modulo del DSK a la alimentación y al PC a través del puerto paralelo.
6. Conecte un generador de señales y una de las entradas del osciloscopio a la entrada análoga del módulo y la otra entrada del osciloscopio a la salida del módulo. Utilice cables BNC-BNC para las conexiones y para conectar el osciloscopio y generador a la entrada del módulo a la vez, utilice el conector en T tipo BNC.
7. Con el generador de señales alimente el modulo con una onda seno de 1 Vp.
8. Abra el programa C3XDSK 1.0. Vaya al menú *opciones* y seleccione la opción *modo conectado*, luego haga click sobre el menú *IIR*, después de esto se abrirá una ventana donde especificará el tipo de filtro y sus características. Especifique un filtro de orden 15, pasabajo tipo butterworth

(el método que usa la aplicación para implementar el filtro es el de transformación bilineal) y haga clic en aceptar.

9. En el generador varíe la frecuencia de la onda seno entre 100 Hz y 3 KHz. Mida el valor de frecuencia para el cual la señal se atenúa en un 70% y anótelo.

10. Compárelo con el obtenido en la grafica obtenida en MATLAB. Calcule el error relativo.

Frecuencia de corte de la simulación	Frecuencia obtenida en tiempo real	Error

Tabla 5.1

11. Nuevamente en el programa C3XDSK 1.0. luego haga click sobre el menú *IIR*, especifique un filtro de orden 1, pasabajo tipo elíptico y frecuencia de corte 1 KHz.

12. Varíe la frecuencia de la onda seno entre 100 Hz y 5 KHz. Mida el valor de frecuencia para el cual la señal se atenúa en un 70% y anótelo.

13. Repita el ítem 11 y 12 pero para un tipo de filtro Chebyshev.

14. Para salir de la aplicación seleccione menú comandos y reiniciar. Y luego haga clic en salir.

## **PREGUNTAS SOBRE EL DESARROLLO DEL LABORATORIO**

1. ¿De acuerdo al error calculado en el ítem 10 de la experiencia cree usted que el procesamiento en tiempo real es confiable? ¿justifique su respuesta?
2. ¿En que ayuda la simulación a la hora de implementar un filtro en tiempo real?
3. ¿Qué implica el utilizar un orden grande en los filtros IIR?
4. De acuerdo a los resultados obtenidos en los ítem 11, 12 y 13 de la experiencia. ¿Como afecta a la frecuencia de corte, la modificación del orden del filtro?. ¿Porque?
5. ¿Dónde encontró mejor resultado en el filtro IIR tipo elíptico o en el Chebyshev? ¿Concuerda esto con la teoría de filtros digitales? Justifique su respuesta.

## **CONCLUSIONES Y OBSERVACIONES**

## **6. IMPLEMENTACION DE UN FILTRO ADAPTATIVO**

Objetivo General:

El propósito de esta práctica es usar filtros adaptativos para la eliminación de una interferencia sinusoidal.

Objetivos Específicos:

- Analizar una simulación en MATLAB de una implementación de filtro adaptativo para eliminar una interferencia sinusoidal.
- Interpretar la operación del sistema de filtro adaptativo implementado en el modulo del DSP en tiempo real.

### **TEMAS DE CONSULTA**

Filtros digitales. Filtros FIR e IIR. Filtros adaptativos. Diseño de filtros adaptativos.

## **BIBLIOGRAFIA**

Proakis John, Manolakis Dimitris. Tratamiento Digital De Señales Principios Algoritmos Y Aplicaciones. Editorial Prentice Hall. Curso DSP incluido en el CD.

## **PRELIMINARES**

Nuestro objetivo será hacer primero una simulación en MATLAB, observar y concluir sobre los resultados obtenidos, para luego pasar a la implementación en tiempo real en el modulo del DSK.

## **PREGUNTAS DE REPASO**

6. Explique Brevemente en que consiste un filtro adaptativo
7. ¿Cuál es la función principal de los filtros adaptativo?
8. Mencione los cuatro tipos de filtros adaptativos.

## **EQUIPO**

- Osciloscopio.
- Generador de señales.
- (3) Cable BNC-BNC.
- (1) conector en T tipo BNC
- PC con todo el software y archivos previamente instalados.

## **PROCEDIMIENTO**

### **SIMULACION**

1. Analice el programa adapta.m, que ilustra la adaptación de un filtro FIR, de orden 32, en la eliminación de una interferencia, como se ilustra en la figura. En el inicio del proceso de adaptación el filtro FIR tendrá coeficientes nulos. Se usa como interferencia a eliminar una señal sinusoidal. Este mismo sistema será implementado en el DSK.

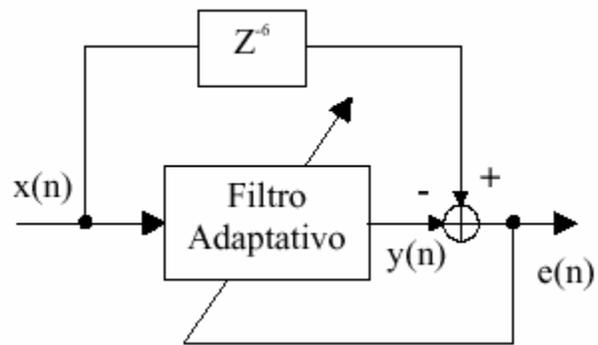


Figura 6.1

2. Concluya sobre la influencia que tiene el paso de adaptación en la frecuencia de la sinusoide de entrada en el proceso adaptativo.

## IMPLEMENTACION EN TIEMPO REAL

1. Conecte el módulo a la alimentación y al PC a través del cable paralelo.
2. Conecte un generador de señales y una de las entradas del osciloscopio a la entrada análoga del módulo y la otra entrada del osciloscopio a la salida del módulo. Utilice cables BNC-BNC para las conexiones y para conectar el

osciloscopio y generador a la entrada del módulo a la vez, utilice una T-BNC.

3. Encienda el módulo.
4. Analice el programa **adaptfir.asm** que implementa un filtro adaptativo, note que en el programa el paso de adaptación esta en 0.0.
5. Encienda el módulo del DSK y ensamble el programa **adaptfir.asm** ubicado en la carpeta DSP.
6. Luego cargue y ejecute el archivo ensamblado **adaptfir.dsk** en el software DSK 1.0 seleccionando la opción *Cargar archivo DSK* en el menú *Archivo*.
7. Aplique a la entrada una señal sinusoidal y visualice la salida con el osciloscopio, utilizando los cables apropiados para tal fin. Varíe la frecuencia entre 800Hz y la frecuencia de *Nyquist*.
8. Coloque el paso de adaptación en 1.0E-12, obtenga un nuevo ejecutable y córralo. Aplique a la entrada una senoide de 1KHz, luego varíe la frecuencia y observe como se comporta el sistema.

9. Repita el paso anterior para diferentes valores de paso de adaptación .
  
10. Salga de la aplicación seleccionando en el menú *comandos* la opción *reiniciar*. Y luego haga clic en *salir*.

## **PREGUNTAS SOBRE EL DESARROLLO DEL LABORATORIO**

1. De acuerdo a los tipos de filtros adaptativos estudiados, ¿cuál cree usted que fue el utilizado en esta experiencia?
  
2. Identifique en el diagrama de bloques del filtro empleado, las variables utilizadas en la simulación y en la implementación en tiempo real.
  
3. De acuerdo a lo observado en el ítem 7 de la experiencia, para un paso de adaptación nulo ¿a qué forma se reduce el sistema adaptativo?
  
4. ¿En qué influye en el que el paso de adaptación sea mayor o menor? ¿qué efecto produce?
  
5. Explique con sus palabras el efecto de este filtro, observado en el ítem 8

## **CONCLUSIONES Y OBSERVACIONES**

## 7. TRATAMIENTO DE SONIDO

### OBJETIVOS

Objetivo General:

El propósito de esta práctica es simular la generación de las vocales a través de un filtro “all pole” de 6 orden.

Objetivos Específicos:

- Analizar una simulación en ambiente MATLAB de una implementación de un filtro para la generación de las cinco vocales.
- Observar un programa de ejemplo, desarrollado para reconocer las vocales a través de un micrófono por medio del PC.

## **TEMAS DE CONSULTA**

Filtros IIR. Tratamiento de sonido y voz.

## **BIBLIOGRAFÍA**

The Scientist and Engineer's Guide to Digital Signal Processing by Steven W. Smith. (anexo en el CD).

## **PRELIMINARES**

Nuestro objetivo será hacer primero una simulación en MATLAB y observar y concluir sobre los resultados obtenidos, para luego pasar al programa de ejemplo de ciencia viva y así tener las bases para una implementación futura en tiempo real en el módulo del DSK.

## **PREGUNTAS DE REPASO**

1. ¿Qué es el espectro de una señal?
2. ¿Por qué es más fácil simular una vocal que una consonante?

## PROCEDIMIENTO

### SIMULACION

1. Analice el programa vocales.m que ilustra tanto analítica como gráficamente, en ambiente MATLAB, la generación de la vocal /a/ usando tres formantes. El valor de cada formante (F1, F2, F3) depende de la vocal que se pretende sintetizar, como consta la siguiente tabla. Se pretende, personalice a su realización usando los formantes correspondientes a su propia voz. Note que el tacto vocal es modelado por un filtro IIR “all-pole” de 6ª orden, que determina la envolvente espectral de los sonidos vocalizados (en el que se incluyen la señal de habla correspondiente a las vocales). El sonido vocalizado de cada vocal sintetizada, se genera de forma simple, excitando el filtro IIR con un tren de impulsos, a una frecuencia de cerca de 100, 200 o 300 por segundo, para hombres, mujeres y niños, respectivamente. Para nuestro caso usamos una frecuencia de 200 Hz. Como alternativa, podrá usar a frecuencia correspondiente a la tonalidad de su propia voz.

Vogal	F1 (Hz)	F2 (Hz)	F3 (Hz)
/à/	576	1202	2652
/é/	523	1467	3257
/i/	389	1794	3212
/ó/	502	930	2632
/u/	306	675	2508

Tabla 7.1

2. Modifique el programa vocales.m de modo que genere cada una de las cinco vocales y escuche el resultado. Para simplificar, no altere el radio de los polos asociados a cada formante, solo altere la frecuencia. Saque conclusiones.

## IMPLEMENTACION EN TIEMPO REAL

Ahora procederemos a implementar y analizar el comportamiento del reconocedor de vocales y tacto vocal de ciencia viva, para tener un principio de cómo es el reconocimiento de voz. Ejecútelo y compárelo con los resultados obtenidos en MATLAB.

1. Descomprima el archivo *Cviva.zip*.

2. Cambie la resolución de la pantalla a 1024 x 768 antes de ejecutar el programa para que pueda ver la totalidad de este, de lo contrario solo le saldrá parte del “pantallazo”
  
3. Ejecute el archivo *Project1*, en el encontrará varias partes, dos graficas con la amplitud y potencia espectral de la voz, un botón para seleccionar el tipo de visualización, la cual esta dividido en dos; *vogais*(vocales) y *harmónica*, en la opción *vogais*, no dará un reconocimiento de la vocal que pronunciamos por el micrófono y la opción *harmónica*, analizara las componentes de la voz para decirnos si el hablante es hombre mujer o niño, mediante unos dibujos que aparecerán al momento de hablar. no dirá la tonalidad que es la que determina el tipo de persona y las formantes que son las que detectan el tipo de vocal pronunciada. Y por último tenemos el botón de salir de programa, para utilizarlo en el momento que dejemos de trabajar con él.
  
4. Utilice primero la opción *harmónica* y hable por el micrófono, por medio de la tonalidad de su voz le dirá si el hablante es hombre, mujer o niño.
  
5. Ahora pronuncie una vocal y observe las frecuencias de las formantes y compárelas con las de la tabla para cada vocal pronunciada.

6. Ahora coloque el modo de visualización en *vogais* y pronuncie una por una las vocales hasta que el programa sea capaz de reconocerlas todas.
7. Cierre el programa.

### **PREGUNTAS SOBRE EL DESARROLLO DEL LABORATORIO**

1. ¿Cómo cree usted que se podría implementar en tiempo real un efecto de generación de vocales?

### **CONCLUSIONES Y OBSERVACIONES**

**LABORATORIO DE TRATAMIENTO DIGITAL DE SEÑALES**

**GUÍA DE PRÁCTICAS DEL DOCENTE**

**FRANCISCO JAVIER OSORIO MARTÍNEZ  
JHONNY MARK VELA SALCEDO**

**CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLIVAR**

**FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA**

**CARTAGENA DE INDIAS, D. T. H. Y C.**

**2003**

## CONTENIDO

INTRODUCCION.....	98
REGLAMENTO DEL LABORATORIO.....	103
PRACTICAS	
8. Introducción al manejo del módulo de prácticas..... y Generación de onda seno	107
9. Efectos de cuantización de señal.....	118
10. Transformada rápida de Fourier.....	127
11. Implementación de filtros FIR.....	135
12. Implementación de filtros IIR.....	144
13. Implementación de filtros adaptativos.....	153
14. Tratamiento de sonido.....	161

## INTRODUCCION

En el transcurso de la carrera no hemos dado cuenta que el estudiante siempre se interesa en implementar y aplicar los conocimientos adquiridos en la materia en ejemplos de aplicación de la vida real y en el caso del DSP desarrollar sistemas en tiempo real, para poner en practica la teoría. El propósito de estas prácticas será el de implementar en hardware la teoría que anteriormente solo se simulaba y desarrollar en el estudiante la habilidad de diseño en sistemas de procesamiento de señales.

De todas las cosas que el estudiante hace en el diseño, la única cosa que no se había hecho antes era la implementación de DSP en hardware. Parte de la preparación de implementar sistemas en tiempo real consiste en aprender a utilizar el hardware y las herramientas de software, para facilitar esto tenemos dos prácticas de introducción y seis más de experimentos para que el estudiante se vaya familiarizando con dichas herramientas y aprenda a utilizar los algoritmos básicos.

Las primeras prácticas de introducción enseñan a utilizar el assembler, el debugger y como correr programas en el DSK. En las siguientes experiencias el estudiante aplica algoritmos de procesamiento de señales tales como filtros FIR,

filtros IIR, filtros adaptativo y FFT. También veremos experimentos de precisión, cuantización y generación de señales, así como tratamiento de sonido.

Cada una de las prácticas estará concatenada con la materia de acuerdo al programa de la asignatura,. La siguiente es la forma como consideramos más apropiada, para el seguimiento de la materia con las prácticas:

### **Generalidades: Señales y sistemas discretos.**

*Práctica 1*

### **Representación digital de señales:**

### **Representación espectral de las señales.**

*Práctica 2*

### **Algoritmos rápidos para el cálculo de espectros.**

*Práctica 3*

### **Sistemas discretos: Implementación software de sistemas filtros digitales lineales. Aplicaciones a audio y video**

*Práctica 4*

*Práctica 5*

## **Filtrado no lineal: Algoritmos básicos**

### *Práctica 6*

## **Señales y sistemas multidimensionales tratamiento de imágenes**

### **Ejemplos de aplicación: Síntesis y reconocimiento de voz**

### *Práctica 7*

Hay tres consideraciones importantes en el de experimentos. La primera es hacer que el estudiante aprenda haciendo, esto se logra haciendo que el estudiante aprenda las habilidades necesarias haciendo todo con sus propias manos. Segundo, el énfasis de este laboratorio es en el procesamiento de señales, no en la programación, si el estudiante esta interesado necesitaría conocimientos básicos de programación en C o en lenguaje ensamblador, por otro lado el estudiante tendrá acceso a los algoritmos de los experimentos escritos en lenguaje ensamblador, Todos los programas estarán en una carpeta en la unidad C desde la cual trabajaremos y en la que también estarán ubicadas las herramientas a utilizar. La tercera consideración es que debido a que la materia solo es de un semestre y que el énfasis no es en programación, muchas subrutinas hechas para la simulación en MATLAB serán provistas en las practicas para que el estudiante las utilice ya sea transcribiéndolas o copiándolas desde la carpeta de trabajo donde también estarán ubicadas. Las subrutinas necesarias

para acceder al conversor A/D y D/A y para la FFT también serán provistas por nosotros.

La descripción de cada experimento será hecha de manera concisa de forma que la guía va llevando al estudiante a través de toda la práctica de forma que tengan una idea clara de que es lo que se va a hacer en el proyecto. Esto será un procedimiento gradual, empezando por experimentos simples hasta llegar a unos mas complejos, en los dos primeros las descripciones serán muy específicas, dando los comandos necesarios y las instrucciones a utilizar mientras el estudiante se acostumbra a ellas, luego simplemente se dará la instrucción y el estudiante seguirá los pasos de acuerdo a lo que ha aprendido. También se recomienda leer teoría de otras fuentes y hacer comparaciones, además de leer el manual de operación y mantenimiento del TMS320C3X DSK y sus herramientas antes de empezar el laboratorio.

El equipo que se suministrará en el laboratorio será:

- PC's
- Módulo DSK
- Osciloscopio
- Generador de señales
- Cables y conectores

El software disponible será:

- MATLAB
- Assembler para TMS320C3X bajo DOS
- Debugger para TMS320C3X bajo DOS
- Software C3XDSK 1.0 bajo Windows
- Aplicaciones Anexas

## REGLAMENTO DE LOS LABORATORIOS

18. Se entenderá por **PRÁCTICA DE LABORATORIO** no sólo la ejecución, sino resultado y conclusiones a que diere lugar cada una de ellas, en consecuencia y para la realización de informes y evaluación se tendrán en cuenta estos factores.
19. En el primer día de clase el profesor procederá a distribuir a los estudiantes debidamente matriculados en subgrupos, cada uno de los cuales tendrá un código o número distintivo. El número de subgrupos permitirá al estudiante conocer previamente que experimento o práctica debe realizar en la sección de laboratorio semana por semana.
20. Los laboratorios inician actividades a la hora señalada en el horario, efectuando el quiz semanal.
21. Una vez presentado el quiz cada subgrupo recibirá del auxiliar del laboratorio el equipo necesario para efectuar la correspondiente práctica. Los estudiantes deben verificar confrontando los elementos y la lista. Cualquier observación debe hacerse inmediatamente al profesor, puesto

que una vez iniciada la práctica todo daño o pérdida corre por cuenta del subgrupo.

22. Verificado el equipo se anotan claramente los nombres de los estudiantes, grupo, subgrupo, hora y fecha de la realización del experimento. Una vez firmada la hoja del equipo ésta será recogida por el auxiliar.

23. El material o equipo de laboratorio se devolverá durante los 30 minutos finales del tiempo estipulado para cada práctica semanal, una vez realizada la parte experimental exigida.

24. Se aclara que el equipo será recibido por el auxiliar mediante inventario físico con el fin de establecer faltantes. Se realizará además un chequeo para determinar si hubo daño en los aparatos utilizados.

25. En caso de daño por negligencia comprobada, o en caso de pérdida de algún elemento se deberá reponer o pagar además de afectarse la calificación de la respectiva práctica.

26. Con ayuda del manual de laboratorio el estudiante deberá prepararse, de tal manera que pueda ejecutar el experimento por sí solo.

27. La evaluación de esta preparación se llevará a cabo por medio de quices semanales y el control del trabajo de cada estudiante durante la práctica mediante preguntas pertinentes sobre los principios físicos, el montaje o instrumentación utilizada. Si se considera que el estudiante está desarrollando la práctica incorrectamente el profesor podrá exigir la repetición de la misma.
28. Durante la práctica el profesor y el monitor del Laboratorio supervisarán el desempeño de cada estudiante. A juicio del profesor se prestará ayuda, si el grado de dificultad en algún aspecto de la práctica lo requiere.
29. Se realizará un informe por cada subgrupo, el cual se consignará en una libreta cuadriculada tamaño carta y será elaborado en su totalidad a tinta. Las gráficas se elaborarán en papel milimetrado recortado al tamaño de la hoja. Cada informe deberá llevar el nombre de los estudiantes que asistieron y el día de realización de la práctica.
30. El tiempo estipulado para la realización de la práctica es de máximo una hora y 45 minutos, al final del cual se suspende la actividad y se procede a la elaboración del informe.
31. Durante la elaboración del informe no está permitido el intercambio de

información entre subgrupos.

32. Para la calificación del estudiante se tendrán en cuenta los siguientes aspectos:

- Quiz semanal
- Desarrollo de la práctica
- Informe

33. En el desarrollo de la práctica, además del conocimiento y preparación de la misma, se evaluarán otros factores tales como: trabajo en grupo, destreza y habilidad para resolver problemas y orden en el laboratorio.

34. El contenido de cada informe debe ser como mínimo el siguiente:

- Título de la práctica
- Integrantes del subgrupo
- Datos tabulados y características de los instrumentos utilizados
- Cálculos tipo incluyendo el cálculo de errores.
- Resultados tabulados y/o gráficas.
- Análisis e interpretación de gráficas y/o resultados.
- Conclusiones.

# 1. INTRODUCCION AL MANEJO DEL MODULO DE PRACTICAS Y GENERACION DE SEÑALES

## OBJETIVOS

Objetivo General:

El propósito de es aprender a usar las herramientas del software DSK y ejecutar un programa de ejemplo en el DSK.

Objetivos Específicos:

- Aprender a usar el assembler/linker **dsk3a**.
- Aprender a usar el debugger, **dsk3d**, para cargar y ejecutar programas y a la vez ver valores de registros de memoria
- Aplicar la herramienta generador del programa C3XDSK 1.0 y analizar las señales producidas, mediante el osciloscopio.

## **TEMAS DE CONSULTA**

DSK Hardware Interface, DSK Communications Kernel, TLC32040 AIC Initialization, Host Software.

## **BIBLIOGRAFÍA**

TMS320C3x DSP Starter Kit.pdf. Manual de operación y mantenimiento

## **MARCO TEORICO**

Ver archivo: *Estructura funcional del TMS320C3X DSK.doc* disponible en la carpeta anexos teóricos del CD del laboratorio.

Ver archivo: *TMS320C3x DSP Starter Kit.pdf* disponible en la carpeta anexos teóricos del CD del laboratorio.

Ver documento: *Manual de mantenimiento y operación del modulo y el software C3XDSK 1.0.*

## PRELIMINARES

El procedimiento a realizar en esta práctica será el de ensamblar un archivo .asm utilizando las herramientas dsk3a.exe, y el c3xdsk 1.0, generando un archivo .dsk que se ejecutará para corroborar su funcionamiento y el del modulo.

## PREGUNTAS DE REPASO

1. Explique brevemente en que consiste el muestreo y la cuantización.

R:\ El muestreo digital de una señal analógica trae consigo una discretización tanto en el dominio temporal como en el de la amplitud. Cuando una señal es pasada de su forma análoga a la forma digital, esta es representada usando bits. Esta representación de un número finito de bits es llamada *Cuantización*. Una muestra representada con 8 bits, tiene mayor precisión que una representada con 4 bits.

2. Explique el teorema del muestreo.

R:/ una señal se debe muestrear a dos veces la frecuencia máxima. Para que esta pueda ser reconstruida sin pérdida de información.

3. ¿Que es la frecuencia de Nyquist?

R:/ Es la máxima frecuencia que debe tener la señal para que pueda ser muestreada.

4. ¿Porque se produce el aliasing?

R:/ Es el fenómeno que se produce cuando no se muestrea una señal como mínimo al doble de su frecuencia máxima.

5. En general, ¿cuales son las condiciones ideales para una buena cuantización?

R:/ Que la señal cumpla con el teorema del muestreo y al frecuencia de muestreo se lo más alta posible. Además que el de conversión A/D Y D/A permita niveles de Cuantización altos.

## **EQUIPO**

- Osciloscopio.
- (1) Cable BNC-BNC.
- PC con todo el software y archivos previamente instalados.
- Parlante del módulo.
- Cable RCA-JACK.

## PROCEDIMIENTO

15. Conecte el módulo del DSK a la alimentación y al puerto paralelo a través del cable db25.
16. Encienda el módulo.
17. Conecte el osciloscopio a la salida del módulo, a través del cable BNC-BNC.
18. Conecte el cable RCA del parlante a la salida RCA del módulo.
19. Inserte el CD-ROM destinado para el laboratorio, copie la carpeta DSP a la unidad C del computador.
20. Ejecute el archivo C3XDSK.EXE. Este archivo extraerá el programa C3XDSK 1.0 a la unidad C, en una carpeta con el mismo nombre.
21. Abra la ventana de comandos de DOS y vaya al directorio donde se encuentran los archivos a ser utilizados, en este caso trabajaremos en el directorio >>C:\DSP\ allí han sido localizados todos los programas que

vamos a utilizar durante el curso. Para esta experiencia cerciorese que el archivo **minuet.txt** se encuentre en la carpeta DSP.

22. Ejecute desde la ventana de DOS la siguiente instrucción: **C:\DSP\dsk3a demo1d.asm** con esto ya habremos ensamblado el archivo, creándose uno ejecutable con el nombre de **demo1d.dsk**.

23. Ejecute la siguiente instrucción: **>>C:\DSP\dsk3d reset** después de esto se debe ver al siguiente ventana:

DISASSEMBLY		C31 DSP STARTERS KIT					
809c03	50700080 start	LDIU	00080h, DP	PC	00809c03	SP	008098de
809c04	08349c2c	LDI	@09c2cH, SP	R0	00000000	R1	00000000
809c05	07608000	LDF	0.000000e+00, R0	R2	00000000	R3	00000000
809c06	c610c1c0	LDI	*AR0, R0    LDI *AR	R4	00000000	R5	00000000
809c07	c610c1c0	LDI	*AR0, R0    LDI *AR	R6	00000000	R7	00000000
809c08	08600100	LDI	256, R0	AR0	00000000	AR1	00000000
809c09	09a09c00	LSH	@09c00H, R0	AR2	00000000	AR3	00000000
809c0a	61809c0e	BRD	jump	AR4	00000000	AR5	00000000
809c0b	07618000	LDF	0.000000e+00, R1	AR6	00000000	AR7	00000000
809c0c	07628000	LDF	0.000000e+00, R2	IRO	00000000	IR1	00000000
809c0d	07630000	LDF	1.000000e+00, R3	ST	00000000	RC	00000000
809c0e	07640000	jump	LDF 1.000000e+00, R4	RS	00000000	RE	00000000
809c0f	087b0003	loop	LDI 3, RC	DP	00000000	BK	00000000
809c10	64809c1a	RPTB	block	IE	00000000	IF	00000000
809c11	02640001	ADDI	1, R4				

COMMAND		MEMORY	
Texas Instruments 1994		809800	00000007 ffffffff 00809802 00809827
load testa		809804	0080982c 00809839 0080983c 0080983f
		809808	00809843 00809842 00809868 0080989a
		80980c	008098a9 10800000 0f350000 0f300000
		809810	0f200000 0f320000 0f280000 0f290000
		809814	1a770004 6a050006 628098a9 50700080

figura 1.1

En la subventana inferior izquierda (ventana de comandos) ejecute la siguiente instrucción: **load demo1d.dsk** esto hace que el programa se cargue en la memoria del DSP, ahora presione F5 para ejecutarlo.

24. Observe la salida en el osciloscopio y escúchela en el parlante. Para cancelar la ejecución del programa presione cualquier tecla, luego de esto ejecute la instrucción **reset** y después **exit** para salir del programa.

25. Copie el archivo demo1d.asm y minuet.txt en la carpeta C3XDSK. Ejecute el archivo c3xdsk.exe ubicado en la misma carpeta. En la barra de menús haga click en la opción *CMP*. Elija demo1d.asm y haga doble click sobre el. Inmediatamente se ensamblará (debe aparecer una ventana temporal durante el ensamble, que luego se cierra).

26. Vaya al menú opciones del software y seleccione modo conectado.

27. En el menú archivo elija cargar archivo demo1d.dsk para cargarlo al módulo y ejecutarlo. Corrobore su efecto en el osciloscopio y escúchelo en el parlante.

R:\ se debe escuchar un Minueto y en el osciloscopio se debe visualizar la señal con múltiples componentes senoidales.

28. Para finalizar el algoritmo, en el menú *comandos* seleccione la opción *reiniciar*. y haga clic en *salir*.

## GENERACIÓN DE SEÑALES

5. Vaya al menú *opciones* del software y seleccione *modo conectado*.
6. haga clic sobre la opción generador. Aparecerá un cuadro con posibilidades para generar formas de onda triangulares, cuadradas y senoidales y de establecer su amplitud y frecuencia.
7. Llene la siguiente tabla de datos, donde la frecuencia ajustada es la que se ha de colocar en el recuadro frecuencia de la herramienta *generador*, ajuste la amplitud en su máximo valor, pulsando el botón seno y después procesar por cada frecuencia ajustada (el voltaje de Offset manténgalo en 0 V).

Frecuencia							
Ajustada	100Hz	500Hz	1.0KHz	3.0KHz	6.5KHz	7KHz	9KHz
Medida	105Hz	520Hz	1.06KHz	3.4KHz	6.7KHz	-	-

Error	5%	4%	6%	13%	3%	-	-
-------	----	----	----	-----	----	---	---

F. ajustada	100Hz	500Hz	1.0KHz	3.0KHz	6.5KHz	9KHz
Voltaje pico	≈3 V	≈3 V	≈3 V	≈2.8 V	2.3 V	<1 V

Tabla 1.1

8. Para salir del programa, En el menú *comandos* seleccione la opción *reiniciar.* y haga clic en *salir*

## PREGUNTAS SOBRE EL DESARROLLO DEL LABORATORIO

5. ¿De acuerdo a lo observado al ejecutar el archivo **demo1d.dsk** de la primera parte de la experiencia, ¿Cómo cree que se pudo haber generado esa secuencia de sonido?
- R:/ por medio de una rutina de ejecución de señales senoidales de diferente frecuencia.
6. ¿De acuerdo al cuadro de dialogo mostrado por al aplicación al generar la señal, porque cree usted que el rango máximo para generación no puede ser mayor de 9999 Hz.?

R:/ Como la frecuencia de muestreo del procesador es cercana a los 20 KHz, la señal a generar no puede ser mayor de los 10 KHz. Además El TMS320C3X DSK posee un filtro antialiasing que actúa como un filtro pasabaja a la salida y entrada del circuito de interface análoga, que es quien realiza las funciones de muestreo.

7. ¿Según las medidas a partir de que valor de frecuencia la señal se vuelve realmente despreciable? ¿Por qué ?

R:/ A partir de la señal de 7 KHz. La señal a generar empieza a partir de esa frecuencia a ser parte del ancho de transición del filtro antialiasing, quien esta a punto de suprimir completamente la señal para evitar sobrepasar los 10 KHz teóricos,(que corresponden mas o menos a 9 KHz en la práctica).

8. Con la aplicación genera una onda triangular y una cuadrada y experimente con varias frecuencias entre 5000 Hz y 9999 Hz. ¿Qué efecto puede apreciar? ¿Porque cree usted que sucede esto?

R:/ se convierten en una onda seno, ya que el método para generar la onda se basa en puntos que son interpolados. Se puede apreciar en el cuadro de dialogo que genera la aplicación *generador* que al procesar la señal para frecuencias muy altas el número de puntos utilizados es insuficiente, con lo

que la señal a generar (ya sea triangular o cuadrada) siempre será la misma.

## **CONCLUSIONES Y OBSERVACIONES**

-Del cuadro de dialogo de la aplicación *Generador*, podemos apreciar que el numero de puntos usados para realizar la generación de ondas se obtiene de dividir la frecuencia de muestreo entre la frecuencia a obtener.

-Si se aplica un aumento de escala a las graficas (Zoom) se puede notar un rizado en las ondas generadas, evidentemente es producto del metodo empleado para la obtención de la misma (interpolación).

-Generar una función con DSPs es mas sencillo que cualquier otro procesador de uso general, se deben conocer manejo de registros, características de muestreo y manejo de reloj maestro.

## 2. EFECTOS DE CUANTIZACION DE SEÑAL

### OBJETIVOS

Objetivo General:

El propósito de esta práctica es observar los efectos de cuantización y redondeo en una señal.

Objetivos Específicos:

- Conocer la relación entre los bits que son usados para representar una señal y los que no.
- Conocer en que afecta la implementación en el desarrollo del sistema.
- Conocer en que afecta la precisión en el desarrollo del sistema.

## **TEMAS DE CONSULTA**

Teorema de muestreo y cuantización, conversión análoga-digital y digital-análoga, efecto aliasing.

## **BIBLIOGRAFIA**

Proakis John, Manolakis Dimitris. Tratamiento Digital De Señales Principios Algoritmos Y Aplicaciones. Editorial Prentice may.

## **MARCO TEORICO**

Ver documento: Laboratorio de Tratamiento Digital de Señales. Sección 2.1 *cuantización y muestreo*.

Ver archivo: *Estructura funcional del TMS320C3X DSK.doc* disponible en la carpeta anexos teóricos del CD del laboratorio.

Ver archivo: *TMS320C3x DSP Starter Kit.pdf* disponible en la carpeta anexos teóricos del CD del laboratorio.

## PRELIMINARES

El procedimiento a realizar en esta práctica será el de introducir una señal seno por la entrada análoga del AIC y luego llevarla a la salida por medio de un programa, en dicha salida se mirará la señal resultante por medio del osciloscopio después de haberse hecho la cuantización y reconstrucción de la señal y se comparará con la señal de entrada. Al final se verá un ejemplo ilustrativo con una señal de audio.

## PREGUNTAS DE REPASO

1. Explique brevemente en que consiste el muestreo y la cuantización.

R:\ El muestreo digital de una señal analógica trae consigo una discretización tanto en el dominio temporal como en el de la amplitud. Cuando una señal es pasada de su forma análoga a la forma digital, esta es representada usando bits. Esta representación de un número finito de bits es llamada *Cuantización*. Una muestra representada con 8 bits, tiene mayor precisión que una representada con 4 bits.

2. Explique el teorema del muestreo.

R:/ una señal se debe muestrear a dos veces la frecuencia máxima. Para que esta pueda ser reconstruida sin pérdida de información.

3. ¿Que es la frecuencia de Nyquist?

R:/ Es la máxima frecuencia que debe tener la señal para que pueda ser muestreada.

4. ¿Porque se produce el aliasing?

R:/ Es el fenómeno que se produce cuando no se muestrea una señal como mínimo al doble de su frecuencia máxima.

9. En general, ¿cuales son las condiciones ideales para una buena cuantización?

R:/ Que la señal cumpla con el teorema del muestreo y al frecuencia de muestreo se lo más alta posible. Además que el de conversión A/D Y D/A permita niveles de Cuantización altos.

## **EQUIPO**

- Osciloscopio.
- Generador de señales.
- (3) Cables BNC-BNC.
- (1) T-BNC.
- PC con todo el software y archivos previamente instalados.
- Parlante del módulo.
- CD de audio.
- Cable RCA-JACK.

## **PROCEDIMIENTO**

11. Conecte el módulo del DSK a la alimentación y al PC a través del puerto paralelo.

12. Conecte un generador de señales y una de las entradas del osciloscopio a la entrada analógica del módulo y la otra entrada del osciloscopio a la salida del módulo. Utilice cables BNC-BNC para las conexiones y para conectar el

osciloscopio y generador a la entrada del módulo a la vez, utilice una T-BNC.

13. Encienda el módulo.

14. Copie el archivo **loopaic.asm** ubicado en la carpeta DSP, a la carpeta C3xdsk y haga clic en la opción CMP (compilar), escoja el archivo y haga doble-click sobre el, inmediatamente se compilará. Generando el archivo **loopaic.dsk**.

15. En el menú *Archivo* seleccione *Cargar Archivo DSK* y ábralo. Quedará cargado en la tarjeta.

16. Introduzca una señal seno por medio del generador de señales a través de la entrada análoga, con una amplitud no mayor a 1V-pico, establézcale una frecuencia de aproximadamente 1.0KHz y conecte una de las entradas del osciloscopio a esta señal para observarla, la otra entrada conéctela a la salida análoga del módulo para poder comparar las dos señales simultáneamente. Si utiliza un osciloscopio análogo colóquelo en modo Both y Alt., en el recuadro de configuración vertical.

17. Ahora varíe la frecuencia y observe que cambio puede tener en la salida.

Tome datos de atenuación de la señal de salida con respecto a la entrada, cambio de fase y amplitud.

Medición	1	2	3	4	5	6	7
Frecuencia	1 KHz	2 KHz	3KHz	4KHz	5KHz	6KHz	9KHz
Amplitud	1Vp	1Vp	1Vp	1Vp	1Vp	1Vp	1Vp
Fase	0°	0°	0°	0°	0°	0°	0°
Atenuación	50%	50%	50%	50%	50%	>70%	>90%

Tabla 2.1

18. Detenga la ejecución del programa y desconecte todos los cables BNC del modulo provenientes del generador de señales y el osciloscopio.

19. Ahora conecte la entrada RCA del módulo a la unidad de CD por medio del cable RCA-JACK y conecte el parlante a la salida RCA y podrá escuchar el sonido.

20. Para salir del programa, En el menú *comandos* seleccione la opción *reiniciar.* y haga clic en *salir*

## PREGUNTAS SOBRE EL DESARROLLO DEL LABORATORIO

1. ¿Que atenuación proporciona el programa *loopaic.dsk* a la señal de entrada?

R:/ una atenuación del 50%, para este caso en particular, se pretende evitar sobrevoltaje por seguridad, reduciendo la ganancia.

2. ¿Cual es la amplitud máxima que se puede obtener a la salida del DSK antes que se recorte la señal?

R:/ 5 Voltios. los ajustes de ganancia solo permiten hasta este valor que es el voltaje con que se alimenta el DSK.

3. ¿A que frecuencia se pierde significativamente la señal en casi 70% de la amplitud promedio?

R:/ En aproximadamente 6 KHz

4. ¿Que produce las limitaciones de frecuencia en el DSK?

R:/ La respuesta en frecuencia del DSK (mas exactamente del circuito de interface análoga). Nos dice que solo se puede muestrear a 20 KHz. Primero que todo por ser una característica del DSP y segundo por tener el sistema en general un filtro antialiasing.

## **CONCLUSIONES Y OBSERVACIONES**

El DSK cuantiza usando 14 bits y puede tener un máximo SNR de 86.05dB. Esto no significa que todas las señales tendrán un SNR de 86.05dB, este es el SNR de una onda seno usando la máxima amplitud de entrada. Si entramos una señal pequeña, algunos de los 14 bits serán siempre cero y el SNR será más pequeño. De aquí que la reconstrucción de la señal haya sido realmente excelente, como bien se pudo apreciar.

### 3. TRANSFORMADA RAPIDA DE FOURIER

#### OBJETIVOS

Objetivo General:

El propósito de esta práctica es aplicar la transformada rápida de fourier en tiempo real.

Objetivos Específicos:

- Implementar la transformada rápida de Fourier en tiempo real.
- Entender el algoritmo de FFT.

#### TEMAS DE CONSULTA

Transformada de Fourier. Transformada discreta de Fourier . FFT

#### BIBLIOGRAFIA

Proakis John, Manolakis Dimitris. Tratamiento Digital De Señales Principios Algoritmos Y Aplicaciones. Editorial Prentice Hall. Curso DSP incluido en el CD.

## MARCO TEORICO

Ver documento: Laboratorio De Tratamiento Digital De Señales seccion 2.2:  
*Transformada De Fourier.*

## PRELIMINARES

Se realizará una simulación en MATLAB, un sencillo algoritmo de ejemplo en el cual aplicamos la Transformada Rápida de Fourier (FFT) a una señal exponencial modulada en amplitud. Como puede ver la transformada se hace ejecutando la instrucción **fft ()** y los demás son solo pasos para graficar. Luego se analizara el efecto de una FFT de 256 en la reconstrucción de una señal cuadrada.

## PREGUNTAS DE REPASO

1. ¿Explique en que consiste la DFT?

R:/ la transformada discreta de Fourier es el proceso que busca discretizar las variables continuas y limitar el números de muestras en los dos dominios (temporal y frecuencial). EL DFT es una aproximación al espectro de la señal analógica original. Su magnitud se ve influenciada por el

intervalo de muestreo, mientras que su fase depende de los instantes de muestreo.

2. ¿Para que se usa la FFT?

R:/ Es un algoritmo que busca calcular eficientemente la DFT.

## **EQUIPO**

- Osciloscopio.
- Generador de señales.
- (3) Cables BNC-BNC.
- (1) T-BNC.
- PC con todo el software y archivos previamente instalados.
- Parlante del módulo.
- CD de audio.
- Cable RCA-JACK.

## **PROCEDIMIENTO**

### **SIMULACION**

5. Ubique el archivo ffej3.m ubicado en la carpeta DSP en la carpeta de trabajo de MATLAB (para la versión 5.0 o superior, la carpeta se llama work, ubicado en al carpeta MATLABR11).
6. En la consola de comandos de MATLAB, digite: ffej3(128,0.2) y presione ENTER.

7. Observe las graficas y saque conclusiones.

R:/ se observa la señal en el dominio del tiempo, la grafica de los módulos de los coeficientes espectrales (es decir la magnitud de cada coeficiente contra la frecuencia ). Al hablar de coeficientes espectrales son los coeficientes de las componentes de la señal en el dominio de la frecuencia, vista como una serie discreta de Fourier.

8. Cierre la ventana de MATLAB.

## IMPLEMENTACION EN TIEMPO REAL

1. Conecte un generador de señales y una de las entradas del osciloscopio a la entrada análoga del módulo y la otra entrada del osciloscopio a la salida del módulo. Utilice cables BNC-BNC para las conexiones y para conectar el osciloscopio y generador a la entrada del módulo a la vez, utilice una T-BNC.

2. Encienda el modulo.

3. Ejecute el software C3XDSK 1.0, vaya al menú *opciones* y seleccione la opción *modo conectado*.

4. Ajuste el generador de señales de tal forma que genere una onda cuadrada con una amplitud no mayor de 1V-pico y frecuencia 1 KHz.

9. Haga clic sobre el menú *FFT* para ejecutar la transformada (dicha transformada está hecha para 256 puntos).

10. Determine la relación entre la amplitud, frecuencia y forma de la señal de entrada y la de salida.

R:/ partiendo del concepto de reconstrucción de señales a través de la transformada de Fourier, se requiere un gran número de puntos para que la señal reconstruida sea lo más parecida a la original, siendo esto así. Vemos la amplitud y frecuencia básicamente iguales pero la onda está distorsionada por los efectos de la reconstrucción (efecto Gibbs) , aunque la FFT de 256 puntos para efectos rápidos muestra ser eficiente para procesamiento en tiempo real.

11. Varíe lentamente la frecuencia en la entrada y observe el cambio a la salida.

R:/ para frecuencias altas, se dificulta la reconstrucción de la señal, esta aparece demasiado distorsionada a la salida.

12. Para salir del programa, En el menú *comandos* seleccione la opción *reiniciar.* y haga clic en *salir*

## **PREGUNTAS SOBRE EL DESARROLLO DEL LABORATORIO**

5. Explique a que se refieren los parámetros 128 y 0.2 ingresados para la ejecución del algoritmo *fftej3*.

R:/ 128 se refiere a los puntos utilizados para implementar la FFT (en el algoritmo se identifica como N). 0.2 se refiere al tiempo de aplicación de los N puntos de muestreo.

6. ¿Qué representa las graficas obtenidas en MATLAB?

R:/ se observa la señal en el dominio del tiempo, la grafica de los módulos de los coeficientes espectrales (es decir la magnitud de cada coeficiente contra la frecuencia ). Al hablar de coeficientes espectrales son los coeficientes de las componentes de la señal en el dominio de la frecuencia, vista como una serie discreta de Fourier.

7. ¿según la experiencia de los ítem 4 y 5 que tratamiento realiza la FFT sobre dicha señal?

R:/ la aplicación muestrea la señal, calcula el espectro mediante una FFT de 256 puntos de muestreo, el resultado que arroja son los llamados coeficientes espectrales que se utilizan para reconstruir la señal. Esta operación demanda todo el *performance* del DSP.

8. ¿Que diferencia significativa encontró en la reconstrucción en la onda cuadrada?

R:/ Rizados, efecto Gibbs. Cierta inestabilidad en la onda.

## **CONCLUSIONES Y OBSERVACIONES**

La FFT de 256 puntos es bastante eficiente al reconstruir una señal simple como una onda seno, es insuficiente al reconstruir una onda cuadrada. Aun más por las limitaciones del TMS320C3X DSK. Se pueden utilizar mayores puntos, pero demanda más rendimiento al procesador, lo que en un sistema de tratamiento en tiempo real tan sencillo como este se traduce en ralentización del sistema, demanda de memoria RAM, etc.

## 4. IMPLEMENTACION DE FILTROS FIR

### OBJETIVOS

Objetivo General:

El objetivo de este proyecto es implementar un filtro FIR en tiempo real.

Objetivos Específicos:

- Simular un filtro FIR con la ayuda de MATLAB.
- Implementar un filtro FIR en tiempo real utilizando el DSP y la herramienta FIR del software C3XDSK 1.0
- Modificar algunos parámetros del filtro con la herramienta y analizar los cambios en la señal de prueba.

### TEMAS DE CONSULTA

Filtros digitales. Filtros FIR. Características de los filtros FIR.

## **BIBLIOGRAFIA**

Proakis John, Manolakis Dimitris. Tratamiento Digital De Señales Principios Algoritmos Y Aplicaciones. Editorial Prentice Hall. Curso DSP incluido en el CD.

## **MARCOTEORICO**

Ver documento: *Laboratorio de Tratamiento Digital de Señales. Sección 2.3 filtros digitales.*

Ver archivo: *manual de operación y mantenimiento del modulo y del software C3XDSK 1.0 (aplicación FIR TÍPICOS)*

## **PRELIMINARES**

Aplicaremos los conceptos estudiados de un filtro FIR para hacer primero una simulación en MATLAB y luego implementarlo en tiempo real a través del módulo del DSK.

## PREGUNTAS DE REPASO

6. ¿Que es un filtro digital?

R:/ Un filtro digital se entiende como cualquier procesamiento realizado en la señal de entrada digital.

7. Enumere los pasos necesarios para el diseño de un filtro digital

R:/ Establecer las especificaciones del filtro para unas determinadas prestaciones. Estas especificaciones son las mismas que las requeridas por un filtro analógico : frecuencias de parabanda y pasabanda, atenuaciones, ganancia dc, etc. Determinar la función de transferencia que cumpla las especificaciones. Realizar la función de transferencia en hardware o software.

8. Enumere tres características de los filtros FIR.

R:/ No hay recursión, es decir, la salida depende sólo de la entrada y no de valores pasados de la salida. La respuesta es por tanto una suma ponderada de valores pasados y presentes de la entrada. De ahí que se denomine Media en Movimiento (Moving Average) La función de Transferencia tiene un denominador constante y sólo tiene ceros.

9. Diga dos ventajas de los filtros FIR con respecto a los dos filtros.

R:/ los filtros FIR son de fase lineal. Los filtros FIR siempre son estables.

## **EQUIPO**

- Osciloscopio.
- (3) Cable BNC-BNC.
- (1) conector en T tipo BNC
- PC con todo el software y archivos previamente instalados.
- Parlante del módulo.
- CD de audio
- Cable RCA-JACK.

## **PROCEDIMIENTO**

13. Abra el programa MATLAB y ejecute el archivo FIR.m.

14. Para el tipo de filtro a diseñar ingrese lp (pasabajos), para el número de puntos 256. en frecuencia de corte normalizada 0.1 y finalmente ingrese hamming, para el tipo de ventana. El programa debe arrojar la respuesta en frecuencia (diagrama de bode) del filtro. Mantenga las gráficas abiertas.

15. Conecte el modulo del DSK a la alimentación y al PC a través del puerto paralelo.
16. Conecte un generador de señales y una de las entradas del osciloscopio a la entrada análoga del módulo y la otra entrada del osciloscopio a la salida del módulo. Utilice cables BNC-BNC para las conexiones y para conectar el osciloscopio y generador a la entrada del módulo a la vez, utilice el conector en T tipo BNC.
17. Con el generador de señales alimente el modulo con una onda seno de 1 Vp.
18. Abra el programa C3XDSK 1.0. Vaya al menú *opciones* y seleccione la opción *modo conectado*, luego haga click sobre el menú *IIR*, después de esto se abrirá una ventana donde especificará el tipo de filtro y sus características. Especifique un filtro de orden 15, pasabajo tipo butterworth (el método que usa la aplicación para implementar el filtro es el de transformación bilineal) y haga clic en aceptar.
19. En la aplicación *FIR típicos*, elija filtro pasabajos, frecuencia de corte 1 KHz también, ventana Hamming y coeficiente 256. deje los demás valores por

defecto. seleccione *aceptar* y observe la salida a través del osciloscopio.

20. En el generador Varíe la frecuencia alrededor de 1KHz. Tome valores de frecuencia para atenuaciones de 50%, 70%, 90% y 95%. Llene la tabla.

Atenuación	50%	70%	90%	95%
Frecuencia	1.3KHz,	1.4KHz,	1.5KHz	1.55KHz

Tabla 4.1

21. Vaya nuevamente a la aplicación FIR típicos y ajuste la frecuencia de corte a 9000 Hz, haga click en aceptar y varíe la frecuencia de la onda seno. Mida el valor de frecuencia para el cual la señal se atenúa en un 70% y anótelo.

R:/ El filtro no debe trabajar, pues a 9000 Hz. la señal ya no debe existir, ya que el filtro antialiasing la trunca al acercarse valor de Nyquist.

22. Vaya al menú *Comandos* y seleccione *Reiniciar*.

23. Conecte la unidad de CD a la entrada del módulo a través del cable RCA – JACK y el parlante a la salida RCA. Ejecute un CD de audio.

24. Vaya el menú *FIR* y seleccione *ecualizador FIR*, con esta aplicación podrá suprimir, atenuar o acentuar componentes de frecuencia, efecto que podrá apreciar de manera auditiva.

## **PREGUNTAS SOBRE EL DESARROLLO DEL LABORATORIO**

6. ¿Qué significado tiene la grafica producto de la ejecución del archivo FIR.m en MATLAB? ¿Para que se utilizan?

R:/ El diagrama de bode del filtro, contiene la respuesta en frecuencia del filtro, con esto determinamos el tipo de filtro (pasabajas, pasaaltas, etc.) la frecuencia de corte, etc.

7. ¿Por qué se utiliza una frecuencia normalizada de 0.1? ¿Cómo se obtiene?

R:/ la normalización se realiza con respecto a la frecuencia de Nyquist, es decir para una frecuencia de muestreo de 19531 Hz, sería aproximadamente 10 KHz. La frecuencia de corte del filtro es de 1KHz, lo que nos da como resultado una frecuencia normalizada de 0.1.

8. De acuerdo a la grafica magnitud contra frecuencia arrojado por MATLAB. ¿Cuál es el valor de la frecuencia de corte?

R:/ La frecuencia de corte normalizada de acuerdo al diagrama de bode de MATLAB es de 0.098, lo que equivale a una frecuencia de aproximadamente 980 Hz.

9. ¿Coincide esta frecuencia de corte con la obtenida en tiempo real con el DSP? ¿Cuál es el valor?

R:/ El valor de la obtenida en tiempo real es de 1.2 KHz lo que da un error de aproximadamente 22% aceptable por encontrarse la frecuencia dentro del ancho de transición estipulado para el filtro.

10. ¿Coincide la frecuencia de corte ajustada (9000 Hz) en el numeral 9 del procedimiento con la obtenida? Explique el resultado.

R:/ No, el filtro no debe trabajar, pues a 9000 Hz. la señal ya no debe existir, ya que el filtro antialiasing la trunca al acercarse valor de Nyquist.

## **CONCLUSIONES Y OBSERVACIONES**

Aplicaciones como las de los FIR típicos facilitarán la observación de resultados y experimentación didáctica con fines pedagógicos. Aunque pareciera que con la aparición de más y mejores software para tratamiento de señales, es absolutamente necesario destinar un procesador para tratamiento de señales que son debidamente acondicionados para trabajar ininterrumpidamente y no utilizar procesadores de propósito general bastante costosos (pentium IV, ATHION, etc.) y no destinados para tal fin.

## 5. IMPLEMENTACION DE FILTROS IIR

### OBJETIVOS

Objetivo General:

El objetivo de este proyecto es implementar un filtro IIR en tiempo real.

Objetivos Específicos:

- Simular un filtro IIR con la ayuda de MATLAB.
- Implementar un filtro IIR en tiempo real utilizando el DSP y la herramienta IIR del software C3XDSK 1.0
- Modificar algunos parámetros del filtro con la herramienta y analizar los cambios en una señal de prueba.

## **TEMAS DE CONSULTA**

Filtros digitales. Filtros IIR. Características de los filtros IIR.

## **BIBLIOGRAFÍA**

Proakis John, Manolakis Dimitris. Tratamiento Digital De Señales Principios Algoritmos Y Aplicaciones. Editorial Prentice Hall. Curso DSP incluido en el CD.

## **MARCO TEORICO**

Ver documento: *Laboratorio de Tratamiento Digital de Señales. Sección 2.3 filtros digitales.*

Ver archivo: *manual de operación y mantenimiento del módulo y del software C3XDSK 1.0 (aplicación IIR)*

## PRELIMINARES

Aplicaremos los conceptos estudiados de un filtro IIR para hacer primero una simulación en MATLAB y luego implementarlo en tiempo real a través del módulo del DSK.

## PREGUNTAS DE REPASO

4. Enumere tres características de los filtros IIR.

R:/ La función de transferencia contiene solo polos. El filtro es recursivo ya que la salida depende no solo de la entrada actual, sino además de valores pasados de la salida (Filtros con realimentación). La respuesta al impulso es normalmente de duración infinita, de ahí su nombre.

5. Diga dos ventajas de los filtros IIR con respecto a los otros filtros.

R:/ Los filtros IIR producen en general distorsión de fase, es decir la fase no es lineal con la frecuencia. El orden de un filtro IIR es mucho menor que el de un filtro FIR para una misma aplicación.

6. mencione dos aplicaciones de los filtros

R:/ Tratamiento de sonido. Eliminación de interferencia.

## EQUIPO

- Osciloscopio.
- (3) Cable BNC-BNC.
- (1) conector en T tipo BNC
- PC con todo el software y archivos previamente instalados.
- Parlante del módulo.
- CD de audio
- Cable RCA-JACK.

## PROCEDIMIENTO

15. Abra el programa MATLAB y ejecute el archivo IIR.m.

16. Para el orden del filtro ingrese 15 y en el tipo de filtro ingrese *butt*. Debe arrojar una grafica de polos y su respuesta en frecuencia (diagrama de bode).

17. Entre el tipo de conversión lp (pasabajos). Y en frecuencia de corte ingrese 1000 Hz.

18. Cuando le pida el método ingrese bil (bilineal) y en frecuencia de muestreo ingrese 19531 (ya que la frecuencia máxima de muestreo comprobada del DSK es de 19531Hz).
19. Conecte el modulo del DSK a la alimentación y al PC a través del puerto paralelo.
20. Conecte un generador de señales y una de las entradas del osciloscopio a la entrada analógica del módulo y la otra entrada del osciloscopio a la salida del módulo. Utilice cables BNC-BNC para las conexiones y para conectar el osciloscopio y generador a la entrada del módulo a la vez, utilice el conector en T tipo BNC.
21. Con el generador de señales alimente el modulo con una onda seno de 1 Vp.
22. Abra el programa C3XDSK 1.0. Vaya al menú *opciones* y seleccione la opción *modo conectado*, luego haga click sobre el menú *IIR*, después de esto se abrirá una ventana donde especificará el tipo de filtro y sus características. Especifique un filtro de orden 15, pasabajo tipo butterworth (el método que usa la aplicación para implementar el filtro es el de transformación bilineal) y haga clic en aceptar.

23. En el generador varíe la frecuencia de la onda seno entre 100 Hz y 3 KHz.

Mida el valor de frecuencia para el cual la señal se atenúa en un 70% y anótelo.

R:/ La frecuencia a la cual se atenúa la señal en un 70% es de aproximadamente 1.1 KHz

10. Compárelo con el obtenido en la grafica obtenida en MATLAB. Calcule el error relativo.

Frecuencia de corte de la simulación	Frecuencia obtenida en tiempo real	Error
1 KHz	1.1 KHz	10%

Tabla 5.1

21. Nuevamente en el programa C3XDSK 1.0. haga click sobre el menú *IIR*, especifique un filtro de orden 1, pasabajo tipo elíptico y frecuencia de corte 1 KHz.

22. Varíe la frecuencia de la onda seno entre 100 Hz y 5 KHz. Mida el valor de frecuencia para el cual la señal se atenúa en un 70% y anótelo.

R:/ Se atenúa a un frecuencia de 3.6 KHz

23. Repita el ítem 11 y 12 pero para un tipo de filtro Chebyshev.

R:/ Se atenúa a un frecuencia de 5.3 KHz

24. Para salir de la aplicación seleccione menú comandos y reiniciar. Y luego haga clic en salir.

## **PREGUNTAS SOBRE EL DESARROLLO DEL LABORATORIO**

9. ¿De acuerdo al error calculado en el ítem 10 de la experiencia cree usted que el procesamiento en tiempo real es confiable? ¿justifique su respuesta?

R:/ Claro que es confiable, solo que en tiempo real se tienen en cuenta factores como: Velocidad de muestreo, resolución de la conversión en el AIC, errores de cuantización y niveles de tensión máximos permitidos por la tarjeta.

10. ¿En que ayuda la simulación a la hora de implementar un filtro en tiempo real?

R:/ Ayuda a hacer una primera predicción del comportamiento del filtro de acuerdo a los parámetros especificados.

11. ¿Qué implica el utilizar un orden grande en los filtros IIR?

R:/ Se adquiere una mayor precisión, pero se necesitan mas cálculos por parte del procesador, lo que implica tiempo y exigencia.

12. De acuerdo a los resultados obtenidos en los ítem 11, 12 y 13 de la experiencia. ¿Como afecto a la frecuencia de corte, la modificación del orden del filtro?. ¿Porque?

R:/ Se perdió exactitud a la hora de obtener la frecuencia de corte, motivo por el cual se alejo bastante del valor requerido, al utilizar orden 1. Por ejemplo para el peor de los casos en el filtro IIR Chebyshev, se consiguió una atenuación del 70% en aproximadamente 5.3 KHz (Una diferencia considerable respecto a 1KHz).

13. ¿Dónde encontró mejor resultado en el filtro IIR tipo elíptico o en el Chebyshev? ¿Concuerta esto con la teoría de filtros digitales? Justifique su respuesta.

R:/ En el elíptico, ya que se alejo menos del valor esperado. Los filtros digitales entre si no son ni mejores ni peores, están sometidos a las condiciones matemáticas dependiendo del comportamiento de la señal.

## **CONCLUSIONES Y OBSERVACIONES**

Las ventajas y desventajas de algunos tipos de filtros digitales con respecto a otros no son notorias en tratamientos a señales fundamentales como la onda senoidal. Se pretende iniciar al estudiante a que experimente en el tratamiento de señales mas complejas, lo cual requerirá un estudio mas profundo de los DSPs.

## 6. IMPLEMENTACION DE UN FILTRO ADAPTATIVO

Objetivo General:

El propósito de esta práctica es usar filtros adaptativos para la eliminación de una interferencia sinusoidal.

Objetivos Específicos:

- Analizar una simulación en MATLAB de una implementación de filtro adaptativo para eliminar una interferencia sinusoidal.
- Interpretar la operación del sistema de filtro adaptativo implementado en el modulo del DSP en tiempo real.

## **TEMAS DE CONSULTA**

Filtros digitales. Filtros FIR e IIR. Filtros adaptativos. Diseño de filtros adaptativos.

## **BIBLIOGRAFIA**

Proakis John, Manolakis Dimitris. Tratamiento Digital De Señales Principios Algoritmos Y Aplicaciones. Editorial Prentice Hall. Curso DSP incluido en el CD.

## **MARCO TEORICO**

Ver documento: *Laboratorio de Tratamiento Digital de Señales. Sección 2.4 filtros adaptativos.*

## **PRELIMINARES**

Nuestro objetivo será hacer primero una simulación en MATLAB, observar y concluir sobre los resultados obtenidos, para luego pasar a la implementación en tiempo real en el modulo del DSK.

## PREGUNTAS DE REPASO

1. Explique Brevemente en que consiste un filtro adaptativo

R:/ Un filtro adaptativo es un filtro cuyos coeficientes son ajustados de forma adaptativa en función de objetivos o condiciones variables en el tiempo que se traducen en una señal de error.

2. ¿Cuál es la función principal de los filtros adaptativo?

R:/ La función principal de los filtros adaptativo consiste en eliminar o modelar ciertas componentes (posiblemente indeseables).

3. Mencione los cuatro tipos de filtros adaptativos.

R:/ Hay esencialmente cuatro tipos de filtros adaptativo: Identificación, modelización inversa, predicción y eliminación de interferencia.

## EQUIPO

- Osciloscopio.
- Generador de señales.
- (3) Cable BNC-BNC.

- (1) conector en T tipo BNC
- PC con todo el software y archivos previamente instalados.

## PROCEDIMIENTO

## SIMULACION

3. Analice el programa adapta.m, que ilustra la adaptación de un filtro FIR, de orden 32, en la eliminación de una interferencia, como se ilustra en la figura. En el inicio del proceso de adaptación el filtro FIR tendrá coeficientes nulos. Se usa como interferencia a eliminar una señal sinusoidal. Este mismo sistema será implementado en el DSK.

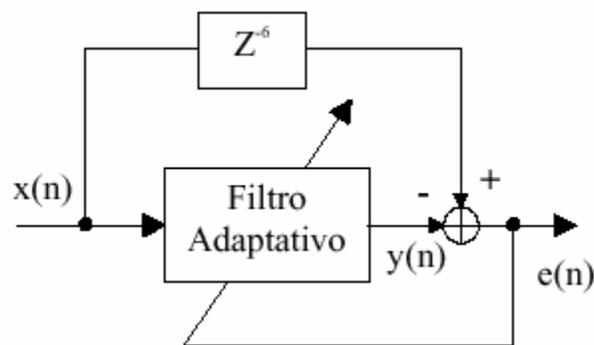


Figura 6.1

4. Concluya sobre la influencia que tiene el paso de adaptación en la frecuencia de la senoide de entrada en el proceso adaptativo.

R:/ El paso de adaptación determina la velocidad con que el sistema responde a un cambio en la entrada de la señal.

## IMPLEMENTACION EN TIEMPO REAL

11. Conecte el módulo a la alimentación y al PC a través del cable paralelo.

12. Conecte un generador de señales y una de las entradas del osciloscopio a la entrada analógica del módulo y la otra entrada del osciloscopio a la salida del módulo. Utilice cables BNC-BNC para las conexiones y para conectar el osciloscopio y generador a la entrada del módulo a la vez, utilice una T-BNC.

13. Encienda el módulo.

14. Analice el programa **adaptfir.asm** que implementa un filtro adaptativo, note que en el programa el paso de adaptación esta en 0.0.

15. Encienda el módulo del DSK y ensamble el programa **adaptfir.asm** ubicado en la carpeta DSP.
16. Luego cargue y ejecute el archivo ensamblado **adaptfir.dsk** en el software DSK 1.0 seleccionando la opción *Cargar archivo DSK* en el menú *Archivo*.
17. Aplique a la entrada una señal sinusoidal y visualice la salida con el osciloscopio, utilizando los cables apropiados para tal fin. Varíe la frecuencia entre 800Hz y la frecuencia de *Nyquist*.
18. Coloque el paso de adaptación en  $1.0E-12$ , obtenga un nuevo ejecutable y córralo. Aplique a la entrada una senoide de 1KHz, luego varíe la frecuencia y observe como se comporta el sistema.
19. Repita el paso anterior para diferentes valores de paso de adaptación.
20. Salga de la aplicación seleccionando en el menú *comandos* la opción *reiniciar*. Y luego haga clic en *salir*.

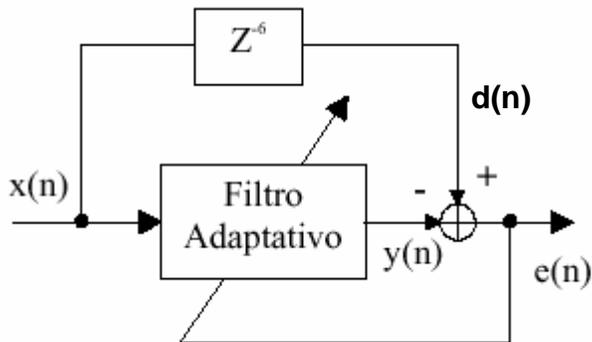
## PREGUNTAS SOBRE EL DESARROLLO DEL LABORATORIO

6. De acuerdo a los tipos de filtros adaptativos estudiados, ¿cuál cree usted que fue el utilizado en esta experiencia?

R:/ Eliminación de interferencia.

7. Identifique en el diagrama de bloques del filtro empleado, las variables utilizadas en la simulación y en la implementación en tiempo real.

R:/



$d(n)$ : Señal

$x(n)$ : Interferencia (Para el ejemplo corresponde a la misma señal de entrada, pero con una desviación)

$e(n)$ : Error

8. De acuerdo a lo observado en el ítem 7 de la experiencia, para un paso de adaptación nulo ¿a qué forma se reduce el sistema adaptativo?

R:/ De acuerdo a este paso de adaptación el sistema se reduce esencialmente a un filtro pasabanda entre 800Hz y la frecuencia de Nyquist.

9. ¿En que influye en el que el paso de adaptación sea mayor o menor? ¿qué efecto produce?

R:/ El paso de adaptación influye en la velocidad con que el sistema adaptativo trata de eliminar la señal de interferencia.

10. Explique con sus palabras el efecto de este filtro, observado en el ítem 8

R:/ Este sistema adaptativo utiliza el error para minimizar la interferencia, el error es realimentado al filtro adaptativo como tal.

## **CONCLUSIONES Y OBSERVACIONES**

Los filtros adaptativos son sistemas de control realimentados que utilizan un factor de adaptación necesario para la minimización del error.

Entender un sistema adaptativo, necesita la comprensión del propósito para el que se implementó. Puesto que los sistemas adaptativos son demasiado específicos.

## 7. TRATAMIENTO DE SONIDO

### OBJETIVOS

Objetivo General:

El propósito de esta práctica es simular la generación de las vocales a través de un filtro “all pole” de 6 orden.

Objetivos Específicos:

- Analizar una simulación en ambiente MATLAB de una implementación de un filtro para la generación de las cinco vocales.
- Observar un programa de ejemplo, desarrollado para reconocer las vocales a través de un micrófono por medio del PC.

## **TEMAS DE CONSULTA**

Filtros IIR. Tratamiento de sonido y voz.

## **BIBLIOGRAFÍA**

The Scientist and Engineer's Guide to Digital Signal Processing by Steven W. Smith. (anexo en el CD del laboratorio en la carpeta anexos teoricos).

## **MARCO TEORICO**

En este trabajo de laboratorio, se hará uso de conceptos ligados a características de la señal de habla, para el que podrá ser conveniente (pero no obligatorio) efectuar una pesquisa adicional, para ahondar mas en el tema.

Desde el punto de vista de este trabajo, interesa fundamentalmente percibir que los sonidos vocales de la señal de habla (por oposición a los no vocales), tienen un carácter periódico y cómo tal su representación espectral puede ser aproximada por una estructura armónica de componentes sinusoidales cuya magnitudes son definidas por una envolvente espectral que depende de cada persona y del sonido vocal pronunciado. Se puede decir que la frecuencia

fundamental de la estructura harmónica (pitch”) identifica la tonalidad de la voz, y que a cada vocal está asociada una envolvente dada espectral cuyos primeros tres máximos locales son los formantes; F, F2 y F3.

A modo de facilitar el entendimiento de estos aspectos, se dispone de una aplicación desarrollada en el ámbito de un proyecto Ciencia Viva, aún en curso, y cuyo ensayo se recomienda bastante. Para los sonidos vocales, ésta aplicación identifica estructuras harmónicas, identifica la frecuencia fundamental, así cómo los primeros tres formantes. Con base en esto, se efectúa también un reconocimiento incipiente de las vocales

## **PRELIMINARES**

Nuestro objetivo será hacer primero una simulación en MATLAB y observar y concluir sobre los resultados obtenidos, para luego pasar al programa de ejemplo de ciencia viva y así tener las bases para un implementación futura en tiempo real en el modulo del DSK.

## PREGUNTAS DE REPASO

3. ¿Qué es el espectro de una señal?

R:/ Es la representación en frecuencia de una señal, generalmente es una grafica de magnitud contra frecuencia.

4. ¿Por qué es mas fácil simular una vocal que una consonante?

R:/ Las vocales por tener un carácter periódico, su representación espectral puede ser aproximada por una estructura harmónica de componentes sinusoidales

## PROCEDIMIENTO

### SIMULACION

3. Analice el programa vocales.m que ilustra tanto analítica como gráficamente, en ambiente MATLAB, la generación de la vocal /a/ usando tres formantes. El valor de cada formante (F1, F2, F3) depende de la vocal que se pretende sintetizar, como consta la siguiente tabla. Se pretende, personalice a su realización usando los formantes correspondientes a su

propia voz. Note que el tacto vocal es modelado por un filtro IIR “all-pole” de 6ª orden, que determina la envolvente espectral de los sonidos vocalizados (en el que se incluyen la señal de habla correspondiente a las vocales). El sonido vocalizado de cada vocal sintetizada, se genera de forma simple, excitando el filtro IIR con un tren de impulsos, a una frecuencia de cerca de 100, 200 o 300 por segundo, para hombres, mujeres y niños, respectivamente. Para nuestro caso usamos una frecuencia de 200 Hz. Como alternativa, podrá usar a frecuencia correspondiente a la tonalidad de su propia voz.

Vogal	F1 (Hz)	F2 (Hz)	F3 (Hz)
/à/	576	1202	2652
/é/	523	1467	3257
/i/	389	1794	3212
/ó/	502	930	2632
/u/	306	675	2508

Tabla 7.1

4. Modifique el programa vocales.m de modo que genere cada una de las cinco vocales y escuche el resultado. Para simplificar, no altere el radio de los polos asociados a cada formante, solo altere la frecuencia. Saque conclusiones.

R:/ La generación no es tan fiel, pero si se aproxima a lo que podría ser un principio básico de sintetización de voz.

## IMPLEMENTACION EN TIEMPO REAL

Ahora procederemos a implementar y analizar el comportamiento del reconocedor de vocales y tacto vocal de ciencia viva, para tener un principio de cómo es el reconocimiento de voz. Ejecútelo y compárelo con los resultados obtenidos en MATLAB.

8. Descomprima el archivo *Cviva.zip*.
  
9. Cambie la resolución de la pantalla a 1024 x 768 antes de ejecutar el programa para que pueda ver la totalidad de este, de lo contrario solo le saldrá parte del “pantallazo”
  
10. Ejecute el archivo *Project1*, en el encontrará varias partes, dos graficas con la amplitud y potencia espectral de la voz, un botón para seleccionar el tipo de visualización, la cual esta dividido en dos; *vogais*(vocales) y *harmónica*, en la opción *vogais*, no dará un reconocimiento de la vocal que pronunciamos por el micrófono y la opción *harmónica*, analizara las

componentes de la voz para decirnos si el hablante es hombre mujer o niño, mediante unos dibujos que aparecerán al momento de hablar. no dirá la tonalidad que es la que determina el tipo de persona y las formantes que son las que detectan el tipo de vocal pronunciada. Y por último tenemos el botón de salir de programa, para utilizarlo en el momento que dejemos de trabajar con él.

11. Utilice primero la opción *harmónica* y hable por el micrófono, por medio de la tonalidad de su voz le dirá si el hablante es hombre, mujer o niño.

12. Ahora pronuncie una vocal y observe las frecuencias de las formantes y compárelas con las de la tabla para cada vocal pronunciada.

13. Ahora coloque el modo de visualización en *vogais* y pronuncie una por una las vocales hasta que el programa sea capaz de reconocerlas todas.

14. Cierre el programa.

## PREGUNTAS SOBRE EL DESARROLLO DEL LABORATORIO

2. ¿Cómo cree usted que se podría implementar en tiempo real un efecto de generación de vocales?

R:/ Implementando la ecuación obtenida mediante un proceso de identificación que se realiza, con base en los datos de frecuencia vistos en la aplicación *Cviva* y los tabulados en la tabla 7.1.

## CONCLUSIONES Y OBSERVACIONES

- Para la síntesis de cada vocal solo basta con alterar las frecuencias de las formantes,  $f_1$ ,  $f_2$  y  $f_3$ , de acuerdo con la tabla dada en la guía de laboratorio. Es importante escuchar el sonido a través de los parlantes del PC los cuales deben estar apropiadamente conectados.
- Después de escuchar la síntesis de cada vocal, SE aprecia que dichos sonidos son señales periódicas, las cuales son mas fáciles de simular que una señal que no lo es, puesto que se forma de un ciclo repetitivo de unas componentes de frecuencia.

- Después de probar el programa de ciencia viva en el modo armónica, pudimos comprobar los datos de la tabla con relación a las componentes de frecuencia de cada vocal, se pudo comprobar que la tonalidad es propia de cada individuo, sea hombre mujer o niño y que se encontraba dentro de los parámetros aprendidos en la teoría.
- En el modo vogais, se observó que es importante que la vocal sea pronunciada lo mas clara y seguramente posible para que el programa sea capaz de reconocerla, además nos da al principio unas aproximaciones de cerca de que vocal se encuentra la que estamos vocalizando hasta que este sea capaz de reconocerla, se pudo notar que habían dos vocales que siempre tendían a confundirse.
- La envolvente espectral de los sonidos vocalizados es propia de cada persona . en el algoritmo se obtiene aplicando un filtro IIR. Realizar una implementación en tiempo real bastaría elaborar el filtro IIR en lenguaje ensamblador , lo cual esta fuera del alcance de sete laboratorio. Se propone la realización de un proyecto basado en esta experiencia demostrativa. Dejamos el algoritmo de un filtro IIR DE 6° ORDEN, tentativamente para su estudio y posible aplicación a la síntesis de voz. El archivo es: *iir6.asm* está incluido en el CD del laboratorio.

**LABORATORIO DE TRATAMIENTO DIGITAL DE SEÑALES**

**MANUAL DE MANTENIMIENTO Y OPERACIÓN DEL MODULO DE PRÁCTICAS  
Y DEL SOFTWARE C3XDSK 1.0**

**FRANCISCO JAVIER OSORIO MARTÍNEZ  
JHONNY MARK VELA SALCEDO**

**CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLIVAR  
FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA  
CARTAGENA DE INDIAS, D. T. H. Y C.**

**2003**

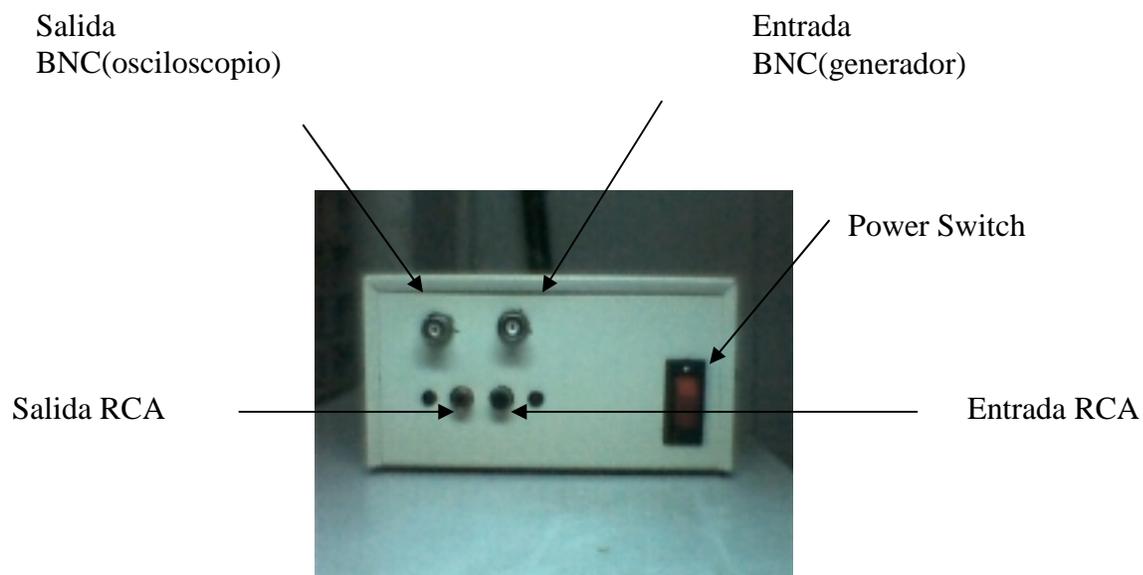
## CONTENIDO

	Pag.
<b>1. Mantenimiento Y Operación Del Modulo de prácticas</b>	<b>172</b>
1.1 Descripción del modulo	172
1.2 Conexión Del Módulo	176
1.3 Recomendaciones	176
<b>2. Mantenimiento Y Operación Del Software C3xdsk 1.0</b>	<b>177</b>
2.1 Generalidades del DSKC3X versión 1.0 para Windows	177
2.2 Requerimientos del software	177
<b>2.3 Operación del software</b>	<b>178</b>
3. Software adicional	192

## 1. Mantenimiento Y Operación Del Modulo de prácticas

### 1.1 Descripción del módulo

#### Panel Frontal

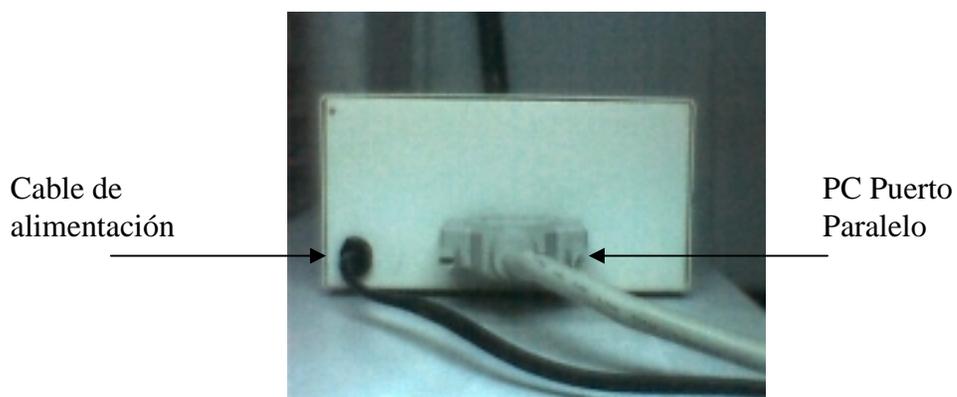


El modulo esta equipado con conectores BNC tanto de entrada como salida para facilitar la conexión a elementos como el osciloscopio y el generador de señales por medio de cables diseñados para tal propósito, con el fin de reducir al máximo el ruido por superficie de contacto y garantizar unos datos fiables a la hora de medir e insertar una señal por mas pequeña que sea.

También se dispone de conectores de entrada y de salida RCA para cuando se desee introducir una señal de audio por ejemplo, y a la salida conectar un parlante para analizar procesos de filtro en tiempo real, además se proporcionan los cables para hacer las conexiones más fáciles y rápidas, manteniendo a la vez la fiabilidad del sonido, dichos cables están equipados con jacks y conectores RCA adaptados para conectar como fuente de sonido una unidad de CD corriente de computador.

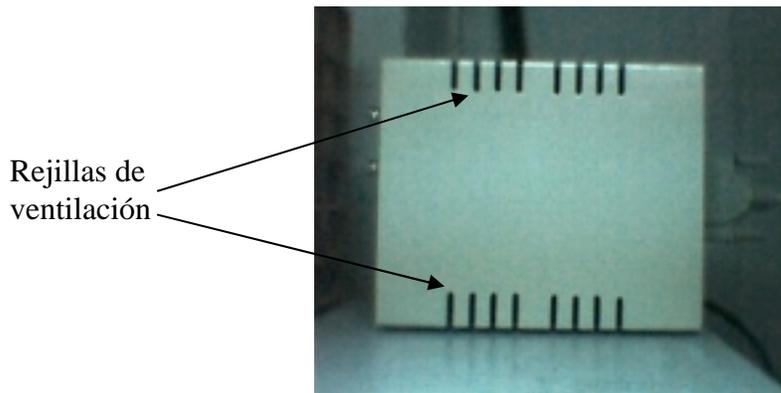
El Power Switch o interruptor general, dispone de una luz piloto para indicar el encendido del modulo, además dentro del modulo se puede observar un led que cambia de color, el cual nos indicara cuando la tarjeta ya entro en comunicación con el PC y esta conectada al software.

### Reverso



Por la parte posterior del módulo podemos observar un cable de alimentación con doble aislamiento y su respectivo seguro de agarre para evitar daños por jalones o enredos. También encontramos un conector DB25 macho para puerto paralelo con su respectivo cable el cual nos servirá de interfase entre el PC y el módulo a la hora de ejecutar los programas.

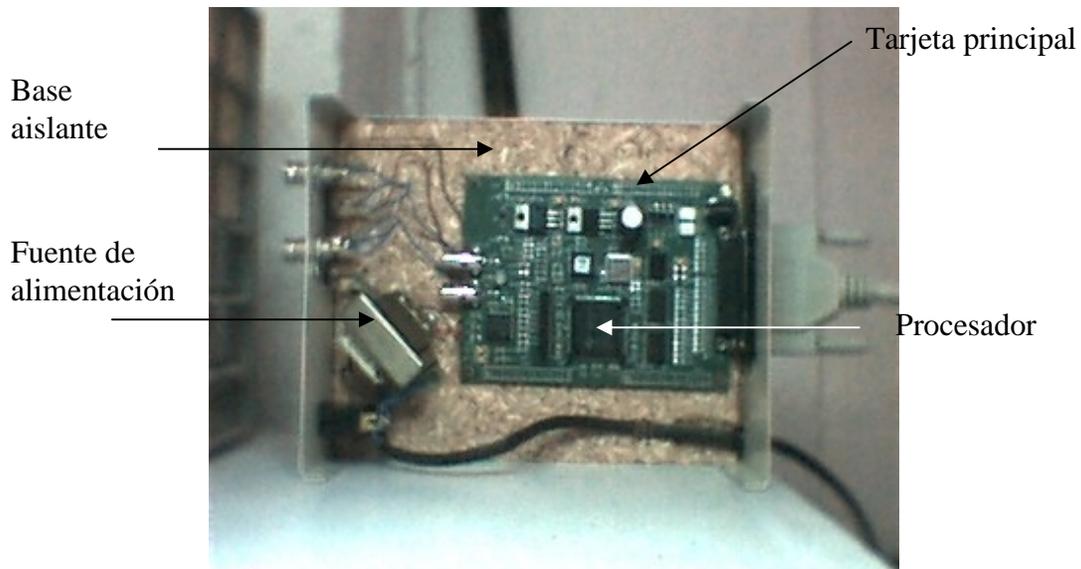
### Vista Superior



La caja del modulo en la parte superior esta equipada con rejillas de ventilación para disipar el calor, además toda la carcasa es metálica y esta conectada a una tierra común para disminuir el ruido por interferencias externas.

Dicha tierra cubre toda la caja y sobre ella están conectadas las tierras de los conectores de entrada y salida, tanto los BNC como los RCA a excepción del conector del puerto paralelo cuya tierra es independiente.

## Interior



En la parte interior podemos observar la tarjeta principal con su respectivo procesador digital de señales (TMS320C31), una fuente de alimentación y cableado, todo sobre una base aislante en aglomerado.

La fuente es de 9Vac, con aproximadamente 800mA de corriente suministrados a través de un transformador, dentro de la tarjeta se encuentra el rectificador de onda completa, filtro y reguladores de -5V y 5V integrados, además de un led que varia en 3 colores que nos indicara cuando hay alimentación y cuando la tarjeta estableció comunicación con el software.

## **1.2 Conexión Del Módulo:**

- Conecte el módulo a una toma de 110 V con el cable de alimentación negro que está ubicado en su parte posterior.
- Conecte el cable DB25 blanco suministrado con el módulo de la siguiente forma: el lado hembra del cable al conector macho ubicado en la parte posterior del módulo y el lado macho del cable al puerto paralelo del PC a utilizar. El módulo está listo para utilizar. Energícelo utilizando el power switch rojo con luz piloto.

## **1.3 Recomendaciones:**

- No destape el módulo, la carcasa hace las veces de Jaula de Faraday para protección antiestática. No es necesaria la manipulación interna del módulo. La tarjeta está debidamente aislada. Y además el módulo posee los conectores necesarios para conexión de cables.
- Aunque la fuente interna está debidamente estabilizada, es recomendable que sea utilizado un estabilizador de voltaje para proteger la tarjeta de alguna posible sobretensión.
- El PC con que se esté interactuando, debe estar debidamente protegido (estabilizador de tensión, supresor de picos y puesta a tierra). Verificar el

correcto funcionamiento del puerto paralelo asegurándose que el modo en que se esté trabajando sea bidireccional. Si la configuración del puerto no es bidireccional la comunicación con la tarjeta podría no funcionar.

## **2. Mantenimiento Y Operación Del Software C3xdsk 1.0**

### **2.1 Generalidades del DSKC3X versión 1.0 para Windows**

La herramienta de software desarrollada, llamada DSKC3X versión 1.0. Permite interactuar con la tarjeta TMS320C3X DSK, cargando los programas en al memoria, compilando los archivos en ensamblador (.asm) pasándolos a archivos objetos (.dsk), implementando filtros con la posibilidad de modificar los parámetros en tiempo real y transformadas rápidas de Fourier (FFT).

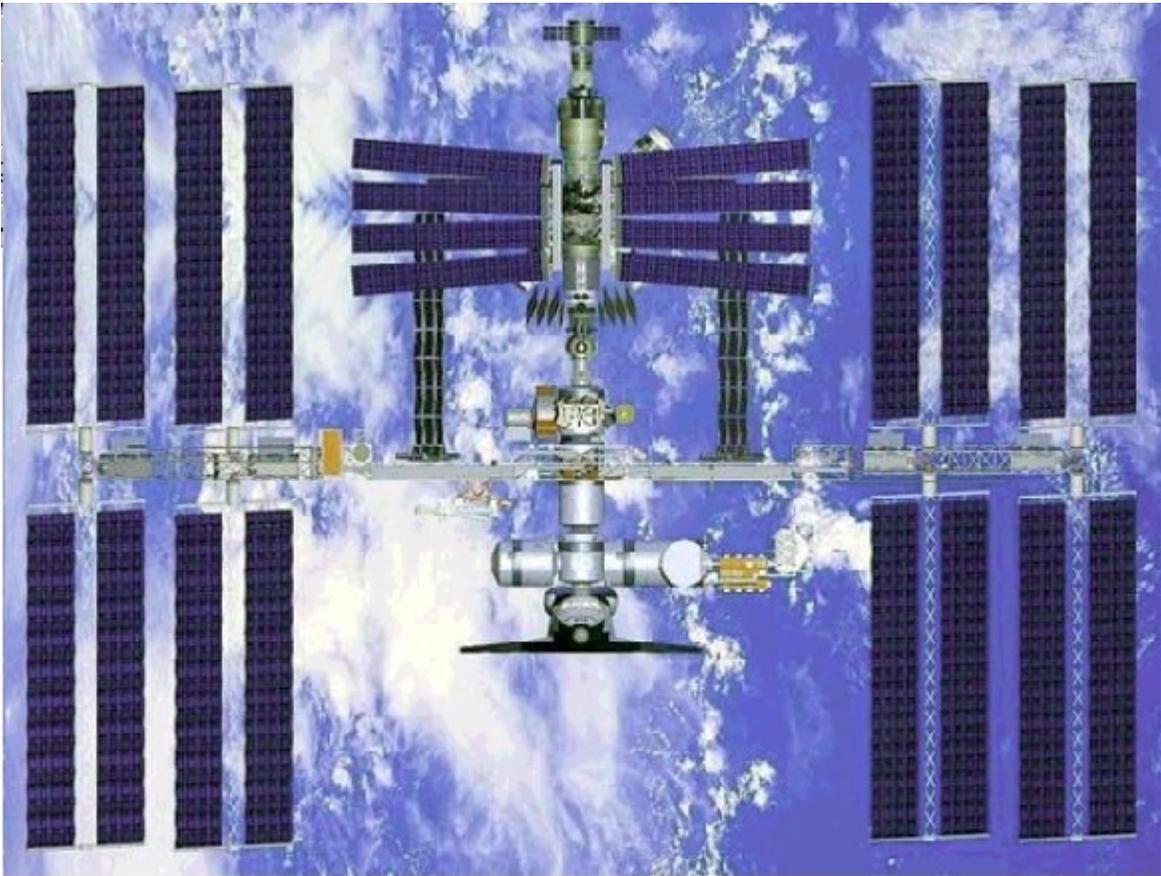
### **2.2 Requerimientos del software:**

- El programa requiere para su funcionamiento un ordenador de la familia 80x86, PC compatible, con los requerimientos hardware necesarios para hacer funcionar correctamente un entorno WINDOWS 95/98/2000 (no NT).
- Se aconseja un procesador 80586 o tipo Pentium.

- El espacio en disco necesario será alrededor de 100 MB, albergar el fichero ejecutable, programas en lenguaje ensamblador del TMS y sus correspondientes ficheros compilados, así como el fichero de ayuda.
- Se recomienda mínimo 64 MB de memoria RAM
- Se requiere una resolución de 800x600 pixeles. Con fuentes pequeñas.
- Se hace prácticamente imprescindible el uso del ratón para un cómodo y eficaz funcionamiento de la aplicación, si bien ésta está diseñada para poder ser manejada desde teclado.
- El puerto paralelo del PC debe ser bidireccional.

### **2.3 Operación del software**

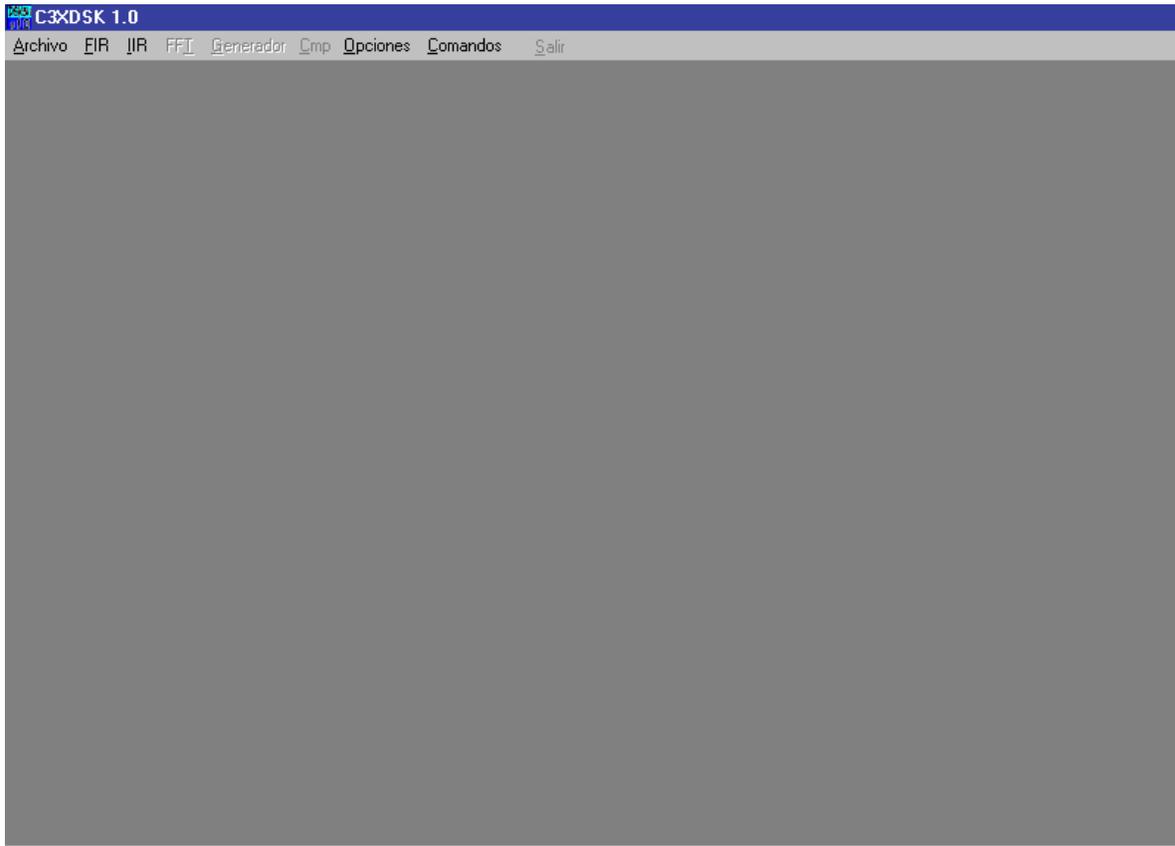
El software se instala ejecutando un autoextraíble ubicado en la carpeta DSP, que viene en el CD del laboratorio. los archivos deben ubicarse en la carpeta llamada C3XDSK. El archivo ejecutable se llama c3xdsk.exe. Al ejecutar este archivo aparece el siguiente pantallazo:



**Software Para Laboratorio De DSP.  
Utilizando La Tarjeta TMS320c3x.  
Por F Osorio y J Vela.**

---

Inmediatamente aparecerá la siguiente ventana:



Con un sistema de menú descrito a continuación:

## MENÚ ARCHIVO

Este menú contiene las funciones básicas siguientes:

### ***Cargar Filtro***

Esta opción permite cargar ficheros de extensión FLT(desde la unidad c: y a:), que contienen parámetros de filtros FIR o IIR, indicando la ruta y nombre del fichero en la caja de diálogo que aparece.



### ***Salvar Filtro***

Esta opción permite guardar en disco los parámetros del último filtro FIR o IIR que se haya seleccionado (si es que hay alguno). Se puede salvar con el nombre que se quiera, la extensión tendrá que ser .FLT



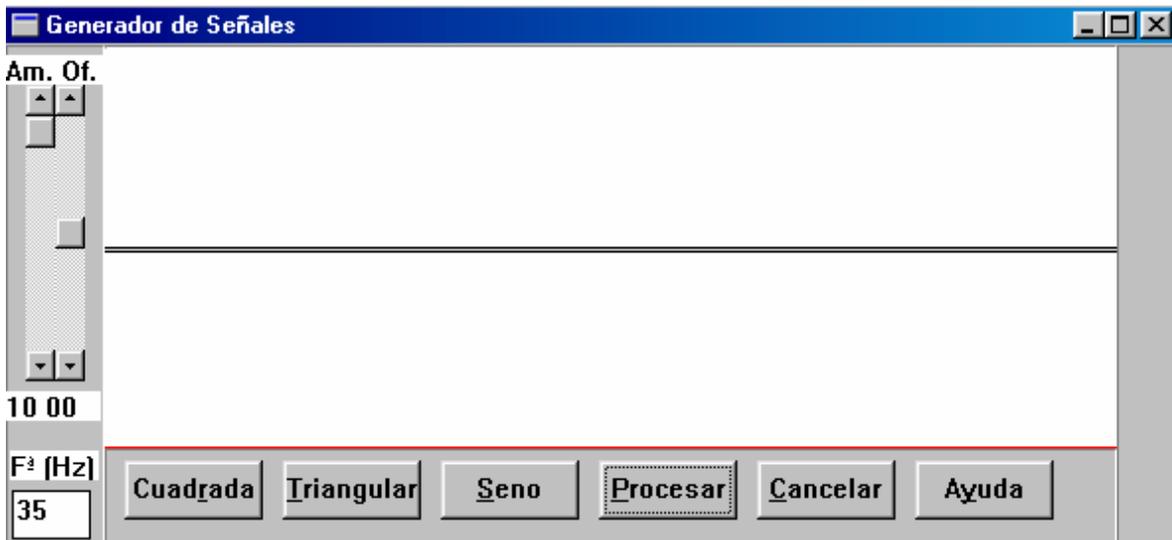
Al igual que para lectura, indicaremos al programa los datos de ruta y nombre con el que queremos salvar el fichero.

### ***Cargar Archivo .dsk***

Esta opción, que sólo permanece disponible en modo conectado, permite cargar programas en la tarjeta del DSP de extensión .DSK, indicando la ruta y nombre del fichero en la caja de diálogo que aparece.



## MENU GENERADOR



En modo conectado permite seleccionar una función de forma senoidal, cuadrada o triangular que se puede obtener a la salida del módulo de prácticas mediante la tarjeta.

También se puede hacer modificaciones sobre la señal que se ha seleccionado, la forma de modificar dicha señal es usando el ratón. El botón derecho del ratón sirve para terminar una modificación de un tramo de la señal. El botón izquierdo del ratón sirva para dibujar un tramo de la señal, de esta forma se puede dibujar un periodo de una señal cualquiera pudiendo visualizarla a la salida de la tarjeta por medio de un osciloscopio.

La opción de amplitud situada en una barra de desplazamiento a la izquierda del cuadro de dialogo sirve para variar el factor de amplitud de la señal seleccionada o modificada, los márgenes de ese factor de amplitud de la señal están comprendidos entre -1 y 1, este factor de amplitud aparece en la parte inferior de la barra de desplazamiento.

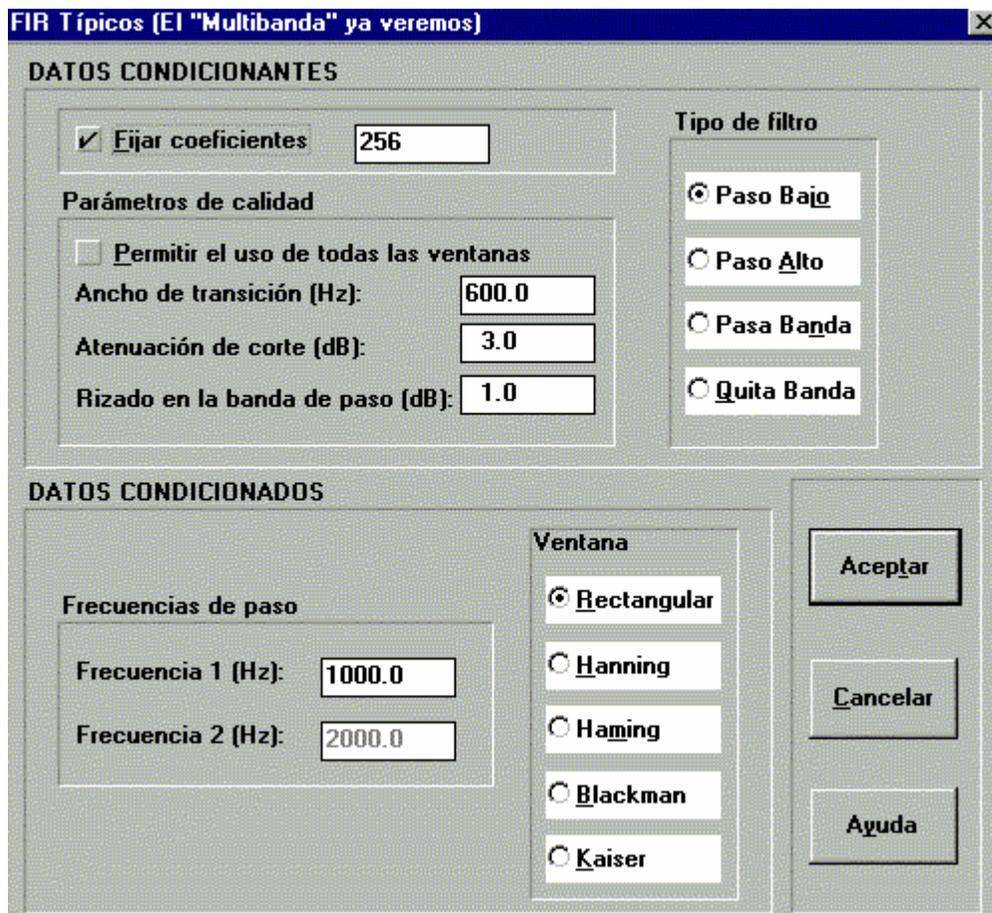
La opción de offset sirve para añadir o disminuir una cierta componente continua de la señal, de esta forma se puede conseguir cualquier tipo de señal periódica con una cierta componente continua siempre y cuando no se paso de las márgenes permitidos.

Cuando se pulsa la opción procesar se mandarán todos los datos a la tarjeta situada en el módulo de practicas y entonces se podrá visualizar la señal dibujada por medio de un osciloscopio.

## **MENU FIR**

### ***FIR Típicos***

Esta opción ofrece la posibilidad de implementar un filtro FIR paso-bajo, paso-alto, paso-banda o rechaza-banda mediante el método de enventanado según los parámetros que introduce el usuario en el cuadro de diálogo que se abre.

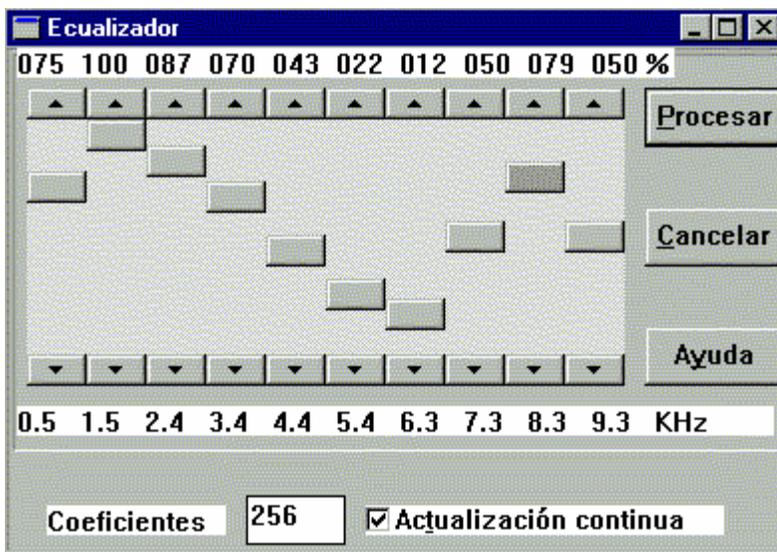


### ***Ecualizador FIR***

Esta opción ofrece la posibilidad de implementar un ecualizador FIR multibanda. Para ello, se divide la banda comprendida entre 0 Hz y la frecuencia de Nyquist en 10 intervalos, a cada uno de los cuales se le asigna una ganancia. Las ganancias se seleccionan usando las barras de scroll que aparecen en la ventana. Los coeficientes de este filtro,  $hmb[n]$ , se calculan según la expresión:

$$h_{mb}[n] = \sum_{k=1}^{N_{mb}} (G_k - G_{k+1}) \frac{\text{sen } 2\pi f_k (n - M/2)}{\pi (n - M/2)}$$

siendo  $N_{mb}$  el número de intervalos que se consideran;  $G_k$ , para  $0 < k < N_{mb}$ , la ganancia de cada banda del filtro y  $G_{N_{mb}+1} = 0$ , y  $f_k$  la frecuencia límite normalizada de cada banda.



Es necesario mencionar que no se ha seguido un criterio específico para la elección del tipo, orden o demás parámetros de los filtros, a no ser las frecuencias de corte adyacentes, dado que lo que se pretende es demostrar la facilidad de implementación de un sistema simple y no el diseño de un ecualizador óptimo, el

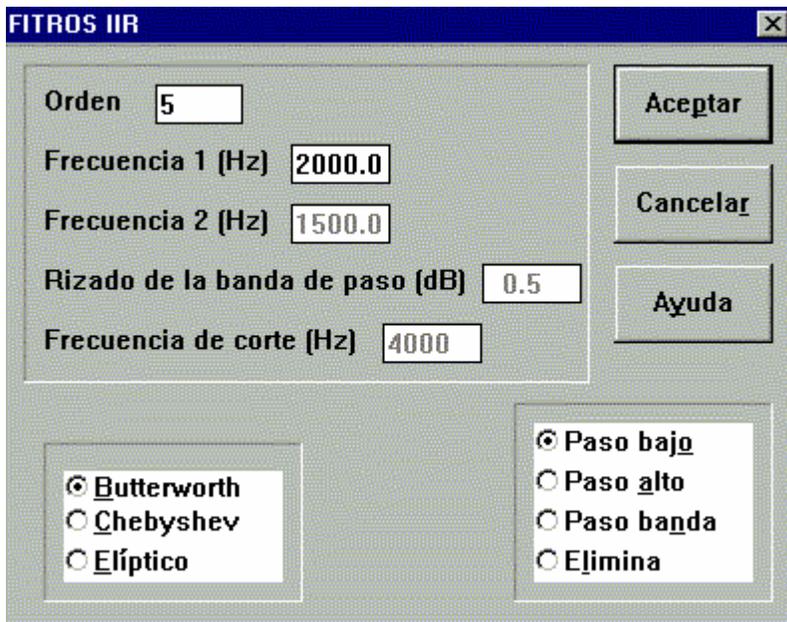
cual sin embargo podría bien ser implementado con las mismas herramientas aquí usadas.

La opción Actualización continua permite al usuario de un PC medianamente potente (un Pentium cualquiera) el procesado continuo de los variaciones que se introducen, y así estar representando constantemente las características y coeficientes del filtro, y si se encuentra en modo conectado, transmitir de forma continua los cambios en los coeficientes a la tarjeta. En otro PC menos potente esta opción ralentiza demasiado el sistema, por lo que cuando se quieran ver los resultados de los cambios introducidos, o mandar los coeficientes a la tarjeta habrá que pulsar procesar, una vez que se ha pulsado a procesar tendremos un filtro con las características que se ha dibujado con la barra.

## **MENU IIR**

### ***IIR***

Esta opción ofrece la posibilidad de implementar un filtro IIR paso-bajo, paso-alto, paso-banda o rechaza-banda mediante el método de la transformación bilineal a partir de un filtro analógico de Butterworth, Chebyshev o elíptico según los parámetros que introduce el usuario en el cuadro de diálogo que se abre.



La frecuencia 1 y frecuencia 2 son las frecuencias críticas con las que el programa C3XDSK 1.0 calculará los coeficientes. Frecuencia 1 siempre estará habilitada, la frecuencia 2 se habilitará cuando el programa necesiten 2 frecuencias, es decir, en los filtros paso banda y elimina banda.

La opción rizado de banda de paso habrá que expresarla en dB y como su propio nombre indica es el rizado en la banda de paso, o dicho de otra manera son las oscilaciones que son producidas por aproximación de un filtro teórico a uno práctico. Esta opción estará habilitada cuando se tenga seleccionado el filtro Chebyshev o filtro Elíptico.

La opción frecuencia de corte puede ser positiva y negativa, cuando se pone como positiva el programa la interpretará como frecuencia de corte en Hz, pero si se pone como negativa entonces se interpreta como la atenuación de dicha frecuencia expresada en decibelios (dB). Esta opción es usada por el programa cuando se quiere realizar el cálculo de coeficientes para un filtro elíptico, en el resto de tipos de filtros esta opción estará deshabilitada.

El orden que se introduce en la casilla de orden será el orden de filtro que se desea realizar, lógicamente cuando más alto sea este número tenemos el inconveniente de que hay que hacer un mayor número de operaciones, en contra se obtendrá un filtro mucho más selectivo.

## **MENU FFT**

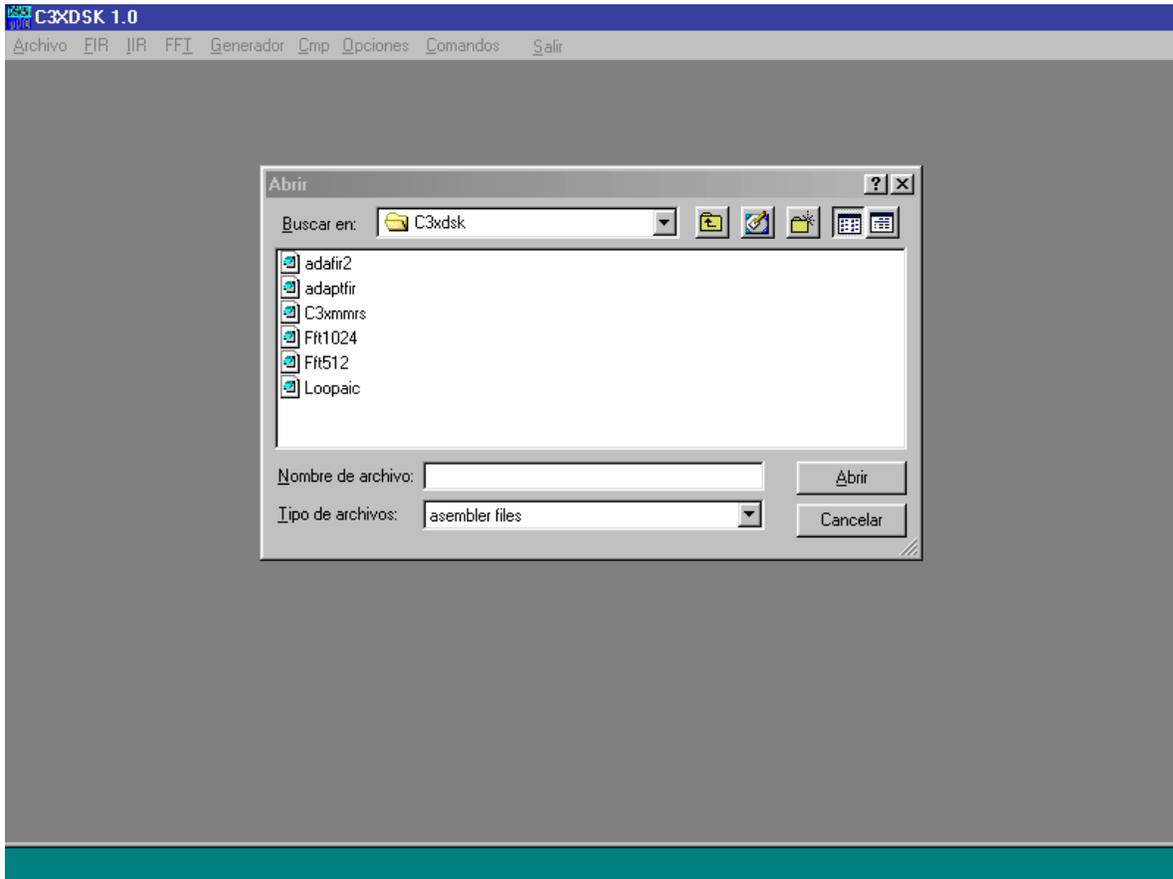
### ***FFT***

En modo conectado, carga en la tarjeta del DSP el programa "fft\_fir.dsk", que calcula la transformada rápida de Fourier de la señal de entrada cuando el host lo precisa. De este modo se puede ver la transformada de cualquier señal mediante el mediante el módulo de practicas.

### **Menú Compilar**

Esta opción despliega un cuadro de dialogo donde se elige un archivo .asm y se compila para obtener un archivo objeto .dsk que se el que va cargado en la memoria del dispositivo. Este archivo queda guardado en la carpeta de trabajo y está listo para cargarse con la opción cargar archivo .dsk del menú archivo explicado anteriormente.

El cuadro resultante que aparece es:



### 3. Software Adicional

C3xcodeexplorer 1.0 ensambla, ejecuta posibilita ver localidades de memoria bajo Windows.

Code Composer. Compilador de lenguaje c para programas del TMS320C3XDSK.

WAVEGEN generador de funciones utilizando al tarjeta de sonido PC.

SCOPE osciloscopio utilizando al tarjeta de sonido del PC.

Se encuentran añadidos al la carpeta software adicional del CD del laboratorio.